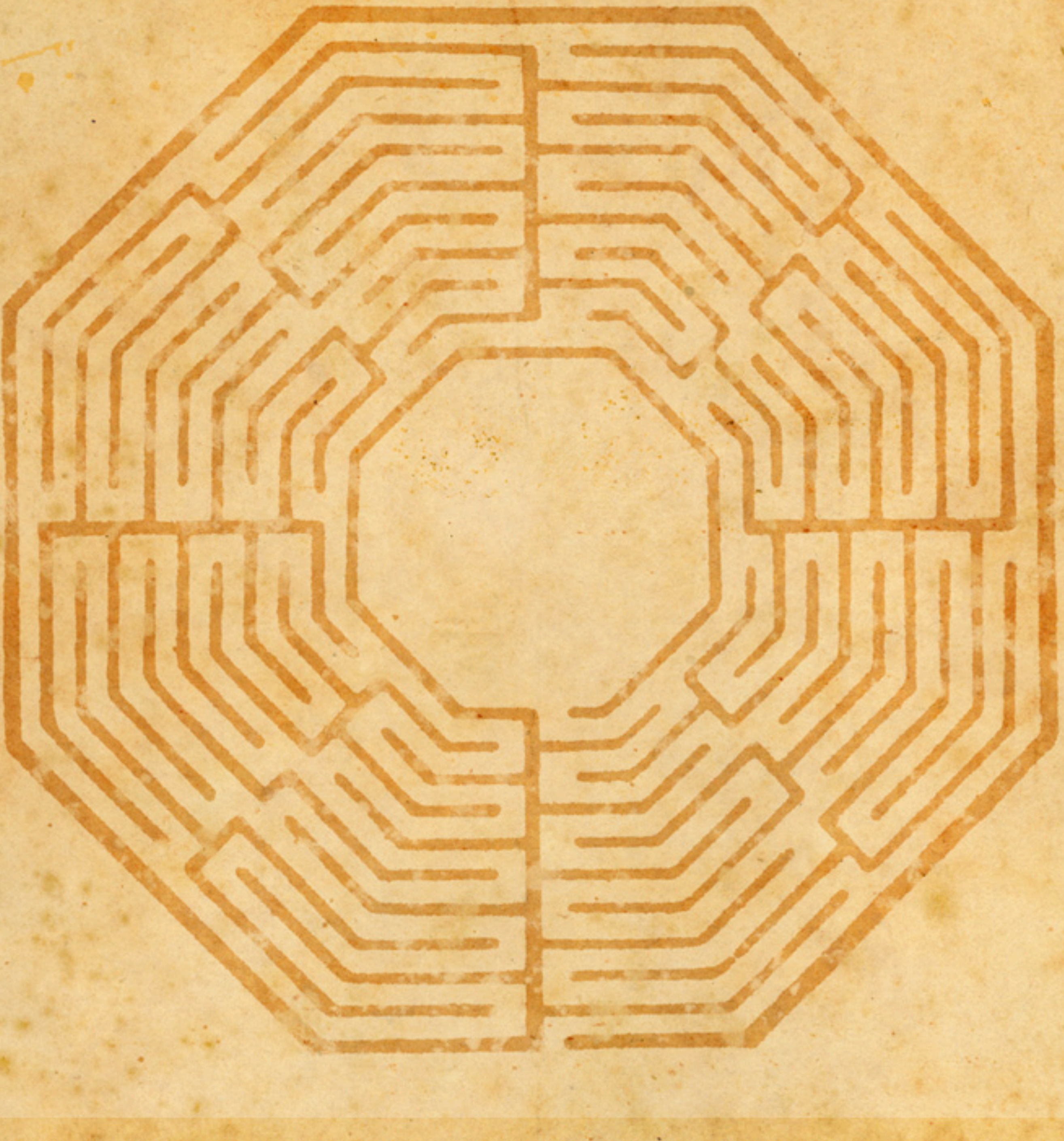
(IN)SECURE

OPEN. INFORMATIVE. TO THE POINT. Issue 18 - October 2008



PROACTIVE DEFENSE - COMPLIANCE - PRIVACY

BROWSER SECURITY - SOFTWARE ARMORING

WEB APPLICATION SECURITY

TABLE OF CONTENTS

Welcome to another issue of (IN)SECURE filled with a variety of hot topics. It's been a busy summer and we have a lot on the table for you. I had the pleasure to visit Greece earlier this month for the 1st NIS Summer School. From what I've seen, information and network security in Europe are in good hands. More about this fine event at page 13.

In other news, Jo Stewart-Rattray, who was one of the authors featured in the November 2007 issue of (IN)SECURE, wanted to apologize for omitting proper attribution in her article - "Information Security, the Nuts and Bolts". The attribution that should have been included is “Information Security Governance: Guidance for Boards of Directors and Executive Management, 2nd Edition”, IT Governance Institute, 2006. The article has been removed from our archives as soon as questions have been raised and we are satisfied with her prompt response.

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

Kaspersky Internet Security and Anti-Virus 2009 versions

Kaspersky Lab announced the North American release of Kaspersky Internet Security 2009 and Kaspersky Anti-Virus 2009. Kaspersky has entirely rebuilt its award-winning anti-malware security engine. Kaspersky Internet Security 2009 and Kaspersky Anti-Virus 2009 are engineered for speed and are packed with a first-of-its-kind arsenal of tools to help to protect computer users from the rapid growth of malicious cybercrime attacks. (www.kaspersky.com)

Forensics on the fly with ArcSight Logger

ArcSight announced a new release of ArcSight Logger that provides "forensics on the fly." This capability, now available across the entire ArcSight SIEM platform, enables IT and forensics teams to quickly conduct informative top-down investigations. These teams can immediately drill down into source events from dashboards, reports, searches, and alerts both in real-time and in support of after-the-fact compliance audits. (www.arcsight.com)

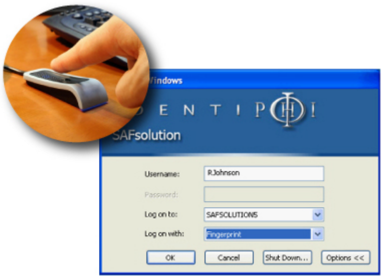
Updated Linux based system lock-down and security management solution

Group Name	Status	Description	Actions	Reports
ITWebsvrs	Active	Corporate Web Servers	Manage	Assessment
HRAdmin	Active	HR Servers	Manage	Assessment
AccountingDB	Inactive	Accounting DB Servers	Manage	

Trusted Computer Solutions announced new compliance features for both the standalone and enterprise versions of Security Blanket. Security Blanket is a system lock-down and security management solution that enables system administrators to automatically configure and enhance the security levels of Linux systems. The new features of Security Blanket provide compliance guidelines for organizations with industrial control systems (ICS), companies that process credit card transactions, and government agencies accessing classified data. (www.tcs-sec.com)

Citrix Ready biometrics with SAFsolution 5

IdentiPHI announced its flagship enterprise security software product, SAFsolution 5, has been verified as Citrix Ready. The Citrix Ready program helps customers identify third-party solutions that add the greatest value in Citrix Application Delivery Infrastructure solutions. SAFsolution 5 completed a rigorous verification process to ensure compatibility with virtualization solutions Citrix XenApp and Citrix Password Manager. XenApp Platinum, which includes single sign-on application security with Password Manager, allows IT to deliver secure applications as a service, providing on-demand access to users while delivering the highest performance and data security. (www.identiphi.net)



Username: RJohnson
Password:
Log on to: SAFSOLUTIONS
Log on with: Fingerprint

High-end cameras for professional security and surveillance

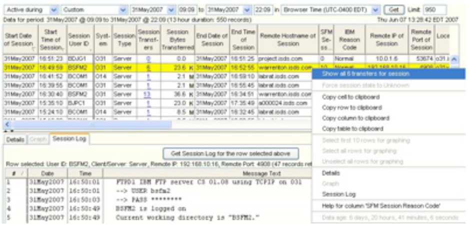
D-Link introduced three high-end fixed network cameras designed for professional surveillance and security applications with features including Day & Night viewing, Megapixel sensors, high power zoom and Power over Ethernet (PoE) support. Cameras released include: D-Link Day & Night Megapixel Camera with PoE Support (DCS-3110), D-Link Day & Night Camera with CCD sensor and PoE Support (DCS-3410) and D-Link Day & Night Camera with 18x Optical Zoom & PoE Support (DCS-3415). (www.dlink.com)

Lenovo mobile phone with fingerprint biometrics

Lenovo Mobile selected Atrua's made-for-mobile fingerprint solution for Lenovo's P960 mobile phone – Lenovo's first commercial mobile phone to incorporate fingerprint biometrics.

Atrua's made-for-mobile fingerprint touch control utilizes the company's own adaptive capacitance and neural matching technology, developed from the outset for mobile applications. (www.lenovo.com)

Secure file transfers for IBM z/OS mainframes



Active during | Custom | 31May2007 | 09:09 to 31May2007 | 22:09 in Browser Time (UTC-0400 EDT) | Get | Limit: 950

Data for period: 31May2007 @ 09:09 to 31May2007 @ 22:09 (13 hour duration: 550 records) — Thu Jun 07 13:28:42 EDT 2007

Start Date of Session	Start Time of Session	Session User ID	Syst-em	Session Type	Session Transf-ers	Session Bytes Transferred	End Date of Session	End Time of Session	Remote Hostname of Session	SFM Se-ss...	IBM Reason Code	Remote IP of Session	Remote Port of Session	Loca
31May2007	16:51:23	BDJG1	O31	Server	0	0.0	31May2007	16:51:25	project.isds.com	0	Normal	10.0.1.6	53674	o31.i
31May2007	16:49:59	BSFM2	O31	Server	6	23.6 K	31May2007	16:52:55	warrenton.isds.com	10	Normal	192.168.10.16	4908	o31.
31May2007	16:41:52	BCOM1	O14	Server	1	2.1 M	31May2007	16:59:10	labrat.isds.com					
31May2007	16:39:55	BCOM1	O31	Server	1	2.1 M	31May2007	16:55:45	labrat.isds.com					
31May2007	16:30:40	BSFM2	O31	Server	13	36.6 K	31May2007	16:34:51	warrenton.isds.com					
31May2007	15:35:10	BJPC1	O31	Server	1	23.0 K	31May2007	17:35:49	a000024.isds.com					
31May2007	15:24:10	BCOM1	O14	Server	1	8.5 M	31May2007	16:32:45	labrat.isds.com					

Details | Graph | Session Log

Get Session Log for the row selected above

Row selected: User ID: BSFM2, Client/Server: Server, Remote IP: 192.168.10.16, Remote Port: 4908 (47 records ret

```
#   Date       Time       Message Text
1   31May2007  16:50:01   FTPD1 IBM FTP server CS 01.08 using TCPIP on O31
2   31May2007  16:50:01   --> USER bsfm2
3   31May2007  16:50:03   --> PASS ********
4   31May2007  16:50:49   BSFM2 is logged on
5   31May2007  16:50:49   Current working directory is "BSFM2."
```

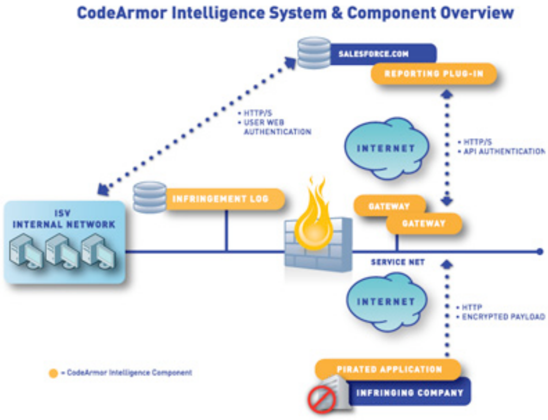
Show all 6 transfers for session
Force session state to Unknown
Copy cell to clipboard
Copy row to clipboard
Copy column to clipboard
Copy table to clipboard
Select first 10 rows for graphing
Select all rows for graphing
Unselect all rows for graphing
Details
Graph
Session Log
Help for column 'SFM Session Reason Code'
Data age: 6 days, 20 hours, 41 minutes, 6 seconds

SSH Communications Security and Software Diversified Services (SDS) announced a comprehensive encryption and secure FTP automation solution for IBM mainframe customers. Through this partnership, SDS will distribute and support a bundled solution combining SSH Tectia; Server for IBM z/OS and the new SDS FTP Manager (SFM) 2.0 product. Together, these products increase the level of security for enterprises conducting batch file transfers on IBM mainframes by combining robust data security with advanced monitoring and management functionality. (www.ssh.com)

CodeArmor Intelligence to combat piracy problems

V.i. Laboratories announced a new anti-piracy strategy that gives software makers multiple options to track, reduce and recover lost revenue due to piracy.

CodeArmor Intelligence is the first turn-key piracy detection and reporting system that integrates with existing applications and Salesforce.com, offering a dynamic method to collect, filter and report on the use of pirated applications and create leads for sales, compliance or legal teams. (www.vilabs.com)



Updated Astaro Security Gateway appliances

Astaro Corporation announced the release of version 7.3 of Astaro Security Gateway. The latest Astaro release offers over 100 new features intended to increase end-user efficiency and secure mail traffic. New features include a redesigned UserPortal for easy management of email and VPN connections, a faster and more accurate Mail Security Engine, a new integrated Active Directory browser, and free email encryption as part of Astaro's Mail Security Package. (www.astaro.com)

New ZoneAlarm Internet Security Suite 8.0

Check Point released ZoneAlarm Internet Security Suite 8.0, featuring core security and performance enhancements along with a new user interface providing consumers and small businesses the safest, fastest and easiest PC and identity theft protection available. ZoneAlarm Internet Security Suite 8.0 expands security with enhancements such as Early Boot Protection, which guards the computer during system start-up where other security products leave systems vulnerable, and Rootkit Protection in the OSFirewall, which blocks attacks targeting processes deep in the operating system. (www.checkpoint.com)

16GB Lexar JumpDrive Secure II Plus USB flash drive

Lexar Media introduced its massive 16GB-capacity JumpDrive Secure II Plus USB flash drive. The 16GB Secure II Plus JumpDrive is intended for consumers and small business users who want to securely back up, store and share large files and amounts of data.

In addition to providing users with 16GB of high speed memory, the JumpDrive Secure II Plus features the innovative and award-winning built-in capacity meter, along with advanced security software that ensures users they can protect their important documents knowing that their information is encrypted — even if the device is lost or stolen. (www.lexar.com)

External hard drives with RFID security key data encryption onboard

Aluratek announced the availability of its new Tornado plus line of external hard drives featuring built-in radio frequency identification (RFID) security key data encryption, fast transfer speeds and storage capacities of up to 1TB.

A swipe of the RFID security key by the external hard drive encrypts the data so that it cannot be accessed without the unique RFID key swipe a second time to unlock it. Two RFID security keys ship with each external drive. (www.aluratek.com)

Panda Security launches its 2009 antivirus products

Panda Security has launched its 2009 range of antivirus solutions for the consumer sector. The product line-up comprises Panda Antivirus Pro 2009, Panda Internet Security 2009 and Panda Global Protection 2009. The new range is based on Collective Intelligence, an innovative web 2.0 security model that generates protection from the “cloud”. The system utilises real time malware information acquired via user community and generates vaccines to neutralise malicious threats. The approach significantly reduces resource consumption and speeds up effective detection and disinfection rates. (www.pandasecurity.com)

Security standpoint
by Sandro Gauci

Closing a can of worms

DNS Cache Poisoning is probably a phrase you have become accustomed to hearing if you have been on this planet for the past 2 months, at least if your work has any relation to Information Security.

On July 2008, a good number of patches came out for various DNS servers such as BIND and MSDNS (Microsoft's DNS) that were found vulnerable to an unspecified security vulnerability. We were told that this DNS Cache Poisoning vulnerability was the result of a feature of the DNS protocol and therefore almost all DNS servers were found to be vulnerable.

Dan Kaminsky had silently been working with DNS vendors over the past months to develop a patch that could be published at one go. Although the idea was to give everyone affected 30 days time until full details of the security flaw were published, this did not keep security researchers from guessing the vulnerability specifics.

Eventually Matasano Security, a security firm that had details of the flaw, leaked out the full details by mistake on July 21st. This was a few days before the official Kaminsky presentation at Blackhat Las Vegas 2008.

Background

How could one exploit the vulnerability which Dan Kaminsky found? The following is a simplified summary of the attack.

The presumption is that the attacker attempting a DNS cache poisoning attack needs to spoof DNS packets to make them appear as valid DNS responses to actual DNS requests. Before this attack became public, it was assumed that the only way to do this was by faking DNS responses for existent names. This involved guessing the right Transaction ID or TXID, which is part of the DNS packet and it is usually randomized. This feature makes such an attack impractical and therefore did not recently affect DNS servers on the Internet.

With the newly disclosed vulnerability, the attacker would generate a large number of requests for nonexistent names on a DNS server (for example, by asking for a.victim.org, b.victim.org, c.victim.org, etc.) and spoof responses for each requests. Before the patch, the only secret between the victim caching DNS server and the authoritative DNS server was the TXID which is only 16 bit long. An attacker could, in many cases, manage to guess a TXID after a couple of seconds of generating DNS requests for nonexistent names and spoofing responses. Apart from a standard response, the attacker's spoofed packet would also contain a DNS record pointing to a host of his or her choosing as an authoritative server.

When the attacker guesses the right TXID this record would be accepted as authoritative which allows for hijacking of the cache entry for that particular domain name. What the patch did was randomize the source port thus creating another 16-bit secret between the legitimate DNS servers. This extends the entropy to 32 bit.

BY HIJACKING THE NAME SERVERS, ATTACKERS CAN ALSO REPLACE EXECUTABLE FILES WHICH ARE DOWNLOADED FROM THE INTERNET WITH ONES THAT CONTAIN MALICIOUS CODE

The implications

The most obvious security issue is that web sites may not be what they appear to be. Someone with malicious intent could make use of this security hole to perform very plausible phishing attacks.

For example, hijack Amazon.com or your favorite bank, and point them to a web server that the attacker controls which asks for the credentials to their account. A phishing website tends to be very convincing by making use of the same layout and template as the legitimate website. If a phishing attack makes use of the DNS flaw the victim usually has little or no way of knowing that he or she is not on the legitimate website.

During his presentation at Blackhat, Dan provided various examples of how the DNS flaw could be exploited for fun and profit. He explained that the whole ".com" could be hijacked.

Many antispam solutions rely on DNS in some form or another; whether it be the blacklists, or the SPF (Sender Policy Framework) that keeps many people's mailbox usable. He gave examples of how attackers can watch Mail Exchange DNS or MX record requests, thus monitoring who is sending emails to whom, and send fake replies to those that seem interesting. This would allow selective snooping of email content - some of which tends to be very sensitive.

Do you remember how you reset your Amazon password last time you forgot it? You fill out a form with your email address and receive a link from Amazon through email. The assumption is that you are the only one reading your email.

By hijacking the name servers, attackers can also replace executable files which are downloaded from the Internet with ones that contain malicious code. An attacker could secretly append malicious code to legitimate application downloads thus leading to remote code execution.

Many organizations have a security policy which allows only the administrators and a few chosen ones are allowed to download and execute third party applications. Guess who also tends to have Windows Domain administrator or root access? This scenario can prove to be disastrous, especially if it goes unnoticed.

What about Automated updates? A new tool was published by Infobyte Security Research (www.infobyte.com.ar) by the end of July called evilgrade. What it effectively does is demonstrate how updates for Java, Winzip, Winamp, iTunes and many others, can be hijacked to automatically introduce malicious code. This attack could be leveraged over the Internet as a result of a DNS hijack or any other means by which an attacker can monitor and modify the victim's network traffic.

Many security professionals were crying out “but isn’t that what encryption is there to prevent?” That is right, but the reality is that we rely so much on clear text traffic in our day to day network usage. Most of the traffic on the network is not encrypted. Even when it is, the encrypted traffic, at least in the case of HTTPS, is typically initiated based on a URL which is sent in clear text. Apart from that, how many times have you actually rejected a certificate that was expired, mismatched or self-signed instead of getting on with work promising yourself that you will look into it later? The sad truth is that SSL alone does not solve our problems.

Even when HTTPS is correctly implemented, it does not mean that the underlying protocols adhere to the security context of the website. Mike Perry gave a talk at Defcon about forcing the web browser to reveal cookies being used on an HTTPS website. Incidentally, yours truly was working on the same research independently from Mike's research and dubbed the vulnerability “Surf Jacking”, also called “Forced Side Jacking” or “Automated HTTPS Cookie Hijacking”. Tools have also been published (see surfjack.googlecode.com and fscked.org/projects/cookiemonster) to demonstrate this issue and various high profile company names were found to be vulnerable to attack.

The security issue relies on the premise that the attacker can view the HTTP traffic being sent and received by the victim. A DNS cache poisoning attack is one way that an attacker can achieve this. Even though an HTTPS session is by its nature encrypted, if the session cookie is not flagged as secure then it will be sent to both clear text and encrypted versions of the website. An attacker can therefore force the victim to browse to the clear text version of the website, thus revealing the session cookie. This allows the attacker to set the session cookie on his browser and gain access to the victim’s account on the target HTTPS server.

YOU DO NOT NECESSARILY NEED TO PERFORM A DNS CACHE POISONING ATTACK TO BE ABLE TO VIEW OTHER PEOPLE’S TRAFFIC

Is this just about DNS?

Most of the concerns mentioned in this article are not just DNS specific. You do not necessarily need to perform a DNS cache poisoning attack to be able to view other people’s traffic.

Many of the examples given in this article become practical when a malicious party can view the victim’s traffic. The following are a few typical and obvious scenarios where this has been possible for quite a while:

- Insecure wireless. Think about the free wireless at the hotel on your last business trip.
- The local area network is typically also vulnerable when an attacker makes use of an ARP cache poisoning attack.

Apart from that, there are more subtle cases where traffic may be intercepted by malicious users. A few months before the DNS cache poisoning issue was published, a totally separate security flaw was patched. For the most part, no one seemed to notice this security flaw which affected SNMP version 3 used in many major network device vendors such as Cisco, Juniper and so on. The flaw allows efficient bruteforce attacks, which can lead to attackers gaining control of network routers and other vulnerable devices. A post on The “Recurity Lablog” (www.phenoelit.net/lablog) is one of the few that covered this vulnerability and explains how compromised routers can affect all network traffic rather than just traffic that relies on correct DNS resolution. Of course, making use of IP addresses instead of domain names is quite painful for day to day use.

While the DNS cache poisoning flaw has been mitigated, it has not been totally fixed. Changing the entropy from 16 bit to 32 bit only increases the amount of packets needed to hit the jackpot, thus making the attack more expensive but not impossible. Evgeniy Polyakov successfully poisoned a patched DNS server in less than 10 hours in lab environment. That is very different from the original time frame for the attack, which was just a few seconds but goes on to show that this vulnerability is still somewhat a concern.

What can we learn from this?

If we were to learn anything from Dan Kaminsky's presentation or the tools and ideas that it sparked off, it is that network traffic can and will be controlled by attackers. As security professionals we need to be less skeptic of new attacks where the attacker needs to be able to view the traffic being sent and received by the victim.

Wireless usage has become a business need and even if it is banished from the corporate network, people with sensitive material such as CEOs, CTOs and upper management will still use their work laptop at the hotel lobby. Maybe they just want to check their email through Outlook Web Access or check out the latest stock market news while they are away from the office.

Most small to medium businesses make use of an Internet Service Provider whose infrastructure may be vulnerable. It may be that DNS server has not yet been patched, the routers have a vulnerability or simply the DSL / Cable service itself. The worst part is for most of us that is no economically feasible way of avoiding these potential pitfalls. If the traffic can be monitored by malicious users most of the times it can also be modified and interrupted.

Once a packet leaves networks that we own there is no guarantee that anything in between cannot be malicious. Therefore it is important to stop assuming that network traffic cannot be intercepted or modified by malicious parties. This applies to both network and software designers. This should be part of the threat modeling of many modern systems. Automated updates especially fail in this aspect and tend to make the false assumption that an attacker cannot control what the client receives.

WEB BROWSERS ARE MAKING IT MORE DIFFICULT TO ACCEPT BAD WEBSITE CERTIFICATES

We should also be fixing our encryption systems. They should have been a good solution to the concerns raised by Dan Kaminsky and other security researchers. Services that require a certain level of security, such as online banking sites, should be avoiding access through websites without encryption and should be making sure that the session cookies are aware of the security context. Web browsers are making it more difficult to accept bad website certificates. Automated security updates should be signed at least to prevent Trojan updates.

Developers also need to take into consideration that signed updates may include old vulnerabilities, so some way of preventing old and vulnerable signed updates needs to be implemented.

In the end all this has to do with proper system design. Systems are typically made up of various components that if not adequately understood could all lead to compromise. When working on a new system, we need to understand what the real attacks can be for each element in the system and that obviously involves a bit of devil's advocate thinking, also known as Threat Modeling.

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

Network and information security in Europe today

by Mirko Zorz

Last week, the 1st NIS Summer School jointly organized by the European Network and Information Security Agency (ENISA) and the Institute of Computer Science of the Foundation for Research and Technology - Hellas (FORTH-ICS) took place in Heraklion, Greece.

The purpose of this gathering was to discuss multi-dimensional issues related to network and information security (NIS), the advances made in the recent past, along with emerging threats, critical compliance and legal issues. The attendees enjoyed the presentations of numerous outstanding speakers from all over the world.

ENISA representatives have a clear idea about the complexity of the problem they're dealing with. Rather than bombarding us with surveys, they simply say they don't know how big the problem is. Nobody does really, statistics differ and companies still under-report security breaches which makes it impossible to see the big picture. We can only accept the fact that we live in uncertainty but at the same time we need to get an understanding of the risks and vulnerabilities since that's the only way we can protect our networks.

It's worth noting that ENISA wants the mandatory reporting of security breaches despite this not being popular with all organizations.

Working together

One of the hot topics at the event was data protection. It's essential for an organization to set a clear set of goals if it wants to achieve an acceptable level of security. What organizations need to realize when discussing the question of security return on investment (ROI) is the fact that good regulation guarantees trust. Naturally, trust brings forward more users and eventually more services. Thus, it's of the essence to work on issues related to the regulatory framework.

Some member states of the European Union are more equipped than others when it comes to developing NIS. One of the roles of ENISA is to broker the way knowledge is exchanged between countries. Fine examples of cooperation are Hungary working with Bulgaria in setting up a government Computer Emergency Response Team (CERT) and Finland supporting Slovenia in organizing awareness raising activities.

You are probably wondering how effective ENISA's work is. A survey showed that the work is influential and of high quality, but it still has to reach its full potential. With a yearly budget of 8 million Euros and so much on their plate, the agency has to choose their research carefully.

Dr. Jorgo Chatzimarkakis, a Member of the European Parliament, emphasized the importance of having politicians acquainted with matters related to computer security. It was refreshing to hear a politician with a significant amount of IT knowledge discuss crucial security issues and their impact on the European Union.

IN ORDER TO ACHIEVE REGULATION, WE NEED GREATER RESPONSIBILITY FROM BOTH INDIVIDUALS AND THE PRIVATE SECTOR

The dark ages of security

Lord Toby Harris from the House of Lords, illustrated the problem with information security today as a poor relation of security and technology. The complication derives from a variety of emotional, cultural and financial issues. He is very critical of the UK government's approach to security on several levels and he's not afraid to demonstrate the topic. He believes there's a danger of complacency in the UK. The public sector compliance with security requirements is poor and a proper disaster recovery plan is nonexistent. Sadly, the same can probably be said for most European countries.

The fact of the matter is that in order to achieve regulation, we need greater responsibility from both individuals and the private sector. The balance of responsibility has to shift and include equipment manufacturers, software producers and service providers. Also essential are adequate resources that allow the enforcement of the rules.

One of the hot topics for privacy advocates in the UK is certainly that of national ID cards. Lord Harris demonstrated the erroneous way in which the government is "selling" them to the public. No, they won't be a good counter-terrorism tool and they offer limited benefits when it comes to illegal immigration and border control. However, they undoubtedly grant citizens the benefit of being able to establish their identity and entitlement. If an ID card was required to open a bank account, they would probably make the identity theft rate go down.

With the strong expansion of broadband and other communication technologies, identity and security matter more every day. People are being increasingly targeted by cyber crooks and they have plenty to worry about: e-crime, data loss and a plethora of malicious attacks.

When it comes to e-crime specifically, it's exceptionally problematic to display the magnitude of the problem in the UK since e-crime is still not recorded separately from other types of fraud. Despite not having concrete data at their disposal, UK citizens are more afraid of e-crime than burglary or mugging. According to Lord Harris, ignorance, carelessness and technology flaws are what puts individuals at risk. Once again we're reminded about the fundamental importance of security awareness.

Lord Harris believes that because of a grave lack of security, the UK critical network infrastructure is at risk. Let's just remind ourselves about the crippling May 2007 attacks in Estonia and the recent cyber disruption in Georgia.

Governments should have a framework that enables them to see which resources are being attacked and, clearly, a proper set of firm guidelines that make sure every system is up to date and working properly.

We are increasing relying on Internet services but, sadly, they are not dependable. The above-mentioned events have demonstrated the persistent threat of Distributed Denial of Service (DDoS) attacks as an effective instrument of cyber-warfare and they can certainly impact the end user. Overlay-based mechanisms can mitigate the impact of DDoS attacks and their impact on performance is relatively low. The problems that remain are awareness and implementation.

As we move to an intrinsically networked world, the possibility of witnessing terrorists using cyber warfare is growing every day. The question isn't "if"- it's "when". While such an attack may not result in lives being lost, the economic impact may be immense and create a variety of long-term consequences.

The importance of research

One of the principal areas of security research today deals with emerging risks. The motivation is simple - you want to prepare for the future and try to stay one step ahead of the attackers by anticipating what lies ahead. As the learning process improves your knowledge of the problem, you develop a culture of security and that's exactly what every organization should invest into.

By collecting a vast amount of information and applying the correct analysis metrics we can at least in some way anticipate what will drive future threats. We have to take into consideration the development of communication technologies, the evolution of hardware as well as other factors such as online services, the size of devices we use, smartphones, and more.

THE INTERNET IS COMPLICATED BECAUSE IT'S DYNAMIC BY NATURE. AS WE RELY ON THE INTERNET MORE EVERY DAY, WE HAVE TO INVEST RESOURCES INTO RESEARCH AND SECURITY ON ALL LEVELS

We live in a world where Web 2.0 applications are gaining momentum. As the Internet userbase grows we can easily foresee a massive adoption of online services.

Mobile phones are becoming more complex and able to perform a variety of tasks. With a generation of users that's doing things "on the go" right now, we're bound to see many more services on mobile devices in the future. All of these things have to be taken into consideration when trying to imagine the future.

Mikko H. Hyppönen, the Chief Research Officer at F-Secure, portrayed a dark picture of today's online world as he talked about gangs, terrorism, espionage, the hacker economy and how computer crime is the fastest growing segment of the IT industry.

Cyber thieves these days deal freely with credit card numbers, keyloggers, worms and Trojans. The Internet's dark side is thriving and there's a lot of money to be made. Unfortunately, the police is not doing much so the threat scenario keeps getting worse.

While today's issues such as phishing, identity theft and spam already pose a significant problem, the future will bring forward problems we still don't think about. Imagine an attacker breaching the security of your networked home and changing the settings on your alarm or the stove. Imagine a proliferation of nasty malware on Bluetooth and GSM networks. If you work in the information security world, I'm sure you can imagine a lot of dangerous complications.

Conclusion

The Internet is complicated because it's dynamic by nature. As we rely on the Internet more every day, we have to invest resources into research and security on all levels. Remember, information security is a journey, there are always new challenges.

What became evident to everyone attending the 1st NIS Summer School is what Dr. Jorgo Chatzimarkakis noted: "Network security is like oxygen - if you lose it, you realize its importance."

Browser security: bolt it on, then build it in

by Jeremiah Grossman

Whether improving ease-of-use, adding new developer APIs, or enhancing security – Web browser features are driven by market share. That's all there is to it. Product managers perform a delicate balancing act of attracting new users while trying not to "break the Web" or negatively impact their experience.

Some vendors attempt an über secure design - Opus Palladianum as an example, but few use it. Others opt for usability over security, such as Internet Explorer 6, which almost everyone used and was exploited as a result. Then, somewhere in the middle, is fan-favorite Firefox. The bottom line is that any highly necessary and desirable security feature that inhibits market adoption likely won't go into a release candidate of a major vendor. Better to be insecure and adopted instead of secure and obscure.

Fortunately, the major browser vendors have had security on the brain lately, which is a welcome change. Their new attitude might reflect the realization that a more secure product could in fact increase market share. The online environment is clearly more hostile than ever, as attackers mercilessly target browsers with exploits requiring no user intervention. One need only to look at this year's massive SQL Injection attacks that infected more than one million Web pages (tinyurl.com/6nthev), including those belonging to DHS, U.N., Sony, and others. The drive-by-download malware had just one goal - compromise the browser - with no interest in looting the potentially valuable data on the sites. Of course, we still have the garden-variety phishing sites out there. This leads to questions regarding the benefits of end-user education. Users are fed up. So let's analyze what the Mozilla and Microsoft camps have done in response.

Buffer overflows and other memory corruption issues in the most recent browsers are declining, plus the disclosure-to-patch timeline is trending properly. Firefox 3 and Internet Explorer 7 now offer URL blacklists that block phishing sites and other pages known to be delivering malware. These features are reportedly a little shaky, but it's clearly better considering there was nothing in place before. Firefox 3 provides additional visibility into the owners of SSL certificates and make it more challenging to blindly accept those that are invalid or self-signed. IE 7 offers a nice red/green anti-phishing toolbar that works with EV-SSL to help users steer clear of dangerous websites. Overall, excellent progress has been made from where we were just a couple years ago, but before the vendors start patting themselves on the back, there's also some bad news.

If you ask the average Web security expert if they think the typical user is able to protect themselves online and avoid getting hacked, the answer will be an unqualified "no". While browser vendors are addressing a small slice of a long-standing problem, most people are not aware of the remaining risks of a default install of the latest version of Firefox or Internet Explorer. When visiting any Web page, the site owner is easily able to ascertain what websites you've visited (CSS color hacks) or places you're logged-in (JavaScript errors / IMG loading behavior). They can also automatically exploit your online bank, social network, and webmail accounts (XSS). Additionally, the browser could be instructed to hack devices on the intranet, including DSL routers and printers. And, if that's not enough, they could turn you into a felon by forcing requests to illegal content or hack other sites (CSRF). The list goes on, but DNS-rebinding attacks get a little scary even for me, and it's not like we haven't known of these issues for years.

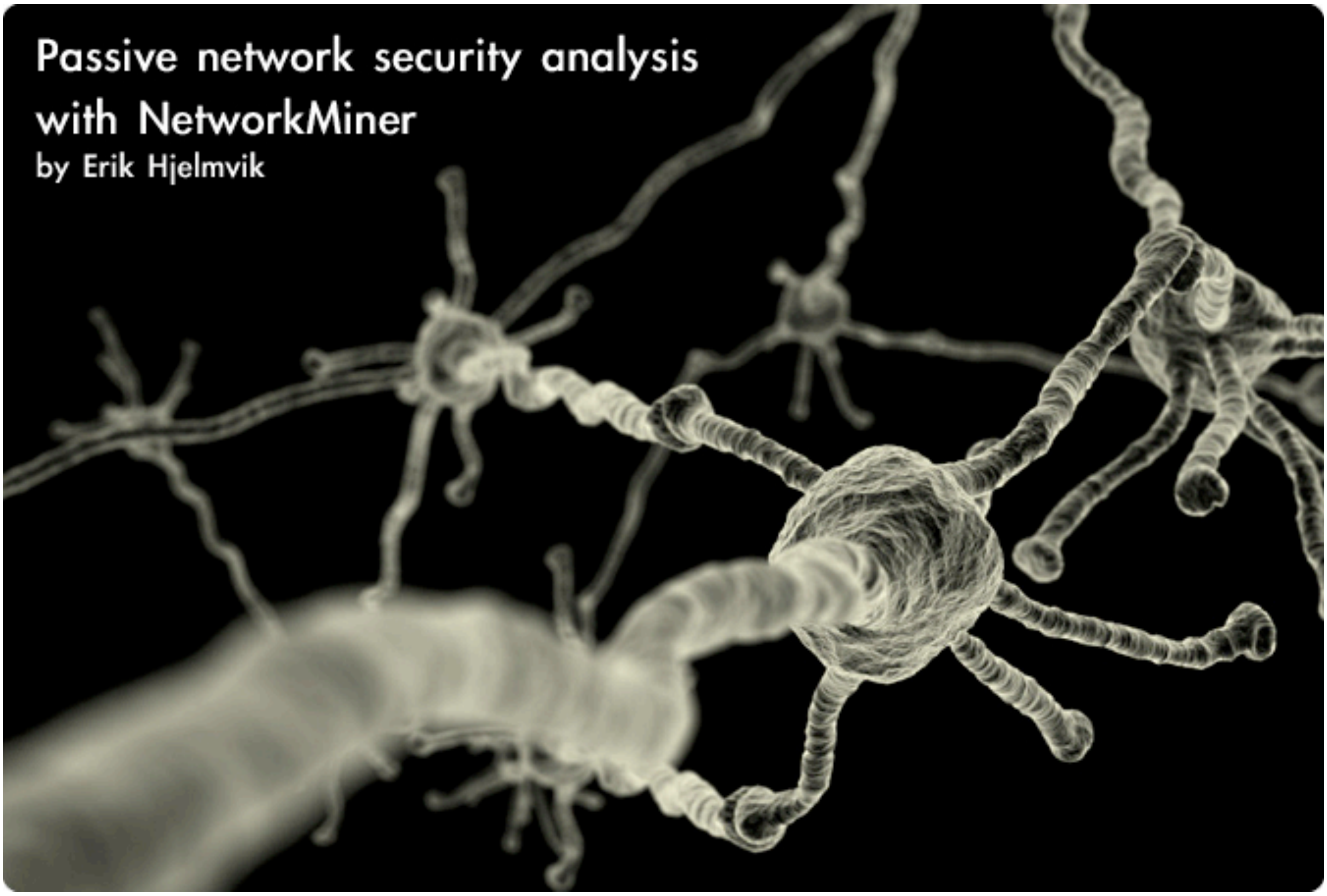
The browser security oxymoron hasn't escaped the watchful eyes of the media's Dan Goodin (tinyurl.com/6nsmtz) and Brian Krebs (tinyurl.com/4nhr4n), who figured out that something isn't quite right. Nor Robert "RSnake" Hansen (CEO, SecTheory), who is a little confused as to why organizations such as OWASP don't pay closer attention to browser security (tinyurl.com/5cutqo). According to sources, only about half of IE users are running the latest, most secure and stable version of the browser. And again, if you ask the experts how they protect themselves, you'll receive a laundry list of security add-ons, including NoScript, Flashblock, SafeHistory, Adblock Plus, LocalRodeo and CustomizeGoogle. Even with these installed, which relatively few people do, openings still exist resulting in an increasing number of people virtualizing their browsers or running them in pairs. Talk about extreme measures, but this is what it takes to protect yourself online.

Today, my philosophy about browser security and the responsibility of the vendors has changed. In my opinion, the last security-mile won't and can't be solved efficiently by the browser vendors, nor should we expect it to. I fully appreciate that their interests in building market share conflicts with those security features experts request, which by the way never ship fast enough. To be fair, there really is no way for browser vendors to make the appropriate amount of security for you, me, or everyone in the world while at the same time defending against all of the known cutting-edge attack techniques. Everyone's tolerance for risk is different. I need a high-level of browser security and I'm OK if that means limiting my online experience; but, for others that could be a non-starter. This leaves the door open for open source or commercial developers to fill in the gaps.

I was recently talking with RSnake about this and he said "I think the browser guys will kill any third party add-ons by implementing their own technology solution, but only when the problem becomes large enough." I think he's exactly right! In fact, this has already happened and will only continue. The anti-phishing toolbars were inspired directly from those previously offered by Netcraft and eBay. The much welcome XSSFilter built into the upcoming Internet Explorer 8 is strikingly reminiscent of the Firefox NoScript add-on. Mozilla is already adopting the model themselves by building their experimental Site Security Policy add-on (tinyurl.com/6j2ch6), which may one day work itself into a release candidate.

At the end of the day, the bad guys are going to continue winning the Web browser war until things get so bad that adding security add-ons will be the norm rather than the exception. Frankly, Web browsers aren't safe now, because they don't need to be. So, until things change, they won't be… secure.

Jeremiah Grossman, founder and chief technology officer of WhiteHat Security (www.whitehatsec.com), is a world-renowned expert in web application security and a founding member of the Web Application Security Consortium (www.webappsec.org).



Passive network security analysis with NetworkMiner

by Erik Hjelmvik

Whether working as a Network Penetration Tester, IT Security Auditor or Network Security Analyst, chances are you have spent time analyzing captured network traffic with applications such as Wireshark. Going through network traffic on a packet-by-packet or byte-per-byte level can be very powerful at times, but as the amount of captured traffic grows the need for more advanced analysis tools becomes apparent.

This article outlines the importance of analyzing captured network traffic and introduces an application called NetworkMiner, which is designed to support the IT security analysis by extracting useful information from captured data.

It is disturbing how often networks are not properly documented in terms of IP plans, network segmentations and network security. Having a good view of the network is essential when performing a network security assessment. As such, one might choose to perform an active network scan with a tool such as Nmap or Nessus in order to quickly gather inventory information of the hosts on a network.

Performing active scanning is, however, not very suitable for situations when the network is being used for operations of critical IT systems such as process control, radar, SCADA, or telecommunications systems. These types of critical IT systems always need to be in operation and scheduled service windows are very rare, so any active scanning should be avoided since it might affect the performance of the network or hosts on the network. Even the so-called “safe checks” in Nessus can cause critical IT systems to malfunction since these systems often are embedded systems running proprietary software with a high number of undiscovered vulnerabilities and bugs.

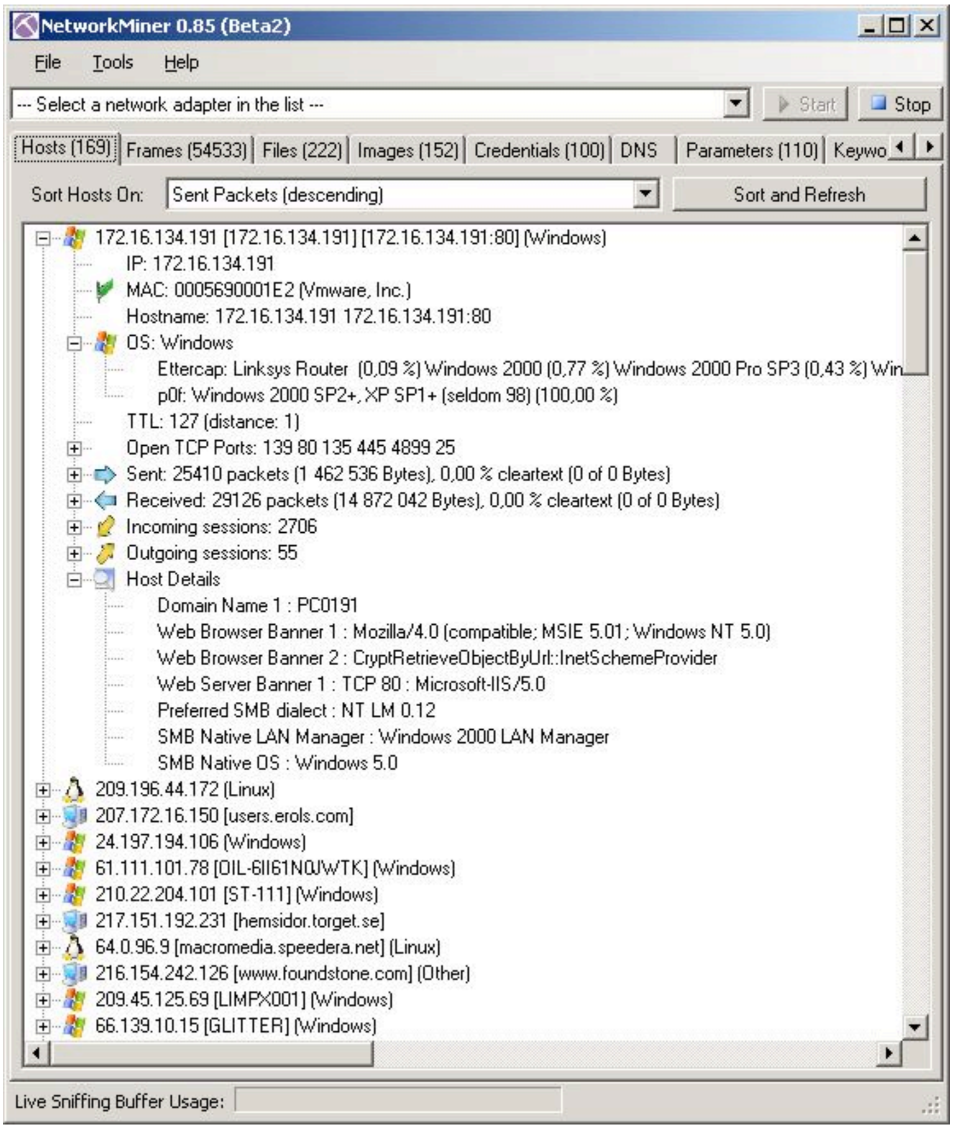
To avoid an emergency shutdown of a nuclear plant on which you might be performing your network security assessment, it is recommended that the analysis be based on passively captured network traffic from the system under investigation.

To passively capture traffic with focus on security is often referred to as “Network Security Monitoring” or “Packet Sniffing”; the less suitable term “Passive Scanning” is also used at times. Performing passive network analysis can be very useful also for non-critical IT systems such as normal business IT systems. One such example is when BlackBox internal penetration testing is performed since it is useful to enumerate hosts, services and protocols while remaining stealthy. Often during an internal penetration test, part of the test is to determine when the organization detects the ethical hacker on the network. The use of passive network analysis can therefore be helpful in the early phase of penetration testing so as to avoid detection as it reduces the need for an active portscan.

The network security tool that I will be relying on in this article is called NetworkMiner (sourceforge.net/projects/networkminer). It is an open source network forensic analysis tool (NFAT) that I developed.

Network discovery

Network traffic is best captured by connecting a packet sniffer to a network tap or monitor port of a switch located at a central point of a network or preferably at the perimeter between two different networks. Ideally, one should ensure that the machine which performs the monitoring cannot emit network traffic to the network being monitored. The packet sniffer can, for example, be a machine running tcpdump or Wireshark, which stores the captured traffic to a pcap file which can be processed later. There are also more comprehensive network monitoring solutions available such as Sguil, but that is beyond the scope of this article. You can, of course, use NetworkMiner to perform live sniffing of network traffic, but the recommended practice is to capture traffic to a pcap file with a purpose built sniffer and to subsequently perform offline analysis with a network forensic analysis tool. The pcap file can also be used as evidence if any illicit traffic is captured.



```
172.16.134.191 [172.16.134.191] [172.16.134.191:80] (Windows)
    IP: 172.16.134.191
    MAC: 0005690001E2 (Vmware, Inc.)
    Hostname: 172.16.134.191 172.16.134.191:80
    OS: Windows
        Ettercap: Linksys Router  (0,09 %) Windows 2000 (0,77 %) Windows 2000 Pro SP3 (0,43 %) Win
        p0f: Windows 2000 SP2+, XP SP1+ (seldom 98) (100,00 %)
    TTL: 127 (distance: 1)
    Open TCP Ports: 139 80 135 445 4899 25
    Sent: 25410 packets (1 462 536 Bytes), 0,00 % cleartext (0 of 0 Bytes)
    Received: 29126 packets (14 872 042 Bytes), 0,00 % cleartext (0 of 0 Bytes)
    Incoming sessions: 2706
    Outgoing sessions: 55
    Host Details
        Domain Name 1 : PC0191
        Web Browser Banner 1 : Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
        Web Browser Banner 2 : CryptRetrieveObjectByUrl::InetSchemeProvider
        Web Server Banner 1 : TCP 80 : Microsoft-IIS/5.0
        Preferred SMB dialect : NT LM 0.12
        SMB Native LAN Manager : Windows 2000 LAN Manager
        SMB Native OS : Windows 5.0
209.196.44.172 (Linux)
207.172.16.150 [users.erols.com]
24.197.194.106 (Windows)
61.111.101.78 [OIL-6II61N0JWTK] (Windows)
210.22.204.101 [ST-111] (Windows)
217.151.192.231 [hemsidor.torget.se]
64.0.96.9 [macromedia.speedera.net] (Linux)
216.154.242.126 [www.foundstone.com] (Other)
209.45.125.69 [LIMPX001] (Windows)
66.139.10.15 [GLITTER] (Windows)
```

I have used the publicly available pcap file "Scan of the Month 27" (sotm27), from The Honeynet Project (tinyurl.com/66jbz2), in order to demonstrate the strength of NetworkMiner in host discovery. When loading the sotm27 capture file into NetworkMiner, it generates an impressive list of 169 hosts together with the host names and the operating systems of the detected hosts. By expanding the nodes in the host list, details such as server banners, open ports and domain names can be displayed. Most of this information is easily retrieved directly from the captured network packets since protocols such as DNS, SMB and HTTP are implemented in NetworkMiner. Other information, such as operating systems, are determined by performing matching of specific fields from protocols such as TCP, IP and DHCP against databases from applications such as Ettercap, p0f and Satori.

A good approach for high security networks is to block all incoming and outgoing traffic except for the specific sessions (client-server-port combinations) which are expected and allowed to cross the network boundary. To create good and tight firewall rules, a network administrator needs to know which sessions are actually taking place across a network boundary. Luckily for the administrator, NetworkMiner provides a list of all incoming and outgoing sessions for each host, so monitoring the traffic between the two network segments is a good first step in creating better firewall rules.

Investigating potential rogue hosts

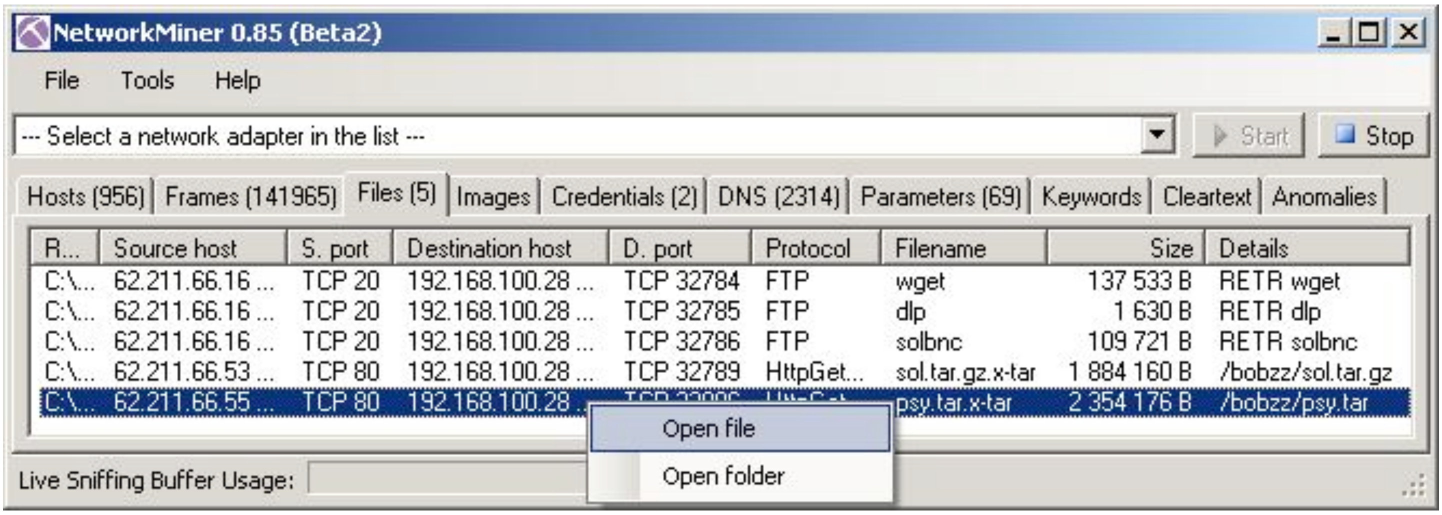
While looking at captured network traffic from a known network with NetworkMiner, new unknown hosts might show up as well as evidence indicating that a host has been compromised. Such hosts might be rogue hosts and should be handled with care. Common procedures might be to locate the rogue host in order to shut it down or remove it from the network, but it is often useful to monitor all the traffic to and from the host for awhile in order to get a better understanding of what is going on. The captured traffic can also provide forensic evidence that might be of value later on. An attacker might also be able to erase all log files and traces on the compromised host but would not be able to delete the captured traffic from your network monitoring system.

The network-based evidence might be the only evidence available for forensic analysis if you are dealing with a skilled attacker. If you do not have the possibility to monitor a host's traffic from the network, then another more primitive option is to actually log into the machine and perform the packet capturing locally. NetworkMiner supports this feature since it can be run directly from a USB thumbdrive and does not require installation. Monitoring hosts locally is, however, more suitable for troubleshooting and network discovery than it is for monitoring a compromised machine since you normally do not want to alter anything on the host being investigated.

Nowadays, a large amount of traffic is being sent through wireless networks, so be sure to monitor your airspace for rogue hosts and rogue access points that use IEEE 802.11 WiFi. Tools such as Kismet can be used to detect hosts and access points using WiFi, but unfortunately Kismet does not provide much information about the detected hosts. By loading Kismet capture files into NetworkMiner, or by performing live WiFi sniffing with NetworkMiner using an AirPcap device, you will be able to get the most out of your wireless monitoring.

Reassembling transferred files

NetworkMiner is also useful for reassembling and extracting files from captured network traffic. Examples of protocols from which NetworkMiner can perform file reassembly are HTTP, FTP and SMB. By loading the pcap files from The Honeynet Project's "Scan of the Month 28" (sotm28) (tinyurl.com/5quoav) into NetworkMiner you will be able to examine not only what the attacker did, but also the contents of the files he downloaded to the compromised machine. By selecting the "files" tab and right clicking a file you get a context menu which allows you to open the file or the parent folder. By looking at NetworkMiner's files tab after loading the pcap files from sotm28, one will see that after gaining control of the machine, the attacker started out by using ftp in order to download wget to the compromised machine. The attacker was then able to use wget to download other applications such as psyBNC, which often is used as a backdoor into a compromised machine or to allow someone to remotely control the machine as a



R...	Source host	S. port	Destination host	D. port	Protocol	Filename	Size	Details
C:\...	62.211.66.16 ...	TCP 20	192.168.100.28 ...	TCP 32784	FTP	wget	137 533 B	RETR wget
C:\...	62.211.66.16 ...	TCP 20	192.168.100.28 ...	TCP 32785	FTP	dlp	1 630 B	RETR dlp
C:\...	62.211.66.16 ...	TCP 20	192.168.100.28 ...	TCP 32786	FTP	solbnc	109 721 B	RETR solbnc
C:\...	62.211.66.53 ...	TCP 80	192.168.100.28 ...	TCP 32789	HttpGet...	sol.tar.gz.x-tar	1 884 160 B	/bobzz/sol.tar.gz
C:\...	62.211.66.55 ...	TCP 80	192.168.100.28 ...	[illegible]	[illegible]	psy.tar.x-tar	2 354 176 B	/bobzz/psy.tar

part of a botnet. The file reassembly functionality in NetworkMiner also allows you to view any webpage which has been retrieved across the monitored network. Therefore, by right-clicking an html file you will be able to open an offline version of that particular web page. Apart from the normal file transfer protocols, NetworkMiner is one of the few applications that also support reassembly of files transferred with the TFTP protocol. TFTP is a lightweight file transfer protocol that is often used by bootloaders of embedded systems in order to retrieve executable firmware images (such as a kernel and a file system) from a remote server. The TFTP protocol might be used by an attacker to replace the firmware of your printers, routers, switches, WiFi access points and even firewalls with a special purpose built firmware. This firmware might, for example, be designed to monitor your network traffic and report data such as captured user credentials to the attacker. This implies that you should not fully trust your firewalls unless you have the ability to see which traffic is entering and leaving your firewall.

By monitoring the network traffic to and from the embedded systems on your network, you actually have the possibility to see if they are acting as expected; you would, for example, not expect your printers to post files to an external FTP server, would you?

If you monitor the traffic that leaves your network you will be able to see what information is being exposed to external non-trusted parties. NetworkMiner also has a keyword search functionality that allows you to search all traffic (regardless of protocol) for keywords such as “confidential”.

Data leakage and data seepage

Another use of NetworkMiner is in evaluating how much data, regarding you and your computer, is being disclosed to the network without your knowledge. By connecting your laptop to an unknown network or unencrypted WiFi access point you make this data available to any malicious lurker who might be sniffing that particular network. Not only might the lurker be able to read your emails and see your passwords, he may also be able to identify your previous IP address and to see which file servers you have network shares on. This type of information is called “Data Seepage” and can be used by an attacker to gain useful information in order to, for example, plan an attack. By launching NetworkMiner locally on your own machine, you will be able to see what information your computer is leaking to potentially malicious network-lurkers who might be performing Man-in-the-Middle or WiFi sniffing. After using NetworkMiner, you will soon learn that connecting your computer into an unknown network (wired or wireless) cannot be considered “safe sex”; so be sure to use protection if you wish to connect your Ethernet cable to a non-trusted RJ45 socket.

Erik Hjelmvik is an independent network security researcher and open source developer. He also works as a software development consultant, specialising in embedded systems. In the past, Erik served as an R&D engineer at one of Europe’s largest electric utility companies, where he worked with IT security for SCADA and process control systems.

Latest additions to our bookshelf

Ajax Security

By Billy Hoffman and Bryan Sullivan

Addison-Wesley Professional, ISBN: 0321491939

Ajax Security systematically debunks today's most dangerous myths about Ajax security, illustrating key points with detailed case studies of actual exploited Ajax vulnerabilities, ranging from MySpace's Samy worm to MacWorld's conference code validator. Even more important, it delivers specific, up-to-the-minute recommendations for securing Ajax applications in each major Web programming language and environment, including .NET, Java, PHP, and even Ruby on Rails.

Big Book of Apple Hacks

By Chris Seibold

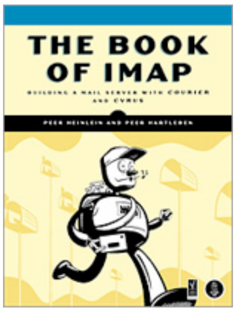
O'Reilly, ISBN: 0596529821

The new Big Book of Apple Hacks offers a grab bag of tips, tricks and hacks to get the most out of Mac OS X Leopard, as well as the new line of iPods, iPhone, and Apple TV. With 125 entirely new hacks presented in step-by-step fashion, this practical book is for serious Apple computer and gadget users who really want to take control of these systems. Many of the hacks take you under the hood and show you how to tweak system preferences, alter or add keyboard shortcuts, mount drives and devices, and generally do things with your operating system and gadgets that Apple doesn't expect you to do.

The Book of IMAP: Building a Mail Server with Courier and Cyrus

By Peer Heinlein and Peer Hartleben

No Starch Press, ISBN: 1593271778

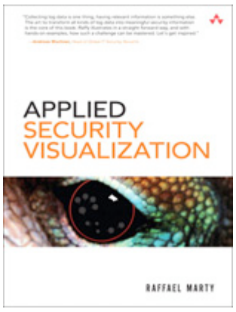


IMAP (the Internet Message Access Protocol) allows clients to access their email on a remote server, whether from the office, a remote location, or a cell phone or other device. IMAP is powerful and flexible, but it's also complicated to set up; it's more difficult to implement than POP3 and more error-prone for both client and server. The Book of IMAP offers a detailed introduction to IMAP and POP3, the two protocols that govern all modern mail servers and clients. You'll learn how the protocols work as well as how to install, configure, and maintain the two most popular open source mail systems, Courier and Cyrus.

Applied Security Visualization

By Raffael Marty

Addison-Wesley Professional, ISBN: 0321510100

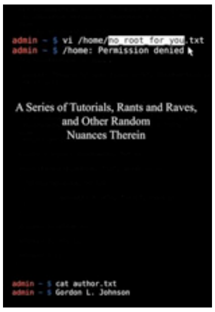


In this book, leading network security visualization expert Raffael Marty introduces all the concepts, techniques, and tools you need to use visualization on your network. You'll learn how to identify and utilize the right data sources, then transform your data into visuals that reveal what you really need to know. Next, Marty shows how to use visualization to perform broad network security analyses, assess specific threats, and even improve business compliance. He concludes with an introduction to a broad set of visualization tools. The book's CD also includes DAVIX, a compilation of freely available tools for security visualization.

No Root For You

By Gordon L. Johnson

Wordclay, ISBN: 1604811862

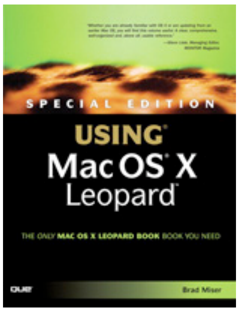


This is a network auditor's quick-reference bible. Not only does it contain step-by-step, illustrated tutorials, but an explanation in regards to why each exploitation, or what have you, works, and how to defend against such attacks. Be prepared, one might also discover a few "rants and raves," as well as other random nuances.

Special Edition Using Mac OS X Leopard

By Brad Miser

Que, ISBN: 0789736535

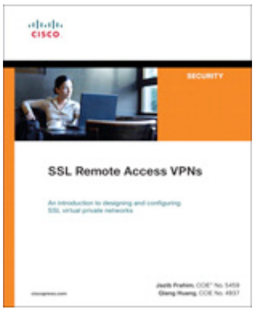


Explore the depths of Mac OS X's core including the desktop, Finder, the Dock, user accounts, the Dashboard and widgets, Spaces, and much more. Master OS X by installing and using Mac OS X applications, customizing the system, making your Mac accessible to everyone, automating your Mac with the Automator, using Unix commands, and working with mobile Macs. Run Windows applications on your Mac for those rare occasions when a Mac application isn't available. Use great Mac OS X tools and techniques to keep your system in top condition and to solve problems.

SSL Remote Access VPNs

By Qiang Huang and Jazib Frahim

Cisco Press, ISBN: 1587052423



SSL Remote Access VPNs provides you with a basic working knowledge of SSL virtual private networks on Cisco SSL VPN-capable devices. Design guidance is provided to assist you in implementing SSL VPN in existing network infrastructures. This includes examining existing hardware and software to determine whether they are SSL VPN capable, providing design recommendations, and guiding you on setting up the Cisco SSL VPN devices. Common deployment scenarios are covered to assist you in deploying an SSL VPN in your network.

Your Brain: The Missing Manual

By Matthew MacDonald

O’Reilly, ISBN: 0596517785

This is a book about that wet mass of cell tissue called the brain, and why it's responsible for everything from true love to getting you out of bed in the morning.

One part science guide, one part self-help concierge, it's grounded in the latest neuroscience, psychology, and nutritional wisdom. The result? An essential guide for the modern brain owner, filled with ready-to-follow advice on everything from eating right to improving your memory.

Cisco Firewall Video Mentor

By David Hucaby

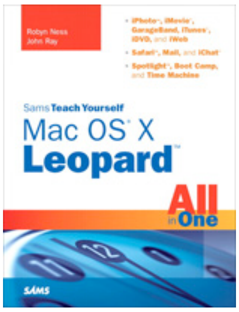
Cisco Press, ISBN: 1587201984

Cisco Firewall Video Mentor is a unique video product that provides you with more than five hours of personal visual instruction from best-selling author and lead network engineer David Hucaby. In the 16 videos presented on the DVD, David walks you through common Cisco firewall configuration and troubleshooting tasks. Designed to develop and enhance hands-on skills, each 10–30 minute video guides you through essential configuration tasks on the Cisco ASA and FWSM platforms and shows you how to verify that firewalls are working correctly.

Sams Teach Yourself Mac OS X Leopard All in One

By Robyn Ness and John Ray

Sams, ISBN: 0672329581



You’ve got a Mac. You’ve got Leopard. Now all you need to do is figure out how to get them to work together so that you can stop thinking about your computer and start thinking about getting things done, having fun, and enjoying everything your Mac has to offer.

This one book is your answer to basically any questions you might have today, and the answer to all the questions about Leopard and your Mac that you’re likely to have in the future.

HELP NET SECURITY

GFiMailEssentials for Exchange/SMTP/Lotus

Download a free trial

The #1 server based anti-spam solution!

Reputation attacks: A little known Internet threat

HOME NEWS ARTICLES SOFTWARE VIDEOS VULNS EVENTS NEWSLETTERS STORE ABOUT ADVERTISE SEARCH RSS

SUBSCRIBE BY E-MAIL

NEWS ARTICLES THE WIRE

- Spam served as visual poetry
- IT security spending on the rise
- Aladdin Knowledge Systems acquires SafeWord product line
- Secure Computing to acquire Securify
- The RFID Security Alliance (RFIDSA) formed

GRAB OUR RSS FEED

Reverse Engineer's Swiss Army Knife

net-security.org

0:00 / 0:00

REVIEWS MALWARE

- Security Power Tools
- Network Warrior
- Google Apps Hacks
- Crimeware: Understanding New Attacks and Defenses
- Big Book of Windows Hacks

WINDOWS LINUX OS X

- LANguard Network Security Scanner 8
- Password Safe 3.14
- Cute Password Manager Pro 2008.1.3.8
- Data Guardian 1.5.1
- CryptoExpert 2008 Professional 7.4.5
- Cain & Abel 4.9.20
- Outpost Firewall Pro 6.5.2358.316.0607
- AVG Anti-Virus Free Edition 8.138.1332

ADVISORIES VULNERABILITIES

- Mandriva Linux Security Update Advisory - python (MDVSA-2008:186)
- Gentoo Linux Security Advisory - MySQL: Privilege bypass (GLSA 200809-04)
- Gentoo Linux Security Advisory - RealPlayer: Buffer overflow (GLSA 200809-03)

Discovery and fuzzing for SQL injections with Web 2.0 applications

Wednesday, 3 September 2008, 7:26 PM CET

This paper describes some techniques and approaches to perform effective assessment on Web 2.0 applications on the basis of our recent experience and cases which were analyzed on the field.

Rootkit evolution

Monday, 1 September 2008, 11:08 PM CET

Rootkit evolution is following the same path as spyware. First, rootkits were identified as a separate class of malware. Then there was a lot of media hype which led to a large number of anti-rootkit tools and products together with a noticeable reaction from the antivirus industry. Today both rootkits and spyware have merged into the general malware stream and no longer cause any particular excitement. However, the concept of evading system features to hide something is obviously still valid and we are very likely to see new threats implementing stealth.

Deploying enterprise software securely

Wednesday, 27 August 2008, 11:37 PM CET

This laundry list of security requirements is a lot to think about for every application deployment, but vigilance in this area can drastically improve an organization's security posture. The requirements can be put into a standardized template, and at the end of the process each requirement should have a mark for pass, fail, or perhaps not applicable. Anything marked as a failure should be noted and can be escalated or accepted as a risk.

Most organizations fail to stop interior network threats

Tuesday, 26 August 2008, 9:25 PM CET

A survey by Opine Consulting revealed nearly half of the IT professionals who responded had endpoints connecting to their corporate networks without their knowledge. Yet compared to other security issues, 86 percent of respondents said controlling network access ranked as a high priority.

Security risks for mobile computing on public WLANs

Monday, 25 August 2008, 11:01 PM CET

This article illuminates the effectiveness of VPN security mechanisms, data encryption, strong authentication and personal firewalls and shows how optimal protection can be achieved by dynamically integrating each of these technologies.

Reverse engineering: Smashing the signature

Wednesday, 20 August 2008, 3:36 PM CET

Many antivirus and antispyware solutions identify malicious programs by looking for known unique signatures contained inside them. Those signatures are stored inside a database which is constantly updated. This tutorial guides you through a number of steps to encrypt the executable file code section in order to render antivirus signature checking techniques ineffective against identifying the malicious code.

Internet terrorist: Does such a thing really exist?

Tuesday, 19 August 2008, 3:25 PM CET

In this article, a former CISO discusses the notion of worrying about the potential risk of terrorism against his organization and how it seems to be the lowest priority given the choices at hand. Ironically, terrorism today seems to be an emerging concern in the commercial world and many are actively pursuing methods and technology to help combat the problem. As a result, he began to research this trend to determine its drivers and potential implications to information security as we know it today.

Reputation attacks: A little known Internet threat

Monday, 18 August 2008, 6:09 PM CET

Reputation attacks target both individuals and companies, and their goal is to ruin the victim's reputation. While attack techniques are varied, the consequences are often the same: a damaged reputation resulting in many cases in financial loss. Attackers can use several methods to ruin a company's reputation.

Qualys: Free Webcast: Proactive Vulnerability Management detailing key aspects, analysis and implementation!

Hard Drive Recovery

QUALYSGUARD®

On Demand

Vulnerability Management

Policy Compliance

PCI Compliance

14 Day Trial>

GFiEventsManager

Download your FREE trial today!

User attempted access to /financesales/

Failed to install Windows security updates

(IN)SECURE

FREE SECURITY MAGAZINE

DOWNLOAD HERE!

Ads by Google

Secure Email

Secure, reliable, fast, virus/spam. No setup fees, free 1800 support.

www.CryptoHeaven.com

SPONSORED LINKS

Vulnerability Scanning

Network Vulnerabilities

Event Log Security

Home Security

Security software Australia

Security Cameras

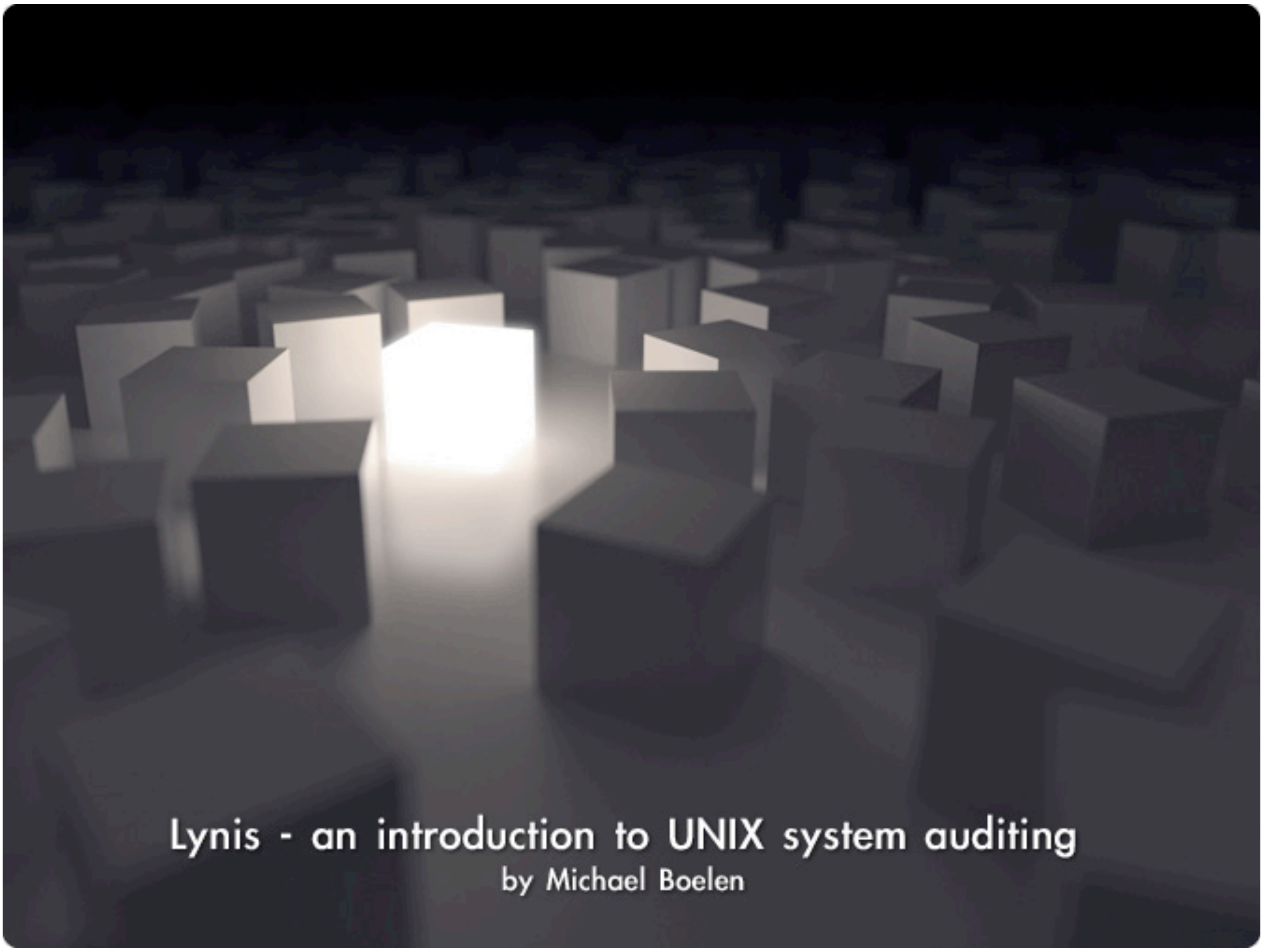
Keylogger and Parental Control

Find Electronic Component

Network Security System

online backup services

Internet Surveillance Cameras



Lynis - an introduction to UNIX system auditing

by Michael Boelen

Most people in the IT industry try to avoid working on a system 'designed' by others, knowing correct file permissions may not be set, applications may be patched several times (or not at all) and documentation is nowhere to be seen. With the proper tools and some administration experience, there hides an auditor in most of us.

When reviewing an unfamiliar system, a lot of commonly asked questions will pass by. Questions like, when was the system installed? What patch level is it currently at? Are the software and data correctly separated? Which dependencies does the system have? Who has access to the system and which programs need to be running to operate it correctly?

When working with a variety of Unix systems and versions, these simple questions become hard to answer very quickly. It could take several hours, or even days to answer all of them. Hence, when it's regarding a few hundreds machines, things really can get time consuming. As curious as we humans are, we rather pay attention to the interesting things and forget about the repeating tasks. This is where Lynis, a new auditing tool, comes into play.

Lynis

Open-source software is often created as an opposite of commercial software. Sometimes as an alternative to existing tools in the field (specific platform support only, not friendly to use, unmaintained) and sometimes as a personal home grown project. With the need to validate personal administration work and that of others, Lynis (rootkit.nl/projects/lynis.html) was born.

Lynis is a small program to assist in performing automated Unix system audits. It is free to use and available under the GPL license.

Its goals are to gather generic system information, check software and patch levels, test configuration file consistency and determine possible security weaknesses. Testing is host based, which makes it different from the existing network based auditing and pen testing tools. Due to local system access, Lynis can obtain additional information which can be combined with information of other tools.

When using several powerful tools together it will be much easier to maintain and control policies on new and existing hosts on the network. Lynis is intended for system/networks administrators, security specialists, auditors and penetration testers.

Lynis is written in shell script and therefore it can be directly run on most UNIX-based systems. The program code consists mainly of system and software tests, which gather information and check for configuration inconsistencies. The remaining code gives the program generic support like OS detection, easy to read screen output, logging and support for profiles. The amount of system and software tests performed will vary on the operating system version and especially the installed software on the system.

Some examples of included tests:

- Reading network information
- Configuration consistency checks
- Expired SSL certificates
- Outdated or vulnerable software
- Logging availability and time synchronization
- User accounts lacking password protection
- Incorrect file permissions on configuration files
- Firewall configuration and rules.

Due to the open-source nature of the project, input comes from a broad audience (with different technical skill sets, but also the personal nice-to-have wishes people tend to have).

Some of the features and strengths of project include:

- Easy to use command switches to customize an audit
- Colored output and clear overview of the warnings/problems
- Option to install as package or directly run from external media
- Easy to parse report file, with warnings, suggestions and system information
- Customizable scan profile support and personal tests
- Detailed log file as addition to screen output
- Easy to upgrade.

Lynis is intended for system/networks administrators, security specialists, auditors and penetration testers.

The auditing process

Auditing is not just simply a matter of sitting in your chair, watching fancy programs doing their thing. Even with simply audits, like a typical LAMP server, the end result depends a lot on the work of the auditor. Most audit reports say something about the weaknesses of the system, but also about the qualities of the auditor. Using a good toolkit is a good thing, working with proven methods and standards is even better.

When performing an audit it is good practice to write down your findings. Write down all time-stamps, what programs are executed (and why) and related file locations which are being checked. These notes can help later to create an advisory report, a hardening checklist or a logbook for future reference. It will also give you proof of what you did and when.

What if Murphy shows up and crashes some application while you performed an audit?

Without a clear log it won't take much time to get some finger pointing started. Last but not least, writing down things will help in improving personal audit skills. The more practice, the more information can be re-used for future audits. When preparing the audit, try to draw the boundaries of what should be included the audit (and what not). This will help to eliminate gathering too much useless information and reduces the time to perform the audit.

Another point which gets overlooked often is the required permissions to perform an audit. Not only the technical permissions to access systems, but also the formal permission which grants you the access to networks and systems and give you the available time (frames) to scan. After all, a well performed audit will take some time due the technical details involved. Report writing and sometimes after care (advising, assisting or even implementing the advisories) will add time.

Then there is the possible risk involved with auditing. Although audits should have a minimized risk to disrupt processes, there is always the risk to get an unexpected result. Things like accidentally overloading the system with requests, a malformed binary getting executed or simply a badly written program which suddenly hangs up the host.

To sum up some of the auditing prerequisites: make notes to track all steps, check your backups (for a possible needed recovery) and arrange technical permissions and the formal authorizations.

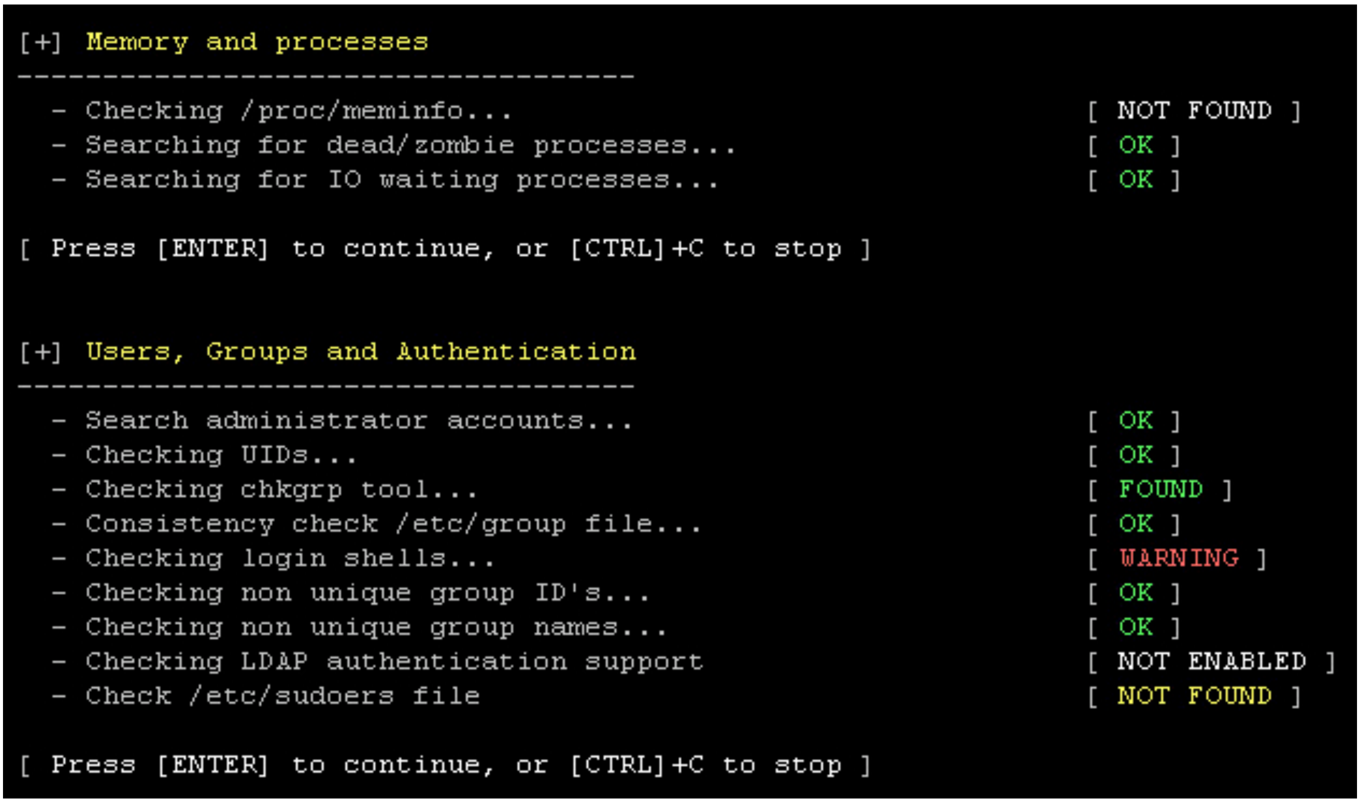
Although audits should have a minimized risk to disrupt processes, there is always the risk to get an unexpected result.

Example scan

As mentioned earlier, Lynis can be run directly after unpacking (or from external media). Only the parameter "-c" (or --checkall) is needed to start a full audit, other options are optional. The default audit profile will be appropriate for most systems, until customization is needed. Depending on the system speed and the amount to be scanned, an audit will most likely be finished within several minutes. Real life example:

- Friend's machine, friend worried about his system
- Operating system FreeBSD 5.4
- Security patches were not installed for some time (the main reason to worry)
- Report was not needed.

Before running Lynis, the system backups were checked and a “ps” listing was saved. Then Lynis was downloaded and executed. After that the scan showed per section what was scanned and revealed the test results. Below is an example screen output:



```
[+] Memory and processes
------------------------------------
  - Checking /proc/meminfo...                                   [ NOT FOUND ]
  - Searching for dead/zombie processes...                      [ OK ]
  - Searching for IO waiting processes...                       [ OK ]

[ Press [ENTER] to continue, or [CTRL]+C to stop ]


[+] Users, Groups and Authentication
------------------------------------
  - Search administrator accounts...                            [ OK ]
  - Checking UIDs...                                            [ OK ]
  - Checking chkgrp tool...                                     [ FOUND ]
  - Consistency check /etc/group file...                        [ OK ]
  - Checking login shells...                                    [ WARNING ]
  - Checking non unique group ID's...                           [ OK ]
  - Checking non unique group names...                          [ OK ]
  - Checking LDAP authentication support                        [ NOT ENABLED ]
  - Check /etc/sudoers file                                     [ NOT FOUND ]

[ Press [ENTER] to continue, or [CTRL]+C to stop ]
```

After a few minutes the audit was finished. The scan report was displayed, including some of the following warnings:

```
[09:10:27] Warning: Found probably incorrect installed package (glib) [test:PKGS-7303] [impact:M]
[09:10:54] Warning: Found one or more vulnerable packages. [test:PKGS-7382] [impact:M]
[09:11:01] Warning: Found SSL certificate expiration (/etc/ssl/ca.crt) [test:CRYP-7902] [impact:M]
```

The package database was checked directly after the audit. Ports were updated and remaining time went to executing the suggestions and cleaning up unused files. Afterwards the system looked much better.

Future additions

Lynis is easy to extend and adapt to your specific needs, in the form of custom checks and profile options. Since a lot of changes are useful for the community, requests are often implemented in new releases. While Lynis is improved piece by piece, some of the future additions to the tool can be already revealed:

Event correlation interface

Host based auditing is interesting when having to deal with a few machines. But when things get bigger and more powerful, they usually should be automated. One of the upcoming features is a web interface which gathers the audit reports and correlates them into smaller pieces of information. This way lots of systems can be checked within minutes instead of days. Customizable overviews show which systems lack security patches, need proper hardening or have security weaknesses.

Profile creation wizard and system profiler

Due a growing amount of options and customization possibilities, managing a big amount of profiles for different system types could become time consuming. To make it easier to customize or create profiles, there will be a wizard to handle this job. A second addition will be extending Lynis to use a predefined system as baseline and create automatically a profile after an audit has been performed.

Specialized audit types

With the upcoming support for plugins it will be easy to adjust the type of an audit and let Lynis perform one or multiple specific scan types. This gives the auditor the opportunity to use Lynis for example as an malware scanner (backdoors, rootkits, exploit code), a forensics tool or a system hardening tool/advisor.

More tests

The tests within Lynis make the tool to what it is currently. New tests will be added and start to focus more and more on applications, configuration issues and security.

Interesting reads

If you want more information about auditing, have a look at csrc.nist.gov/publications/ and www.sans.org/reading_room/. These pages contain a lot of auditing related information, including testing methodologies, checklists and reporting examples. If you would like to know more about Lynis, visit the project page, which can be found at rootkit.nl/projects/lynis.html.

Michael Boelen is a UNIX engineer at Snow B.V., a Dutch consultancy company with network and UNIX specialists (www.snow.nl). His main interests are Unix, security, auditing and forensics.

As the author of Rootkit Hunter and Lynis, Michael is a supporter of open-source software solutions. In his spare time he enjoys mountain biking, maintaining the project pages at rootkit.nl and reading technical books, blogs or websites.

Windows driver vulnerabilities: the METHOD_NEITHER odyssey

by Anibal Sacco

Device drivers are a fundamental piece of the windows model. They let you interact with hardware or perform operations in kernel mode. By exposing an interface to user mode a user mode process can establish a communication channel with a driver in order to send and receive data in a predefined way.

Lately new driver vulnerabilities are being reported day after day. This is nothing new, there have always been vulnerabilities in drivers, it is just that very few people were looking for them.

There are much less programmers dedicated to the development of drivers and ring 0 software than user-mode software. And it is understandable, it's not an easy task. For a long time the official documentation was very incomplete. The community used to hide their findings, yet a lot of functionality was discovered and documented by the community reversing the windows binaries first, and looking at the leaked sources later.

For that reason, for a long time (and maybe even today) the controls applied to the windows driver development were focused on making them stable and reliable, leaving aside sometimes some very basic security checks.

The windows drivers are exposed to vulnerabilities as any normal process executed in user-mode like MS Word, MS Messenger or even the Calculator. The difference relies on the execution privileges obtained by exploiting a vulnerability in a Ring 0 process, which implies the execution with the maximum privileges, giving the attacker the possibility to control or crash the whole system.

In this article, I will try to do a short introduction to the communication channel established by the windows drivers so that I can explain how to face a common vulnerability these drivers are exposed to, due to their specific design characteristics. Also, I will explain one of the possible attack vectors to get code execution through this kind of vulnerabilities.

Driver's structure

Unlike the normal user-mode process, the drivers don't make use of all its functionality executing it linearly. Normally they are composed of a main DriverEntry() function, usually compared with the library's DLLMain concept because it is mapped on memory just when the driver is loaded, but is only executed once, when the OS loads the module.

In this function, in the simplest scenario, the code in charge of the driver initialization is located. It performs tasks such as creating a symbolic link (to facilitate the way in which the user-mode process opens a handle for it) and the initialization of the "Function Dispatch Table", which is a table of pointers contained in the DRIVER_OBJECT structure that are basically used to expose the true functionality of the driver. These pointers will be invoked from the user-mode process through the IOManager, to execute the desired code in kernel mode.

The DRIVER_OBJECT

Each driver, when loaded, is represented by a kernel data structure called DRIVER_OBJECT. A pointer to the driver object is one of the input parameters to a driver's DriverEntry and is initialized when DriverEntry is called.

This is the structure:

```
typedef struct _DRIVER_OBJECT
{
     SHORT Type;
     SHORT Size;
     PDEVICE_OBJECT DeviceObject;
     ULONG Flags;
     PVOID DriverStart;
     ULONG DriverSize;
     PVOID DriverSection;
     PDRIVER_EXTENSION DriverExtension;
     UNICODE_STRING DriverName;
     PUNICODE_STRING HardwareDatabase;
     PFAST_IO_DISPATCH FastIoDispatch;
     LONG * DriverInit;
     PVOID DriverStartIo;
     PVOID DriverUnload;
     LONG * MajorFunction[28];
} DRIVER_OBJECT, *PDRIVER_OBJECT;
```

One of its fields, the MajorFunction array pointer, is initialized by the driver making, it point to its own functions. This is a very important field because these functions will be called by the IO Manager; which will depend on the kind of IRP request made from user-mode.

For example, when closing a driver with the CloseFile() API, the function pointed by MajorFunction[IRP_MJ_CLOSE] will be called.

The IRPs

From MSDN: "The Microsoft Windows family of operating systems communicates with drivers by sending input/output (I/O) request packets (IRPs). The data structure that encapsulates the IRP not only describes an I/O request, but also maintains information about the status of the request as it passes through the drivers that handle it. Because the data structure serves two purposes, an IRP can be defined as:

- A container for an I/O request, or
- A thread-independent call stack.

In this case, we are talking about the former.

The way in which a user-mode process can communicate with the device is through requests. These requests 'tell' the driver which function of the MajorFunction array pointer must be called and, if necessary, manages the buffers used to send and receive data.
These requests are called IRP Major requests.

The IOCTLs (or IRP_MJ_DEVICE_CONTROL) request

This is a key request, because it is used to send and receive data to, and from, the driver through DeviceIoControl. This is its prototype:

```
BOOL WINAPI DeviceIoControl(
      __in          HANDLE hDevice,
      __in          DWORD dwIoControlCode,
      __in_opt      LPVOID lpInBuffer,
      __in          DWORD nInBufferSize,
      __out_opt     LPVOID lpOutBuffer,
      __in          DWORD nOutBufferSize,
      __out_opt     LPDWORD lpBytesReturned,
      __inout_opt   LPOVERLAPPED lpOverlapped
    );
```

When the DeviceIoControl function is called from user-mode with the handle of an open driver, the function defined at MajorFunction[IRP_MJ_DEVICE_CONTROL] will be called with a pointer to the IRP object passed as an argument.

This function will receive through this structure important data such as the input buffer, output buffer, and its corresponding lengths. But depending on the defined method, the buffers could be handled in different ways by the IOManager.

From Microsoft knowledge Base Q115758:

"A driver can use one of the three different I/O methods: "buffered," "direct," or "neither." After you use a kernel-mode driver to create a device object, you can specify in the Flags field of the device object which I/O method you want it to use. One of the two values, DO_BUFFERED_IO or DO_DIRECT_IO, can be OR'ed into the flag field. Or you can choose not to specify either method in the flag field. In this case, we will say the driver has chosen the "neither" method. The method selected in the Flags field affects I/O read or write requests dispatched to this device object through the driver's Read or Write dispatch routines."

The method relevant for us is METHOD_NEITHER. This method will be used by the IOManager when the last XXX bits are turned on, and is especially problematic because, unlike the others (where the IOManager manages the buffers, and safely brings to the driver a kernel-allocated intermediate buffer to write, and read in) the IOManager does not touch or check the buffers in any way. It just passes the user-mode buffer pointers to the driver function through DeviceIocontrol, leaving it with all the responsibility of doing the necessary checks before accessing them.

The vulnerability

Leaving aside the chosen method for the request between user-mode and kernel-mode, in general lines, the mechanism is always the same:

- The user-mode process opens a handle to access the driver.
- Sends an IOCTL through DeviceIoControl, with some data in the input buffer and specifies the output buffer.
- The driver receives the ioctl, and, depending on the data on Inputbuffer, does some operations and returns data to Output buffer.
- The user-mode process receives the data and keeps running.

The problems with this method arise when the driver implements lazy checks (or none at all) when validating the pointers received from user-mode. If it is not done properly, the driver will try to retrieve the data in the output buffer, writing directly to the memory pointed by the user-mode process, and depending on the address sent, it could be exposed to write to an invalid memory address generating a Blue Screen of Death (BSOD) or it could be used, as will explain below, to modify certain kernel-mode structures that could allow the unprivileged user mode process to execute code in ring 0, in order to elevate privileges.

What usually differs between different scenarios is *what* the driver will try to return in the output buffer, but I could ensure that in most cases the *what* is not so important. With the possibility to write and modify kernel memory with a predictable value and just a bit of imagination (and some voodoo magic sometimes) the bug can be levered to obtain code execution.

To get a clear sight of the issue explained, I will use as example the vulnerability (CVE-2007-5756) reported on the Winpcap 4.x driver (a software package that facilitates real-time link-level network access for Windows-based operating systems).

We can see below the main driver routine and, as I've explained before, this contains the instructions to initialize the MajorFunctions array pointer with the driver functions. The most important line here, for our purposes, is the initialization of the IRP_MJ_DEVICE_CONTROL entry, telling us that the NPF_IoControl will be the function used to handle the IOCTLs received from user-mode.

```
NTSTATUS DriverEntry( IN PDRIVER_OBJECT DriverObject, IN PUNICODE_STRING
RegistryPath)
        {
        ...
            // Set up the device driver entry points.
            DriverObject->MajorFunction[IRP_MJ_CREATE] = NPF_Open;
            DriverObject->MajorFunction[IRP_MJ_CLOSE]  = NPF_Close;
            DriverObject->MajorFunction[IRP_MJ_READ]   = NPF_Read;
            DriverObject->MajorFunction[IRP_MJ_WRITE]  = NPF_Write;
            DriverObject->MajorFunction[IRP_MJ_DEVICE_CONTROL]  =
NPF_IoControl;
            DriverObject->DriverUnload = NPF_Unload;
```

This is the the vulnerable function. Let's see its code:

```
        NTSTATUS NPF_IoControl(IN PDEVICE_OBJECT DeviceObject,IN PIRP
Irp)
        {
        ...
            IrpSp = IoGetCurrentIrpStackLocation(Irp);         (1)

FunctionCode=IrpSp->Parameters.DeviceIoControl.IoControlCode;
            Open=IrpSp->FileObject->FsContext;
        ...
        ...
            case BIOCGSTATS: //function to get the capture stats  (2)

                TRACE_MESSAGE(PACKET_DEBUG_LOUD, "BIOCGSTATS");

                if(IrpSp->Parameters.DeviceIoControl.OutputBufferLength
< 4*sizeof(UINT))  (3)
                {
                    SET_FAILURE_BUFFER_SMALL();
                    break;
                }

                pStats = (PUINT)(Irp->UserBuffer);  (4)
```

```
pStats[3] = 0;  (5)
pStats[0] = 0;
pStats[1] = 0;
pStats[2] = 0;      // Not yet supported

for(i = 0 ; i < NCpu ; i++)  (6)
{

    pStats[3] += Open->CpuData[i].Accepted;
    pStats[0] += Open->CpuData[i].Received;
    pStats[1] += Open->CpuData[i].Dropped;
    pStats[2] += 0;     // Not yet supported   (7)
}

SET_RESULT_SUCCESS(4*sizeof(UINT));

break;
```

At (1), the IRP Stack Pointer is retrieved through IoGetCurrentIrpStackLocation. This structure contains, among other things, the parameters sent from user-mode. Then, at the next line, the IOCTL parameter is stored in the FunctionCode variable that will be used inside a switch;case sentence to choose the operation to be done.

In this case, the value in which we are interested in is (2) BIOCGSTATS.

At (3), it checks the output buffer size parameter to make sure that it can hold the data to be written (four unsigned ints). If it's not possible, it jumps out of the switch;case sentence.

At (4), it gets the address sent from user-mode as the output buffer.

Then, at (5) we can see the vulnerability itself. The driver writes 16 zeroes to the address specified from user-mode, without doing any kind of checks on it. In a normal scenario, this address should be a valid buffer pointer in the user address range, but an invalid address could be provided generating an access violation exception that, due to executing in ring 0, will lead to a BSOD. Let’s go further.

At (6), we have a loop that at each iteration adds different values to those contained in the array, except the third DWORD, which remains zeroed during the entire loop.

After that, execution leaves the switch; case.

The exploitation

From what was explained above, it should be trivial to exploit this vulnerability to crash the whole system. We just need to send an ioctl specifying an invalid kernel address like 0x80808080 as the output buffer. But let’s go a bit deeper.

Taking advantage of this bug gives us the possibility to modify 16 bytes at any writeable kernel address. At this moment we don’t really know with which values exactly, but without further analysis we can say that the 3rd DWORD will be always zeroes.

The question now would be: How can we obtain code execution in this context?

Patching the SSDT

The System Service Descriptor Table (SSDT) is a kernel structure containing a list of function pointers. These function pointers are called by the system service dispatcher when certain user-mode APIs, which need to do some operations at kernel mode, are called.

For example, when calling to the AddAtom() function from a user-mode process, the code at the DLL is responsible for the validation of some parameters and then it does the context switch to ring 0 via int 2e or sysenter (depending on de windows version) referencing the desired function by its index in the table. Then, the system service dispatcher redirects the execution to the corresponding pointer resending (and sometimes completing, or even modifying) the user mode parameters. This is the output of KD when looking at the SSDT. The addresses where the pointers are located remain constant between different versions of windows.

```
        kd> dds poi(KeServiceDescriptorTable)
        ...
        8050104c  805e8f86
    nt!NtAccessCheckByTypeResultListAndAuditAlarmByHandle
        80501050  8060a5da nt!NtAddAtom
        80501054  8060b84e nt!NtQueryBootOptions
        80501058  805e0a08 nt!NtAdjustGroupsToken
        8050105c  805e0660 nt!NtAdjustPrivilegesToken
        80501060  805c9684 nt!NtAlertResumeThread
        80501064  805c9634 nt!NtAlertThread
        80501068  8060ac00 nt!NtAllocateLocallyUniqueId
        8050106c  805aa088 nt!NtAllocateUserPhysicalPages
        80501070  8060a218 nt!NtAllocateUuids
        80501074  8059c910 nt!NtAllocateVirtualMemory
        ...
```

A possible attack vector (and a widely used one) when exploiting this kind of vulnerabilities relies in the utilization of the bug to modify some of the pointers in this table with a controlled value, to make it point to some user-range allocable memory region.

In this case, we know that for any address specified as output buffer, the driver will write:

- 8 bytes of unknown data (what gets written is pretty obvious actually, but we don't need to know it)
- 4 bytes with zeroes
- 4 bytes of unknown data.

At first sight, the only predictable values are the four zeroes. But, what can we do by patching a pointer with zeroes?

Well, a little trick can be used to allocate memory at page zero to put some code in there. This is possible by calling NtAllocateVirtualMemory with a base address of 1 because this function will round the value to the lower page, allocating memory starting at 0x0.

```
    PVOID Addr=(PVOID)0x1;
    NtAllocateVirtualMemory((HANDLE)-1, &Addr, 0, &Size, MEM_RESER-
VE|MEM_COMMIT|MEM_TOP_DOWN, PAGE_EXECUTE_READWRITE);
```

Then we can use these four zeroes to patch the desired entry. We just need to send the BIOCGSTATS ioctl to trigger the bug, passing to the driver the address of the function to patch - 8. After that, our selected function will be patched with 0s, pointing exactly to our allocated buffer.

```
DeviceIoControl(hDevice, 0x9031, lpInBuffer, nInBufferSize, (Address of
the selected function - 8), nOutBufferSize, &ret, NULL)
```

This technique has a little problem, because we are trashing 4 consecutive function pointers, so we must be very careful when selecting the functions to patch. These functions should be rarely used, and must be non critical ones. We can attach a debugger to set some breakpoints and see which functions are not called so often.

Finally, just a call to the user-mode counterpart of the patched function is needed to make the system service dispatcher call the kernel -patched- pointer, obtaining in this way, the so precious privileged execution of our user-mode allocated code. Normally, the code allocated at 0x0 will use one of the known methods to elevate the privileges of a desired process, but that's another topic.

As I was writing this article, I’ve found another little bug in the Winpcap code. Do you think you can find it?

Anibal Sacco is a SSr Exploit Writer at Core Security Technologies. He has been researching vulnerabilities and developing exploits for Windows, OS X and Linux for almost 3 years. Focusing for the past year and a half, on kernel-mode windows vulnerabilities.

Removing software armoring from executables

by Danny Quist

In the article published in issue 17 of (IN)SECURE I discussed various methods used to reverse engineer software armoring. This article will discuss the methods used by reverse engineers to remove software protections from an executable. Primarily the target of this exercise will be windows PE files, but the techniques should work for other environments as well. In order to understand the effect of armored code, we will need to develop an understanding of the process in which code is protected.

How software armoring works

Software armoring is a general term for any protection employed by an executable to prevent analysis or modification. It is often referred to as packing. Any sort of armoring system has a few key features. First, the executable must be modified in some manner to prevent analysis. This modification can be a simple xor encoding or as complicated as actual encryption or compression. The second feature is the executable must have some way to translate the encoded, encrypted, or compressed data into actual executable machine code. This step is important as the CPU can only execute valid machine code for its architecture. If the code is not valid the CPU will raise an exception resulting in a program crash. Third, the armoring process must not interfere with the original program's execution. This preserves the original behavior and execution of the running process.

Understanding the Windows Portable Executable (PE) format is important for reverse engineering applications. They key feature of the PE format is the AddressOfEntryPoint. This is the address where the Windows loader begins execution after the program has been loaded. In a normal executable this address would be the beginning of the original code, or the original entry point (OEP). When a program is armored, the entry point is modified to be a pointer to the armoring code.

This is important for the application as it translates the encoded executable data into machine code. Once the program is translated execution can then pass to the OEP and the program will execute as normal.

Removing software armoring

There are a couple of predominant methods for removing software armoring. These are manually translating the encrypted code via an external decryption program, manually unpacking with a debugger, and automated techniques.

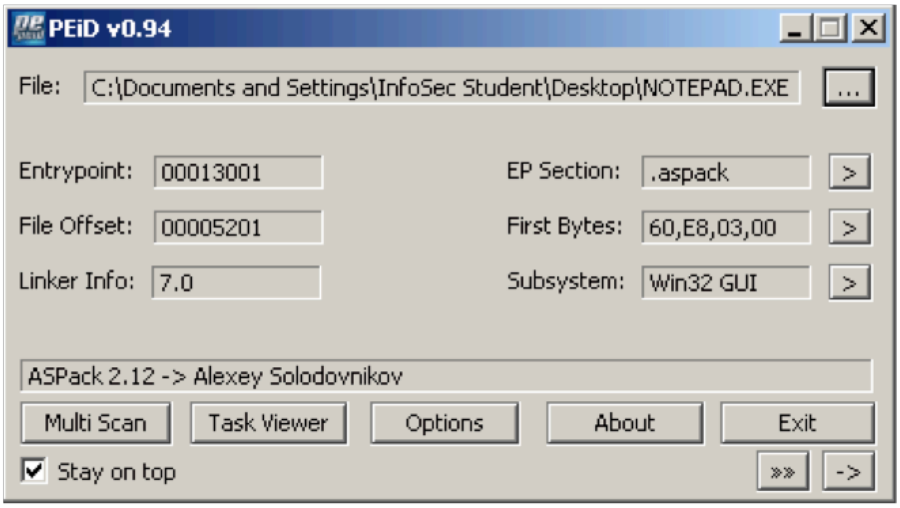
Decoding the program via an external decryption program is the first I will discuss. This technique works well when an executable's format is known ahead of time. For instance, the UPX program includes a command line switch that will decompress the executable to its original form. Many antivirus engines also include manual translators for various armoring techniques. An advantage to this technique is that it does not require the program to be executed. This is a much safer technique as there is no chance that malicious code can be executed. The problem with this is that a tool must be devised for every single armoring system. Slight changes to an algorithm will result in an invalid decode or a program crash.

In order for a program to be developed, the unpacking code must be reverse engineered so that it will decode the data properly. This can be a tedious and time consuming process.

Manually unpacking an armored executable is a method used with a debugger. The general process is to identify the type of unpacker, decode the file, and then create a memory snapshot. If the algorithm or packer ID is unknown, then the program must be executed in such a way to watch for execution of unpacked code. For this next section I will manually outline the process for decoding a known packer: ASPack 2.12.

The first step is to identify the armoring program. This can be done using tools such as PEiD or manual analysis. PEiD is a tool that uses signatures to identify a particular packer. It uses an internal database and an external one to identify relevant portions of the PE file to identify a packer. One can also look at the executable manually. Using a parsing library such as pefile,signatures in the section headers can be analyzed. The UPX and ASPack programs will rename section headers to include their names.

Let's begin removing the armor from a program. In this case we will look at an armored version of the Windows notepad program. If the armoring method is not known the first step is to identify it. In this case PEiD can be used to determine the packer.



File: C:\Documents and Settings\InfoSec Student\Desktop\NOTEPAD.EXE

Entrypoint: 00013001

File Offset: 00005201

Linker Info: 7.0

EP Section: .aspack

First Bytes: 60,E8,03,00

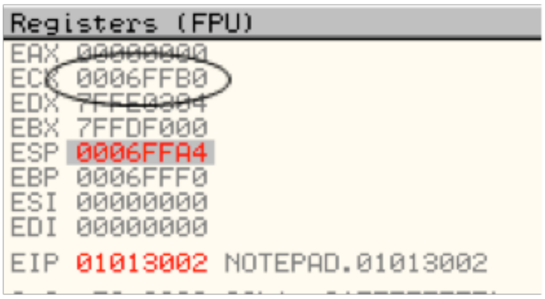
Subsystem: Win32 GUI

ASPack 2.12 -> Alexey Solodovnikov

Figure 1: PEiD showing ASPack 2.12

PEiD identifies the program as being protected by ASPack. Looking inside of IDA Pro, we can see that the program is obfuscated as the program information is very limited. To unpack this program, we will use several tools. The first is the OllyDbg debugger. OllyDbg allows you to single-step the program and monitor the program's execution. The next tool will be the OllyDbg plugin, OllyDump. This will allow you to take a snapshot of the running program memory. Let's load the executable inside of OllyDbg now.

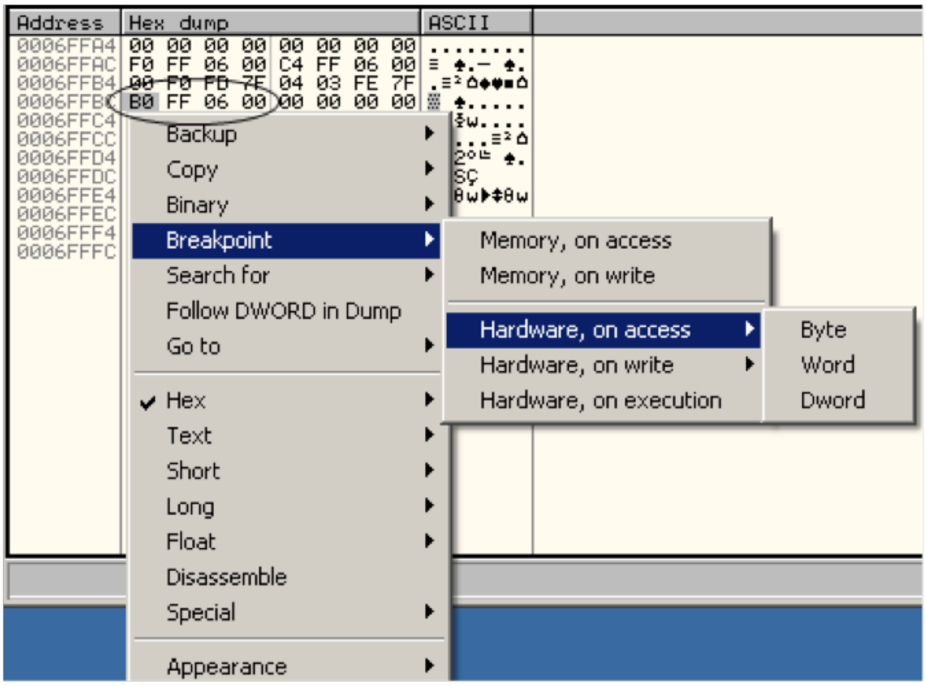
One of the first instructions of ASPack is the PUSHAD instruction. This copies all of the register values onto the stack. We begin by single-stepping the program once to allow the PUSHAD instruction to execute. The next step is to find the value of the stack pointer, esp, in the program's memory. The easiest way to do this is to right-click the register value and select "Follow in Dump." This will display the contents of the address in OllyDbg's memory dump window. Subsequently we must find the value of a register and set a hardware breakpoint. For this demonstration we will look at the value of the ecx register and find it in the memory dump. Remember that the bytes will be reversed due to the little endian byte ordering of the Intel CPU. Once the address has been found, simply set a hardware breakpoint on access for a DWORD.



```
Registers (FPU)
EAX 00000000
ECX 0006FFB0
EDX 7FFE0304
EBX 7FFDF000
ESP 0006FFA4
EBP 0006FFF0
ESI 00000000
EDI 00000000

EIP 01013002 NOTEPAD.01013002
```

Figure 2: Register Window



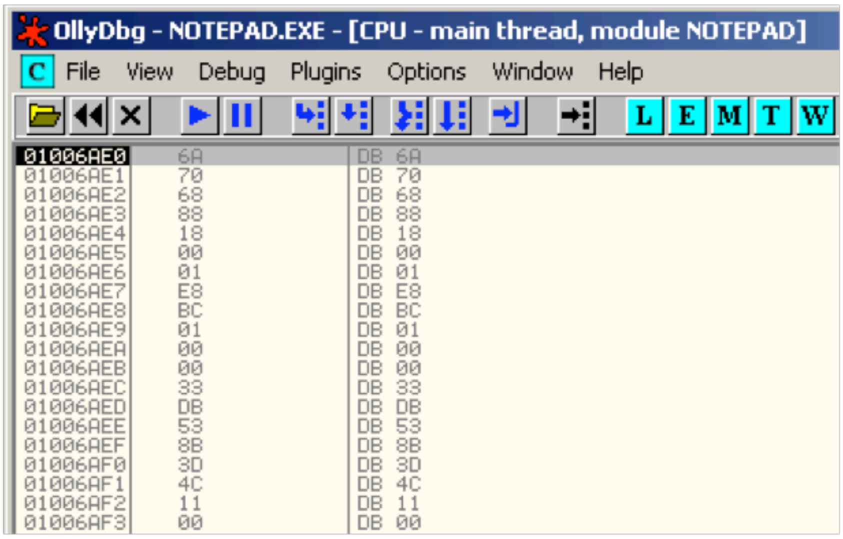
```
Address   Hex dump                 ASCII
0006FFA4  00 00 00 00 00 00 00 00  ........
0006FFAC  F0 FF 06 00 C4 FF 06 00  ≡ ♠.─ ♠.
0006FFB4  00 F0 FD 7F 04 03 FE 7F  .≡²⌂♦♥■⌂
0006FFBC  B0 FF 06 00 00 00 00 00  ░ ♠.....
0006FFC4
0006FFCC
0006FFD4
0006FFDC
0006FFE4
0006FFEC
0006FFF4
0006FFFC
```

Figure 3: Setting a breakpoint on access

After the breakpoint has been set, run the program. The hardware breakpoint should trigger after a POPAD instruction. Single-step the program a bit more and then something curious will happen: OllyDbg will show execution in an area that does not contain valid assembly.

This area of the program is actually the unpacked code at the OEP. Forcing OllyDbg to reanalyze this code (CTRL-A) will show that this is actually valid assembly. This is the code that has been unpacked and is ready to execute. We can now use the OllyDump plugin to create a snapshot of the executable from memory. Afterwards imports can be rebuilt using the Import Reconstructor tool. This allows creation of a fully de-armored executable.

Let's review what we've done. First we attached a debugger to the running program. Next we used our special trick for ASPack to find the unpacked segment of code. Finally we took a snapshot and rebuilt pertinent information from the running executable. It is important to note that most armoring programs have a trick for fast forwarding through the unpacking process. The problem with this method is that considerable effort must be put into finding this trick. When a new obfuscation technique emerges, this can become a tedious process to engage in.



```
01006AE0   6A   DB 6A
01006AE1   70   DB 70
01006AE2   68   DB 68
01006AE3   88   DB 88
01006AE4   18   DB 18
01006AE5   00   DB 00
01006AE6   01   DB 01
01006AE7   E8   DB E8
01006AE8   BC   DB BC
01006AE9   01   DB 01
01006AEA   00   DB 00
01006AEB   00   DB 00
01006AEC   33   DB 33
01006AED   DB   DB DB
01006AEE   53   DB 53
01006AEF   8B   DB 8B
01006AF0   3D   DB 3D
01006AF1   4C   DB 4C
01006AF2   11   DB 11
01006AF3   00   DB 00
```

Figure 4: Execution in unknown code

Automation of unpacking

Fundamentally we can observe that armoring programs work in a simple way. A decoding stub is executed that translates the encoded data into actual machine code. This leads to a fairly simple observation: at some point written memory must be executed via the decoding process. If we can track these writes and watch for execution in that written memory area, there's a good chance we have the unpacked code.

There are a couple of different methods for tracing memory writes and executions. The first is via a typical debugger. Automated scripts can be employed to track all the writes and executions to written memory. The problem with using a debugger is that this can be detected quite easily. The second method for tracing a program is to use dynamic instrumentation (DI). DI is a class of programs used to trace the runtime performance of a monitored program. This allows the program to trace all memory writes and executions. DI also suffers from the detection as it modifies the program memory. Ultimately the success of any automated de-obfuscation system relies on how well it can hide from the executable. Methods for subverting these systems range from full CPU emulation (PolyUnpack, many antivirus products), to OS feature overloading (OllyBonE, Saffron), to virtualization based systems (Azure).

Hybrid obfuscation methods

Recently a trend has emerged whereby malware is using a hybrid system to hide and obscure the executable. This next section will highlight a recent storm worm sample and detail the methodology used to extract data.

The storm worm is one of the most prolific viruses ever written. It is a front-end for a variety of illicit activities including distributed denial of service, spam, and phishing. It has sustained a near constant presence on the Internet and currently accounts for approximately 2% of all spam (tinyurl.com/64uyky). The authors of the storm worm have used a variety of techniques to elude analysis. This article will cover one of the latest techniques that storm uses: Process injection from the Windows kernel.

Analysis

Given the prolific nature of the storm worm, finding a copy simply involves reading a spam folder. The infection vector is an automatic download which saved the file on my hard disk and waited for me to click the link. I downloaded the file inside of my Windows XP Vmware image and started analysis.

The first method of analysis used was iDefense's system call monitoring tool called Sysanalyzer (tinyurl.com/67bw33). This is a good tool to use for determining what a particular sample is doing. Unfortunately results from this tool were very limited. The only information provided was that a file was dropped called C:\Windows\System32\diperto4417-e33.sys. Looking at the file inside of IDA Pro did not reveal any useful information either. Filemon showed that another file diperto.ini was written and contained. According to analysis from other researchers (tinyurl.com/6qybuk) this contains a list of Overnet (tinyurl.com/mz28s) P2P nodes.

Analysis of diperto4417-e33.sys

The next technique was to load the device driver into IDA and see what could be learned. This is where things got more interesting. The driver was not packed at all and analysis was possible without too much trouble. The file was not encrypted or packed and there were only 14 function calls.

The bulk of the complexity is performed by the function sub_106EE. Its sole purpose is to inject an executable into the services.exe process. This technique is very similar in functionality to a DLL injection; however this method exists entirely inside of kernel.

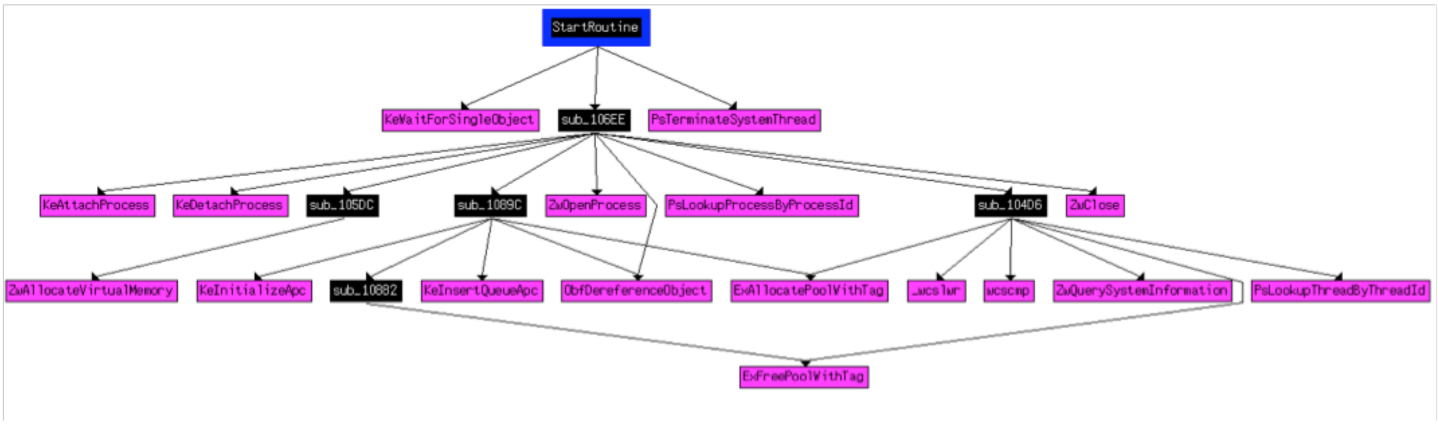
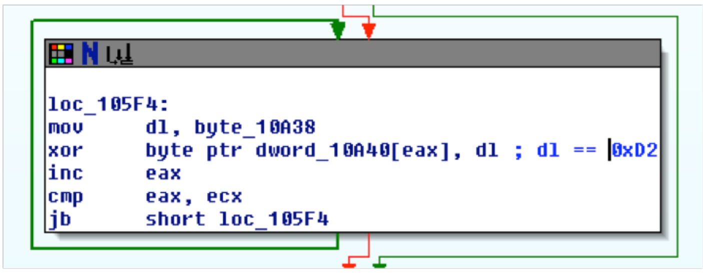


Figure 5: Call graph of xrefs from StartRoutine in diperto4417-e33.sys

Usermode process injection

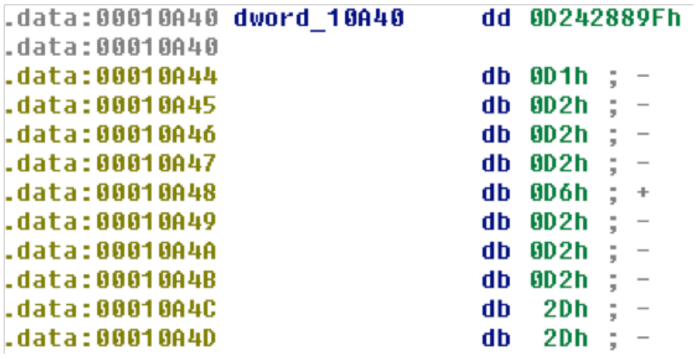
The first thing Storm does to find the process in memory is to enumerate the process list. The function sub_104D6 does this by loading the address at 0xC0000004 to enumerate the running processes. The result is returned as an ETHREAD pointer. Sub_105DC handles the insertion of the payload into the process's memory. The first task 105DC does is to decode the payload. The decode loop for the payload is a simple xor cipher located at dword_10A40. The size of the payload is contained at off_10A3C and is 122,880 bytes.



```
loc_105F4:
mov     dl, byte_10A38
xor     byte ptr dword_10A40[eax], dl ; dl == 0xD2
inc     eax
cmp     eax, ecx
jb      short loc_105F4
```

Figure 6: Decode loop for the payload inserted into services.exe

Before the xor decoding, the data at dword_10A40 is obviously garbage:



```
.data:00010A40 dword_10A40     dd 0D242889Fh
.data:00010A40
.data:00010A44                 db 0D1h ; -
.data:00010A45                 db 0D2h ; -
.data:00010A46                 db 0D2h ; -
.data:00010A47                 db 0D2h ; -
.data:00010A48                 db 0D6h ; +
.data:00010A49                 db 0D2h ; -
.data:00010A4A                 db 0D2h ; -
.data:00010A4B                 db 0D2h ; -
.data:00010A4C                 db  2Dh ; -
.data:00010A4D                 db  2Dh ; -
```

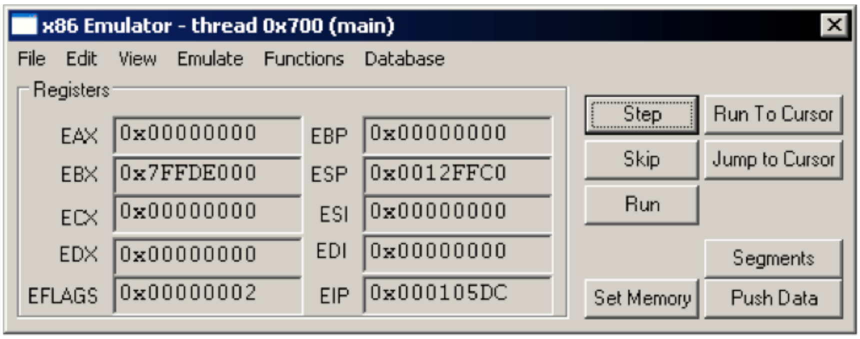
Figure 7: dword_10A40 before decoding

The next step is to attach to the process space and allocate memory. KeAttachProcess and ZwOpenProcess are used to prepare access to the process's memory. The function sub_105DC handles the insertion of code into the running process. Decoding the payload is addressed later. To execute code inside of the process, the undocumented asynchronous procedure call API is used. These calls are typically used to handle a completed I/O request from a device driver. Callback code can be registered to handle completed I/O events. In the case of the Storm worm, the decoded memory is created inside the user process space and then registered as a NormalRoutine inside of the APC data structure. This callback code is executed in the context of the userspace process instead of the kernel. There are many sources of documentation (tinyurl.com/5l6kmk) of this attack and it illustrates the future of these attacks.

Decoding the Payload using x86emu

There are many methods to decode the payload data. The first is to write a script inside of IDA using any number of methods. (IDC scripting, IDAPython, etc.) Second you can manually decode the data by hand manually. Given that this is a large 122k file, it is best to let a tool perform the decoding for you. Another option you can use is the excellent x86emu (idabook.com/x86emu/) tool by Chris Eagle. X86emu partially emulates an Intel instruction set which allows you to run small portions of the code. This was the method I chose to decode the data. The goal for decoding this portion of the payload is to extract the data that is being injected into the services.exe application. X86emu helps to ease this process. The first step is to install x86emu using the instructions provided in the README. After you have done that switch IDA to text mode if you haven't already done so. Next highlight the first instruction of the sub_105DC function. This should be "mov edi, edi". Invoke x86emu by pressing alt-F7. You should see a screen like that in figure 4.

The first thing that should be noticed is that EIP is set to the address of 105DC, corresponding to the function named sub_105DC.



x86 Emulator - thread 0x700 (main)

Registers

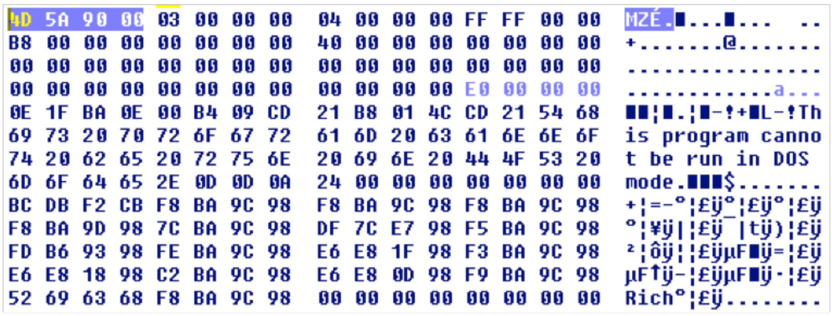
EAX 0x00000000
EBX 0x7FFDE000
ECX 0x00000000
EDX 0x00000000
EFLAGS 0x00000002
EBP 0x00000000
ESP 0x0012FFC0
ESI 0x00000000
EDI 0x00000000
EIP 0x000105DC

Figure 8: x86emu after startup

The next step is to push some data onto the stack via the "Push Data" button. In the window that pops up enter "0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0" or something similar.

You can now single-step the emulator to step through the assembly and observe it in action. Skipping the decode loop is of critical importance as it will save us the work of actually having to step 122,880 times. To do this, find the instruction immediately following the loop (.text:00010605 push ebx) and click the “Run to Cursor” button.

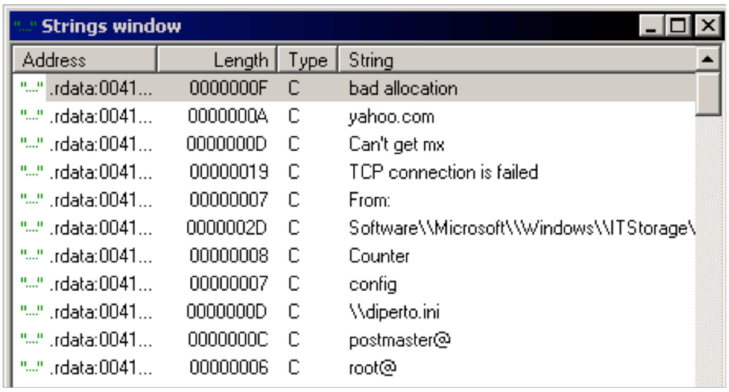
This decodes the entire address space and modifies the IDA database to contain the correct value. Observing the memory located at address 10A40 in hex mode will show a familiar series of bytes (see figure 5). This memory looks to be the beginnings of a normal portable executable (PE) file.



```
4D 5A 90 00 03 00 00 00  04 00 00 00 FF FF 00 00  MZÉ.■...■...  ..
B8 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  +.......@.......
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00  00 00 00 00 E0 00 00 00  ............a...
0E 1F BA 0E 00 B4 09 CD  21 B8 01 4C CD 21 54 68  ■■¦■.¦■-!+■L-!Th
69 73 20 70 72 6F 67 72  61 6D 20 63 61 6E 6E 6F  is program canno
74 20 62 65 20 72 75 6E  20 69 6E 20 44 4F 53 20  t be run in DOS 
6D 6F 64 65 2E 0D 0D 0A  24 00 00 00 00 00 00 00  mode.■■■$.......
BC DB F2 CB F8 BA 9C 98  F8 BA 9C 98 F8 BA 9C 98  +¦=-°¦£ÿ°¦£ÿ°¦£ÿ
F8 BA 9D 98 7C BA 9C 98  DF 7C E7 98 F5 BA 9C 98  °¦¥ÿ|¦£ÿ ¯|tÿ)¦£ÿ
FD B6 93 98 FE BA 9C 98  E6 E8 1F 98 F3 BA 9C 98  ²¦ôÿ¦¦£ÿµF■ÿ=¦£ÿ
E6 E8 18 98 C2 BA 9C 98  E6 E8 0D 98 F9 BA 9C 98  µF↑ÿ-¦£ÿµF■ÿ·¦£ÿ
52 69 63 68 F8 BA 9C 98  00 00 00 00 00 00 00 00  Rich°¦£ÿ........
```

Figure 9: Decoded payload after using x86emu

The next step is to dump the contents of the memory. X86emu has a memory dumping tool that can be found by going to File->Dump. The dialog box will ask you to enter in a range of memory to dump starting at the address specified. To dump the contents of the memory simply take the starting address, 0x00010A40 and add the size 122,880 to it (0x1E000). This will yield the address of 0x0002EA40. Provide a filename and you can get a copy of the code that is injected into service.exe. This executable, when loaded into IDA, yields the actual program with good strings.



Address	Length	Type	String
"..." .rdata:0041...	0000000F	C	bad allocation
"..." .rdata:0041...	0000000A	C	yahoo.com
"..." .rdata:0041...	0000000D	C	Can't get mx
"..." .rdata:0041...	00000019	C	TCP connection is failed
"..." .rdata:0041...	00000007	C	From:
"..." .rdata:0041...	0000002D	C	Software\\Microsoft\\Windows\\ITStorage\
"..." .rdata:0041...	00000008	C	Counter
"..." .rdata:0041...	00000007	C	config
"..." .rdata:0041...	0000000D	C	\\diperto.ini
"..." .rdata:0041...	0000000C	C	postmaster@
"..." .rdata:0041...	00000006	C	root@

Reverse engineering any software armoring system requires patience, skill, and constant practice. Leveraging the inherent weaknesses present in all armoring systems allows the analyst to remove any protection. The state of the art of software protection is advancing steadily but so too are the subversion methods. The sophistication of evasion tactics is increasing and will require further innovation to be able to maintain automated analysis techniques. In many cases traditional packers are being replaced with simpler encoding techniques combined with more complicated subversion methods.

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.

Insecurities in privacy protection software

by Shrikant Raman

The widespread usage of personal information over the Internet has led to malware programs capitalizing on this information and stealing it from computers in a variety of different ways. User education is probably the best way to prevent the inadvertent loss of personal information, but considering that a majority of Internet users seldom worry about protecting this information, the task of preventing its theft has become a daunting task.

Security software vendors adopted a new approach – the installation of software that would notify the users if personal information left their computers over the network. Over the past few years the demand for Personal Information Protection software has exploded exponentially, and security software vendors have been trying to incorporate a number of features into their programs to protect an individual's PII (Personally Identifiable Information). This information includes bank account numbers, credit card numbers, user names, passwords, SSNs, postal addresses etc. The programs request this confidential information from the user and monitor any network activity to prevent the inadvertent loss of this information over the network. The problem is – if this information is not protected adequately by the security software itself, it becomes a new security risk – as malware can now just look at specific memory locations to gain access to this information.

When software – such as the ones provided by the security software vendors - requests your passwords, credit card numbers, Social Security Numbers or other personal information, in an attempt to monitor and protect that data from insecurely or inadvertently leaving your computer, it needs to ensure it does so in a secure manner. Ideally, it should create a secure hash of this data, completely delete all instances of the original information in its buffers and temporary locations (both in memory and the disk), and use this hash for all future uses if possible. Never should this information be stored in the clear. Any time the program needs to make sure information leaving the computer is not part of the user provided list, it should compare the hashes.

Recently, I happened to read an excellent article in PC World by Erik Larkin titled "Does your Security Suite Also Protect Your Privacy?" The article touched upon a number of key features present in four software packages that had privacy features incorporated in them. However, the article missed reviewing one point: the protection of your private data by the software itself.

In an attempt to bridge that gap, I reviewed the same security packages available (with one change - I replaced Kaspersky Internet Security with Panda Internet Security 2008) - for the security the package provided to the user's confidential after receiving it from the user. I used publicly available programs like Notmyfault, Pmdump, Winhex, IdaPro and Filemon to dump the program from memory, analyze the contents and monitor any file activity.

For each of the 4 software programs, I did the following:

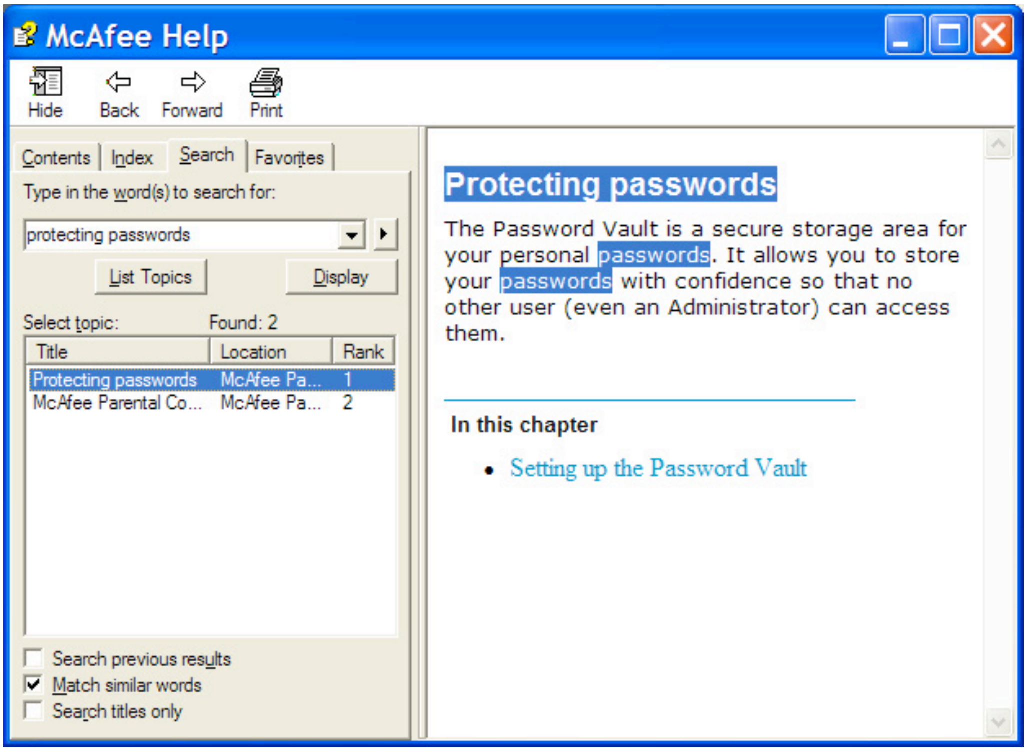
1. I entered test data for the program to protect for various pieces like SSN, Card number, Bank account number etc. I then logged out, shutdown the computer, restarted it and reviewed the data in the software I had entered earlier by clicking on the particular option for it. The goal was to validate that it does not echo any sensitive data back on the screen while attempting to look at it through the program.

2. I then looked at the processes that were started by the particular software suite and reviewed each process in an attempt to identify the process that was responsible for the privacy control feature. Once that was identified, I dumped the contents of memory used by that particular process to a file for further analysis. I searched through the file for any of the original test data that I had entered earlier.

The following results of the testing only show how the packages handled scenarios (1) and (2) described above. They do not provide any insight into any other software bugs or features and should not be treated as recommendations for any of the software suites. There are a number of other features that should be looked at while evaluating software for your use and protection of your sensitive information in memory is only one of them.

McAfee Internet Security Suite

The McAfee Internet Security Suite offers two features for information protection:
a. The Personal Information Protection page.
b. The Password Vault – to securely store login and password information.



Title	Location	Rank
Protecting passwords	McAfee Pa...	1
McAfee Parental Co...	McAfee Pa...	2

□ Search previous results
☑ Match similar words
□ Search titles only

Protecting passwords

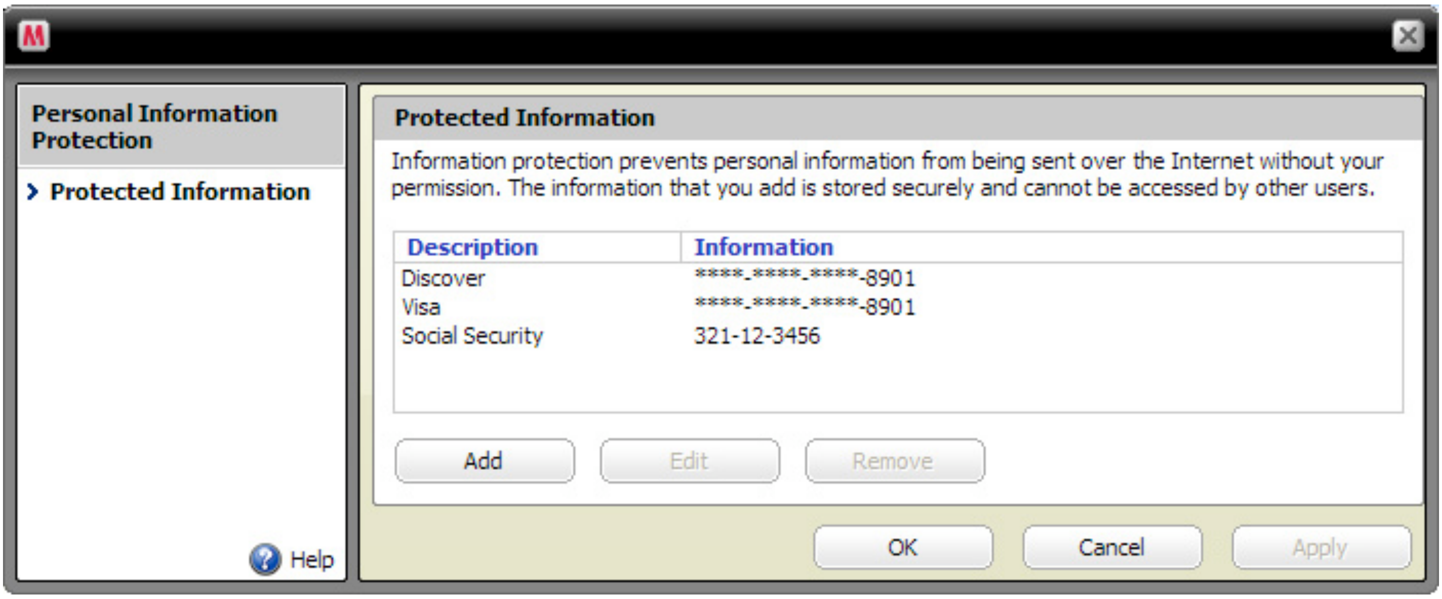
The Password Vault is a secure storage area for your personal passwords. It allows you to store your passwords with confidence so that no other user (even an Administrator) can access them.

In this chapter

- Setting up the Password Vault

Figure 1 - McAfee Help Screen on Password Vault.

While reviewing McAfee's Personal Information Protection, I entered two credit card numbers and one Social Security Number. The program was smart enough to mask the full 16 digit card numbers and displayed only the last 4 digits, but for the SSN – it echoed the entire SSN back on the screen every time I reviewed the page – all 9 digits.



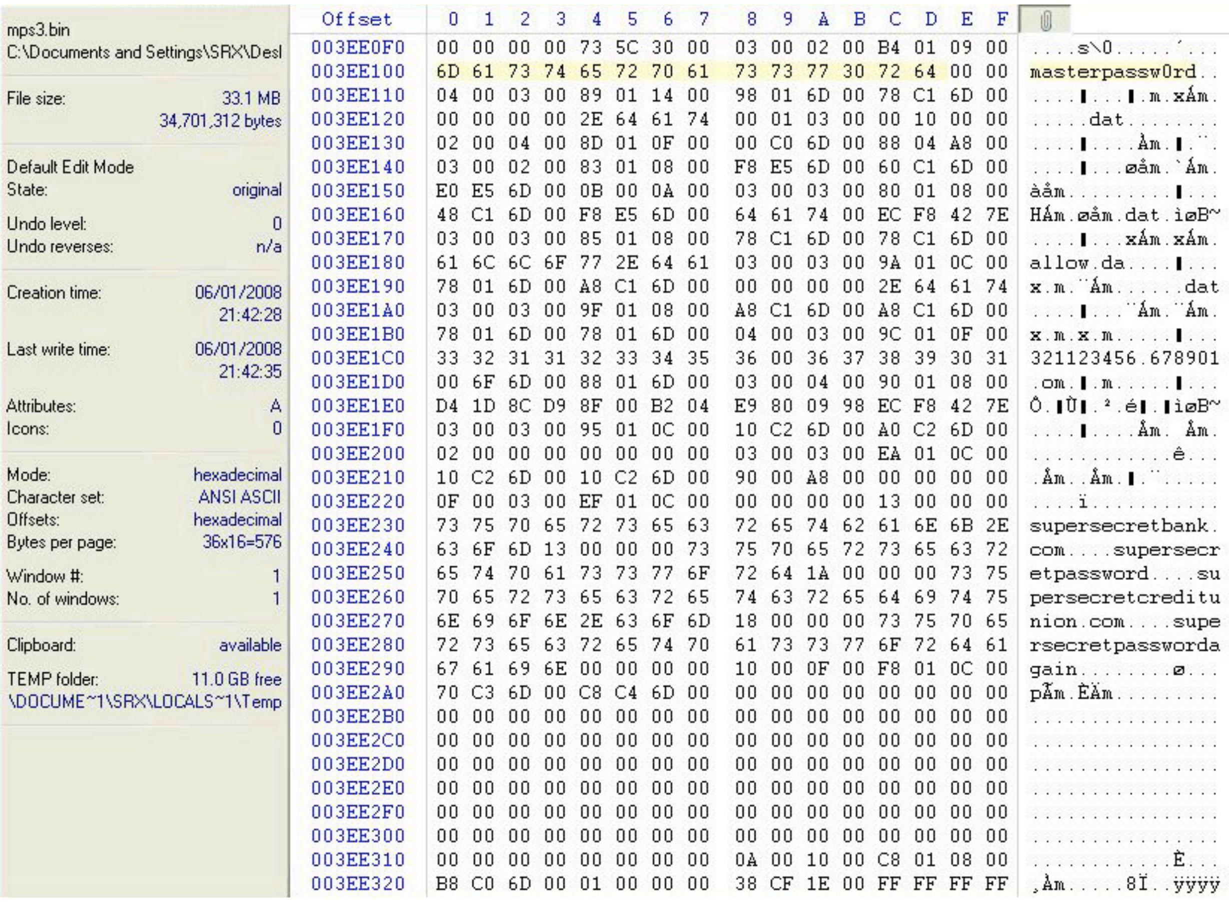
Protected Information

Information protection prevents personal information from being sent over the Internet without your permission. The information that you add is stored securely and cannot be accessed by other users.

Description	Information
Discover	****-****-****-8901
Visa	****-****-****-8901
Social Security	321-12-3456

Figure 2 - McAfee software displaying full social security numbers.

I then proceeded to review memory for the various processes created and used by the program. The main logic is in a process called mps.exe. After dumping the contents of mps.exe, a quick WinHex/IDAPro view provided me with all the information I had entered: full 16 digit card numbers, all my bank logins and passwords, and even the master password (masterpassw0rd) that I used to protect the password vault itself!



mps3.bin
C:\Documents and Settings\SRX\Desl

File size: 33.1 MB
34,701,312 bytes

Default Edit Mode
State: original

Undo level: 0
Undo reverses: n/a

Creation time: 06/01/2008 21:42:28

Last write time: 06/01/2008 21:42:35

Attributes: A
Icons: 0

Mode: hexadecimal
Character set: ANSI ASCII
Offsets: hexadecimal
Bytes per page: 36x16=576

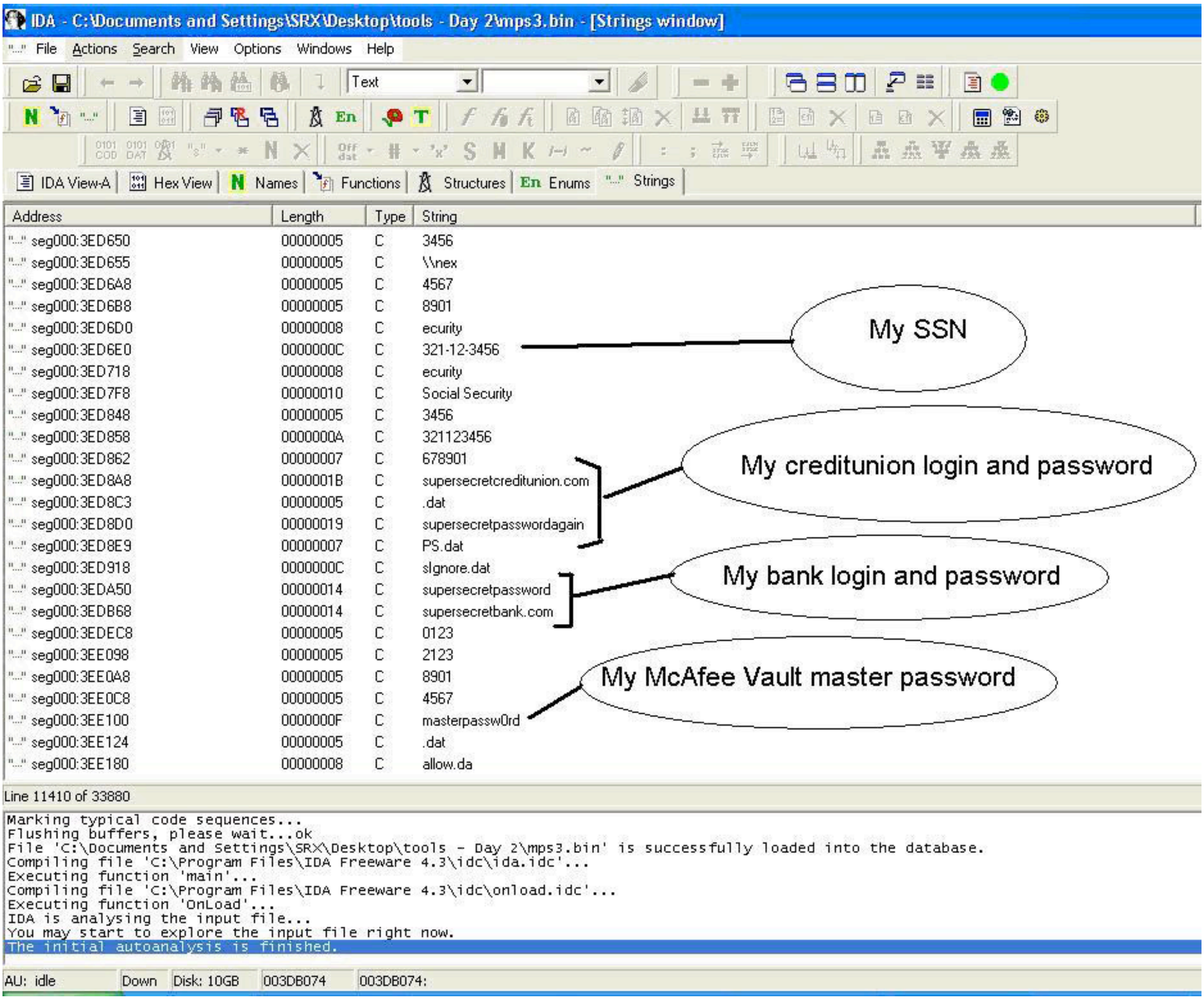
Window #: 1
No. of windows: 1

Clipboard: available

TEMP folder: 11.0 GB free
\DOCUME~1\SRX\LOCALS~1\Temp

```
  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
003EE0F0   00 00 00 00 73 5C 30 00  03 00 02 00 B4 01 09 00   ....s\0.....´...
003EE100   6D 61 73 74 65 72 70 61  73 73 77 30 72 64 00 00   masterpassw0rd..
003EE110   04 00 03 00 89 01 14 00  98 01 6D 00 78 C1 6D 00   ....▮...▮.m.xÁm.
003EE120   00 00 00 00 2E 64 61 74  00 01 03 00 00 10 00 00   .....dat........
003EE130   02 00 04 00 8D 01 0F 00  00 C0 6D 00 88 04 A8 00   ....▮....Àm.▮.¨.
003EE140   03 00 02 00 83 01 08 00  F8 E5 6D 00 60 C1 6D 00   ....▮...øåm.`Ám.
003EE150   E0 E5 6D 00 0B 00 0A 00  03 00 03 00 80 01 08 00   àåm.........▮...
003EE160   48 C1 6D 00 F8 E5 6D 00  64 61 74 00 EC F8 42 7E   HÁm.øåm.dat.iøB~
003EE170   03 00 03 00 85 01 08 00  78 C1 6D 00 78 C1 6D 00   ....▮...xÁm.xÁm.
003EE180   61 6C 6C 6F 77 2E 64 61  03 00 03 00 9A 01 0C 00   allow.da....▮...
003EE190   78 01 6D 00 A8 C1 6D 00  00 00 00 00 2E 64 61 74   x.m.¨Ám......dat
003EE1A0   03 00 03 00 9F 01 08 00  A8 C1 6D 00 A8 C1 6D 00   ....▮...¨Ám.¨Ám.
003EE1B0   78 01 6D 00 78 01 6D 00  04 00 03 00 9C 01 0F 00   x.m.x.m.....▮...
003EE1C0   33 32 31 31 32 33 34 35  36 00 36 37 38 39 30 31   321123456.678901
003EE1D0   00 6F 6D 00 88 01 6D 00  03 00 04 00 90 01 08 00   .om.▮.m.....▮...
003EE1E0   D4 1D 8C D9 8F 00 B2 04  E9 80 09 98 EC F8 42 7E   Ô.▮Ù▮.².é▮.▮iøB~
003EE1F0   03 00 03 00 95 01 0C 00  10 C2 6D 00 A0 C2 6D 00   ....▮....Âm. Âm.
003EE200   02 00 00 00 00 00 00 00  03 00 03 00 EA 01 0C 00   ............ê...
003EE210   10 C2 6D 00 10 C2 6D 00  90 00 A8 00 00 00 00 00   .Âm..Âm.▮.¨.....
003EE220   0F 00 03 00 EF 01 0C 00  00 00 00 00 13 00 00 00   ....ï...........
003EE230   73 75 70 65 72 73 65 63  72 65 74 62 61 6E 6B 2E   supersecretbank.
003EE240   63 6F 6D 13 00 00 00 73  75 70 65 72 73 65 63 72   com....supersecr
003EE250   65 74 70 61 73 73 77 6F  72 64 1A 00 00 00 73 75   etpassword....su
003EE260   70 65 72 73 65 63 72 65  74 63 72 65 64 69 74 75   persecretcreditu
003EE270   6E 69 6F 6E 2E 63 6F 6D  18 00 00 00 73 75 70 65   nion.com....supe
003EE280   72 73 65 63 72 65 74 70  61 73 73 77 6F 72 64 61   rsecretpassworda
003EE290   67 61 69 6E 00 00 00 00  10 00 0F 00 F8 01 0C 00   gain........ø...
003EE2A0   70 C3 6D 00 C8 C4 6D 00  00 00 00 00 00 00 00 00   pÃm.ÈÄm.........
003EE2B0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
003EE2C0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
003EE2D0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
003EE2E0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
003EE2F0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
003EE300   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ................
003EE310   00 00 00 00 00 00 00 00  0A 00 10 00 C8 01 08 00   ............È...
003EE320   B8 C0 6D 00 01 00 00 00  38 CF 1E 00 FF FF FF FF   ¸Àm.....8Ï..ÿÿÿÿ
```

Figure 3a - McAfee software storing the master password in clear text.



IDA - C:\Documents and Settings\SRX\Desktop\tools - Day 2\mps3.bin - [Strings window]
File Actions Search View Options Windows Help
IDA View-A | Hex View | Names | Functions | Structures | Enums | Strings
My SSN
My creditunion login and password
My bank login and password
My McAfee Vault master password

Address	Length	Type	String
seg000:3ED650	00000005	C	3456
seg000:3ED655	00000005	C	\\nex
seg000:3ED6A8	00000005	C	4567
seg000:3ED6B8	00000005	C	8901
seg000:3ED6D0	00000008	C	ecurity
seg000:3ED6E0	0000000C	C	321-12-3456
seg000:3ED718	00000008	C	ecurity
seg000:3ED7F8	00000010	C	Social Security
seg000:3ED848	00000005	C	3456
seg000:3ED858	0000000A	C	321123456
seg000:3ED862	00000007	C	678901
seg000:3ED8A8	0000001B	C	supersecretcreditunion.com
seg000:3ED8C3	00000005	C	.dat
seg000:3ED8D0	00000019	C	supersecretpasswordagain
seg000:3ED8E9	00000007	C	PS.dat
seg000:3ED918	0000000C	C	sIgnore.dat
seg000:3EDA50	00000014	C	supersecretpassword
seg000:3EDB68	00000014	C	supersecretbank.com
seg000:3EDEC8	00000005	C	0123
seg000:3EE098	00000005	C	2123
seg000:3EE0A8	00000005	C	8901
seg000:3EE0C8	00000005	C	4567
seg000:3EE100	0000000F	C	masterpassw0rd
seg000:3EE124	00000005	C	.dat
seg000:3EE180	00000008	C	allow.da

Line 11410 of 33880

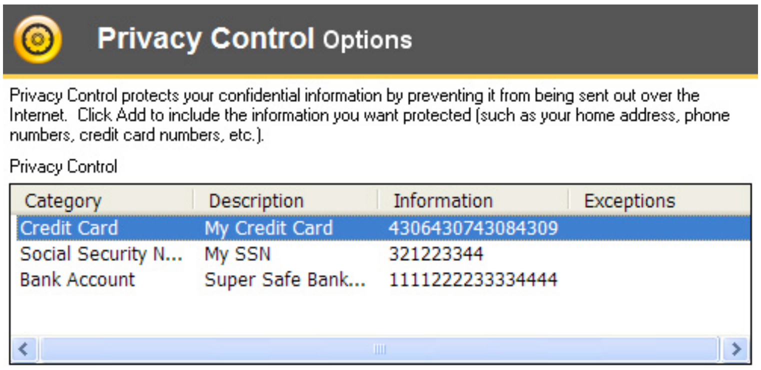
```
Marking typical code sequences...
Flushing buffers, please wait...ok
File 'C:\Documents and Settings\SRX\Desktop\tools - Day 2\mps3.bin' is successfully loaded into the database.
Compiling file 'C:\Program Files\IDA Freeware 4.3\idc\ida.idc'...
Executing function 'main'...
Compiling file 'C:\Program Files\IDA Freeware 4.3\idc\onload.idc'...
Executing function 'OnLoad'...
IDA is analysing the input file...
You may start to explore the input file right now.
The initial autoanalysis is finished.
```

AU: idle | Down | Disk: 10GB | 003DB074 | 003DB074:

Fig 3b - Strings window displaying sensitive information stored in cleartext in memory.

Norton Internet Security 2008

Norton Internet Security 2008 offers the Privacy Control feature through its free Norton Add-on Pack. Norton Internet Security's password vault called Identity Safe. After entering all the card data and during review, I noticed that the card information that I had stored in it required an additional password to unlock. Once I successfully entered the password, it displayed all the information about the card and carefully masked all but the last 4 digits of the credit card number. The privacy control feature, which was part of the add-on package, also required a password, but on successfully entering the password, however, it displayed all the information stored in it – card number, bank account number and SSN.

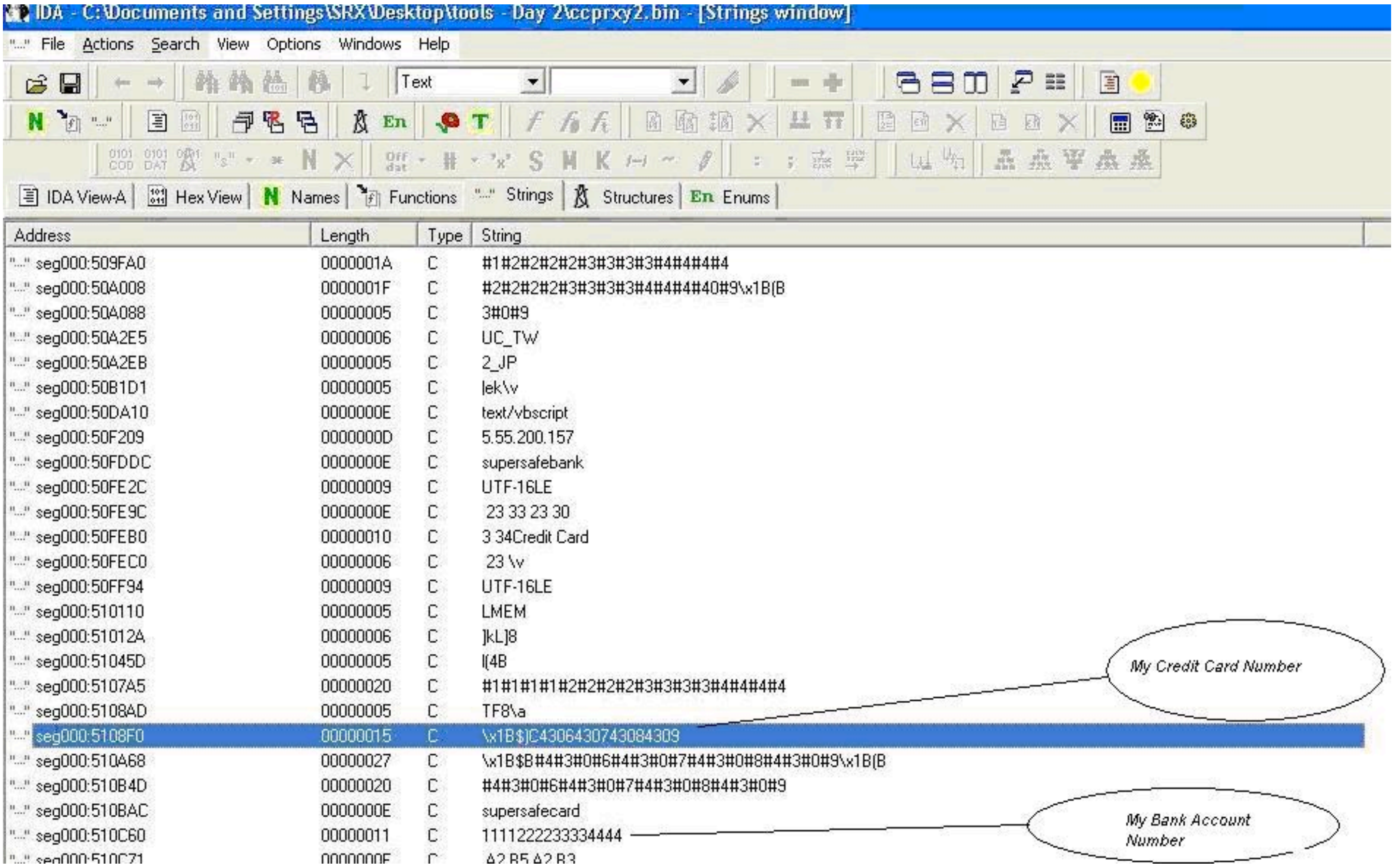


Privacy Control protects your confidential information by preventing it from being sent out over the Internet. Click Add to include the information you want protected (such as your home address, phone numbers, credit card numbers, etc.).

Privacy Control

Category	Description	Information	Exceptions
Credit Card	My Credit Card	4306430743084309	
Social Security N...	My SSN	321223344	
Bank Account	Super Safe Bank...	1111222233334444	

The Norton process that handles privacy control is called ccProxy.exe. On reviewing the ccProxy.exe in memory I could clearly see a number of occurrences of my SSN, my card numbers and bank account numbers in clear text.



Address	Length	Type	String
seg000:509FA0	0000001A	C	#1#2#2#2#2#3#3#3#3#4#4#4#4#4
seg000:50A008	0000001F	C	#2#2#2#2#3#3#3#3#4#4#4#4#40#9\x1B(B
seg000:50A088	00000005	C	3#0#9
seg000:50A2E5	00000006	C	UC_TW
seg000:50A2EB	00000005	C	2_JP
seg000:50B1D1	00000005	C	lek\v
seg000:50DA10	0000000E	C	text/vbscript
seg000:50F209	0000000D	C	5.55.200.157
seg000:50FDDC	0000000E	C	supersafebank
seg000:50FE2C	00000009	C	UTF-16LE
seg000:50FE9C	0000000E	C	23 33 23 30
seg000:50FEB0	00000010	C	3 34Credit Card
seg000:50FEC0	00000006	C	23 \v
seg000:50FF94	00000009	C	UTF-16LE
seg000:510110	00000005	C	LMEM
seg000:51012A	00000006	C	]kL]8
seg000:51045D	00000005	C	l(4B
seg000:5107A5	00000020	C	#1#1#1#1#2#2#2#2#3#3#3#3#4#4#4#4
seg000:5108AD	00000005	C	TF8\a
seg000:5108F0	00000015	C	\x1B$)C4306430743084309
seg000:510A68	00000027	C	\x1B$B#4#3#0#6#4#3#0#7#4#3#0#8#4#3#0#9\x1B(B
seg000:510B4D	00000020	C	#4#3#0#6#4#3#0#7#4#3#0#8#4#3#0#9
seg000:510BAC	0000000E	C	supersafecard
seg000:510C60	00000011	C	1111222233334444
seg000:510C71	0000000F	C	A2 B5 A2 B3

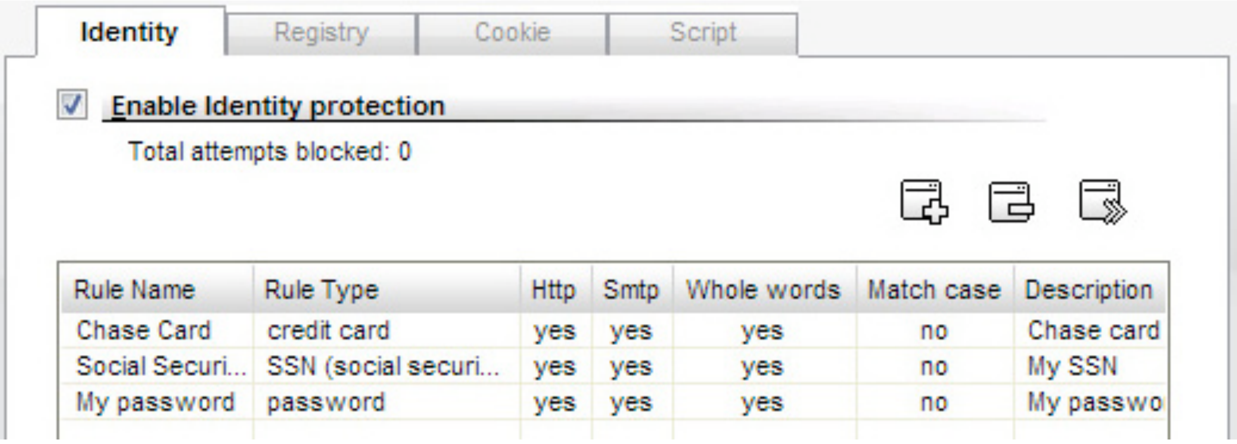
My Credit Card Number

My Bank Account Number

Figure 5 - Memory content of the Symantec Software showing cleartext storage of sensitive information.

BitDefender Internet Security 2008

BitDefender 2008 allowed for privacy control features in its Advanced Identity Control Settings. I entered my personal information, rules and descriptions each of those rules. While reviewing the data I input earlier, I saw that my entire personal information that I input earlier, was masked.

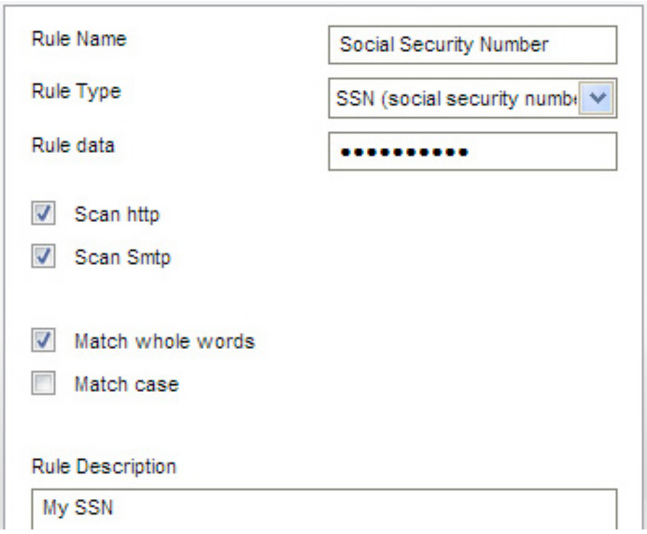


☑ Enable Identity protection

Total attempts blocked: 0

Rule Name	Rule Type	Http	Smtp	Whole words	Match case	Description
Chase Card	credit card	yes	yes	yes	no	Chase card
Social Securi...	SSN (social securi...	yes	yes	yes	no	My SSN
My password	password	yes	yes	yes	no	My passwo

Figure 6a - BitDefender’s Identity Protection Module.



Rule Name: Social Security Number

Rule Type: SSN (social security numb

Rule data: ●●●●●●●●●●●

☑ Scan http

☑ Scan Smtp

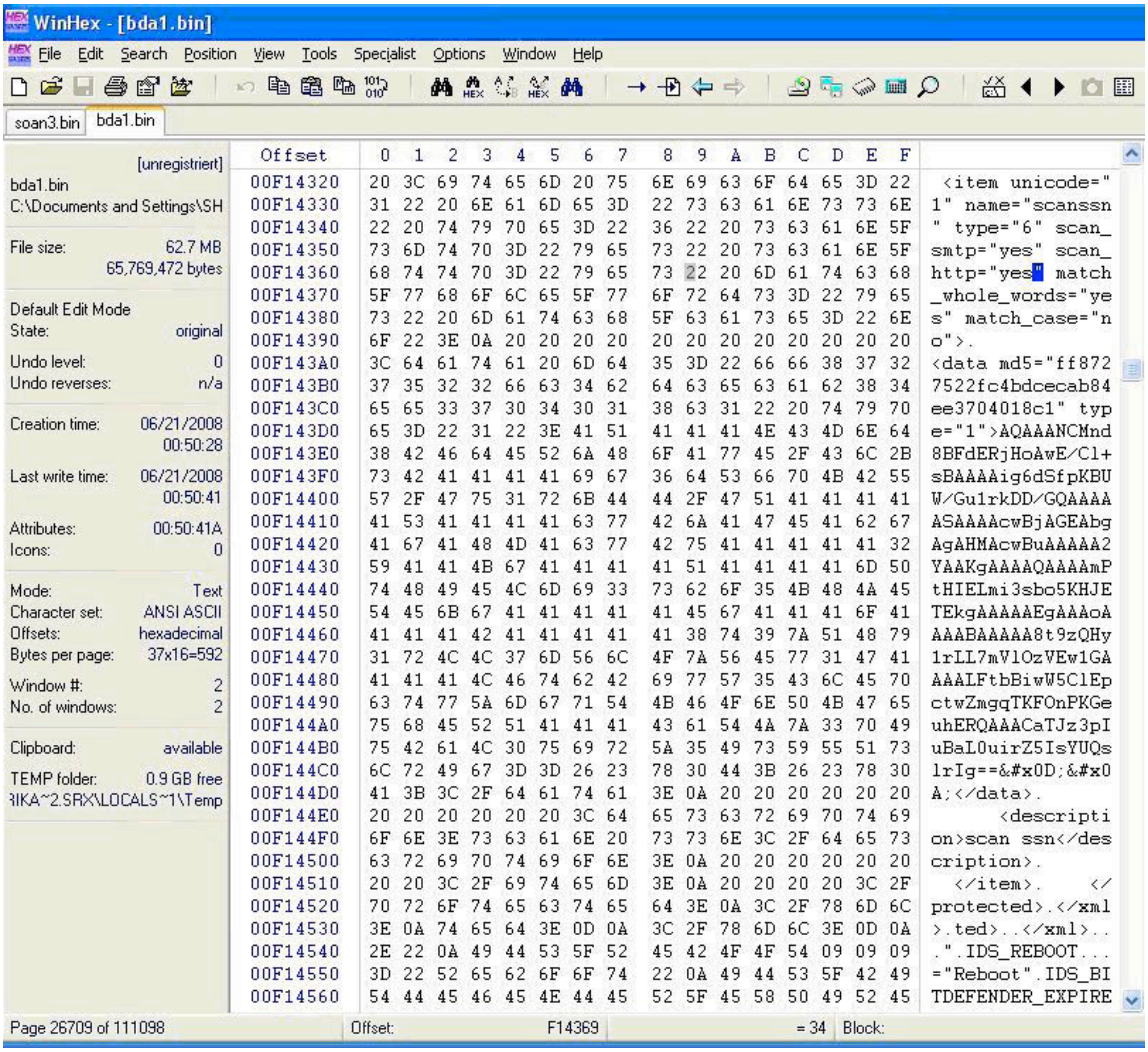
☑ Match whole words

□ Match case

Rule Description: My SSN

Figure 6b - BitDefender’s Identity Protection Module.

I proceeded to review the process space for BitDefender’s privacy control module – bdagent.exe. I didn’t find any instances of my card numbers, bank account numbers or SSNs. I found some interesting snippets of data, with what looked like MD5 hashes, presumably my personal information.



WinHex - [bda1.bin]
File Edit Search Position View Tools Specialist Options Window Help
soan3.bin
bda1.bin

```
[unregistriert]
bda1.bin
C:\Documents and Settings\SH

File size:             62.7 MB
               65,769,472 bytes

Default Edit Mode
State:                original

Undo level:                  0
Undo reverses:             n/a

Creation time:      06/21/2008
                      00:50:28

Last write time:    06/21/2008
                      00:50:41

Attributes:           00:50:41A
Icons:                       0

Mode:                     Text
Character set:      ANSI ASCII
Offsets:           hexadecimal
Bytes per page:     37x16=592

Window #:                    2
No. of windows:              2

Clipboard:           available

TEMP folder:       0.9 GB free
}IKA~2.SRX\LOCALS~1\Temp
```

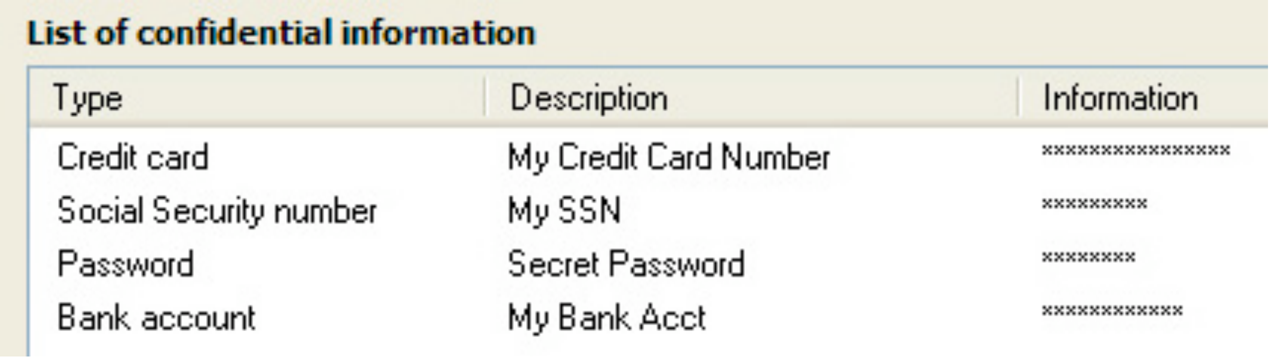
```
 Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00F14320   20 3C 69 74 65 6D 20 75  6E 69 63 6F 64 65 3D 22   <item unicode="
00F14330   31 22 20 6E 61 6D 65 3D  22 73 63 61 6E 73 73 6E  1" name="scanssn
00F14340   22 20 74 79 70 65 3D 22  36 22 20 73 63 61 6E 5F  " type="6" scan_
00F14350   73 6D 74 70 3D 22 79 65  73 22 20 73 63 61 6E 5F  smtp="yes" scan_
00F14360   68 74 74 70 3D 22 79 65  73 22 20 6D 61 74 63 68  http="yes" match
00F14370   5F 77 68 6F 6C 65 5F 77  6F 72 64 73 3D 22 79 65  _whole_words="ye
00F14380   73 22 20 6D 61 74 63 68  5F 63 61 73 65 3D 22 6E  s" match_case="n
00F14390   6F 22 3E 0A 20 20 20 20  20 20 20 20 20 20 20 20  o">.
00F143A0   3C 64 61 74 61 20 6D 64  35 3D 22 66 66 38 37 32  <data md5="ff872
00F143B0   37 35 32 32 66 63 34 62  64 63 65 63 61 62 38 34  7522fc4bdcecab84
00F143C0   65 65 33 37 30 34 30 31  38 63 31 22 20 74 79 70  ee3704018c1" typ
00F143D0   65 3D 22 31 22 3E 41 51  41 41 41 4E 43 4D 6E 64  e="1">AQAAANCMnd
00F143E0   38 42 46 64 45 52 6A 48  6F 41 77 45 2F 43 6C 2B  8BFdERjHoAwE/Cl+
00F143F0   73 42 41 41 41 41 69 67  36 64 53 66 70 4B 42 55  sBAAAAig6dSfpKBU
00F14400   57 2F 47 75 31 72 6B 44  44 2F 47 51 41 41 41 41  W/Gu1rkDD/GQAAAA
00F14410   41 53 41 41 41 41 63 77  42 6A 41 47 45 41 62 67  ASAAAAcwBjAGEAbg
00F14420   41 67 41 48 4D 41 63 77  42 75 41 41 41 41 41 32  AgAHMAcwBuAAAAA2
00F14430   59 41 41 4B 67 41 41 41  41 51 41 41 41 41 6D 50  YAAKgAAAAQAAAAmP
00F14440   74 48 49 45 4C 6D 69 33  73 62 6F 35 4B 48 4A 45  tHIELmi3sbo5KHJE
00F14450   54 45 6B 67 41 41 41 41  41 45 67 41 41 41 6F 41  TEkgAAAAAEgAAAoA
00F14460   41 41 41 42 41 41 41 41  41 38 74 39 7A 51 48 79  AAABAAAAA8t9zQHy
00F14470   31 72 4C 4C 37 6D 56 6C  4F 7A 56 45 77 31 47 41  1rLL7mVlOzVEw1GA
00F14480   41 41 41 4C 46 74 62 42  69 77 57 35 43 6C 45 70  AAALFtbBiwW5ClEp
00F14490   63 74 77 5A 6D 67 71 54  4B 46 4F 6E 50 4B 47 65  ctwZmgqTKFOnPKGe
00F144A0   75 68 45 52 51 41 41 41  43 61 54 4A 7A 33 70 49  uhERQAAACaTJz3pI
00F144B0   75 42 61 4C 30 75 69 72  5A 35 49 73 59 55 51 73  uBaL0uirZ5IsYUQs
00F144C0   6C 72 49 67 3D 3D 26 23  78 30 44 3B 26 23 78 30  lrIg==&#x0D;&#x0
00F144D0   41 3B 3C 2F 64 61 74 61  3E 0A 20 20 20 20 20 20  A;</data>.
00F144E0   20 20 20 20 20 20 3C 64  65 73 63 72 69 70 74 69        <descripti
00F144F0   6F 6E 3E 73 63 61 6E 20  73 73 6E 3C 2F 64 65 73  on>scan ssn</des
00F14500   63 72 69 70 74 69 6F 6E  3E 0A 20 20 20 20 20 20  cription>.
00F14510   20 20 3C 2F 69 74 65 6D  3E 0A 20 20 20 20 3C 2F    </item>.    </
00F14520   70 72 6F 74 65 63 74 65  64 3E 0A 3C 2F 78 6D 6C  protected>.</xml
00F14530   3E 0A 74 65 64 3E 0D 0A  3C 2F 78 6D 6C 3E 0D 0A  >.ted>..</xml>..
00F14540   2E 22 0A 49 44 53 5F 52  45 42 4F 4F 54 09 09 09  .".IDS_REBOOT...
00F14550   3D 22 52 65 62 6F 6F 74  22 0A 49 44 53 5F 42 49  ="Reboot".IDS_BI
00F14560   54 44 45 46 45 4E 44 45  52 5F 45 58 50 49 52 45  TDEFENDER_EXPIRE
```

```
Page 26709 of 111098    Offset:    F14369    = 34    Block:
```

Figure 7 - Memory view of BitDefender’s bdagent module.

Panda Internet Security 2008

Panda Internet Security 2008 had a module called “Confidential Information Control” that allowed me to enter my confidential information into it. On reviewing the data I entered, I noticed that like BitDefender, Panda IS also masked all my data that was previously entered.

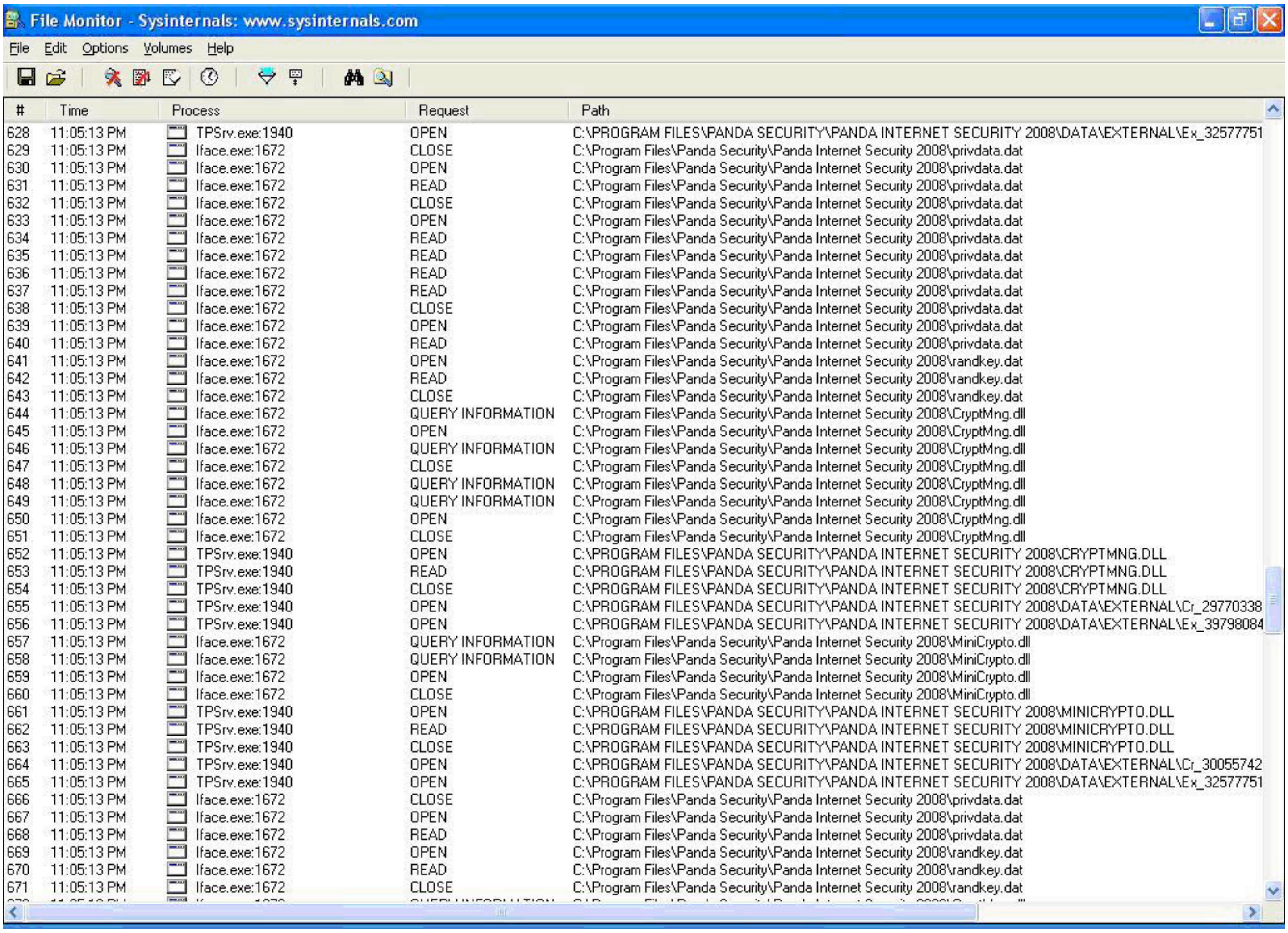


List of confidential information

Type	Description	Information
Credit card	My Credit Card Number	****************
Social Security number	My SSN	*********
Password	Secret Password	********
Bank account	My Bank Acct	************

Figure 8 - Panda Internet Security masking sensitive information.

After taking a look at the process space for any of the confidential information I had entered earlier, I found nothing. Also, while monitoring the file activity that Panda caused when I attempted to forcibly upload confidential data, I could see calls to crypto dlls and the use of private and public keys cryptography.



File Monitor - Sysinternals: www.sysinternals.com
File Edit Options Volumes Help

```
#    Time         Process          Request            Path
628  11:05:13 PM  TPSrv.exe:1940   OPEN               C:\PROGRAM FILES\PANDA SECURITY\PANDA INTERNET SECURITY 2008\DATA\EXTERNAL\Ex_32577751
629  11:05:13 PM  Iface.exe:1672   CLOSE              C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
630  11:05:13 PM  Iface.exe:1672   OPEN               C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
631  11:05:13 PM  Iface.exe:1672   READ               C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
632  11:05:13 PM  Iface.exe:1672   CLOSE              C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
633  11:05:13 PM  Iface.exe:1672   OPEN               C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
634  11:05:13 PM  Iface.exe:1672   READ               C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
635  11:05:13 PM  Iface.exe:1672   READ               C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
636  11:05:13 PM  Iface.exe:1672   READ               C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
637  11:05:13 PM  Iface.exe:1672   READ               C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
638  11:05:13 PM  Iface.exe:1672   CLOSE              C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
639  11:05:13 PM  Iface.exe:1672   OPEN               C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
640  11:05:13 PM  Iface.exe:1672   READ               C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
641  11:05:13 PM  Iface.exe:1672   OPEN               C:\Program Files\Panda Security\Panda Internet Security 2008\randkey.dat
642  11:05:13 PM  Iface.exe:1672   READ               C:\Program Files\Panda Security\Panda Internet Security 2008\randkey.dat
643  11:05:13 PM  Iface.exe:1672   CLOSE              C:\Program Files\Panda Security\Panda Internet Security 2008\randkey.dat
644  11:05:13 PM  Iface.exe:1672   QUERY INFORMATION  C:\Program Files\Panda Security\Panda Internet Security 2008\CryptMng.dll
645  11:05:13 PM  Iface.exe:1672   OPEN               C:\Program Files\Panda Security\Panda Internet Security 2008\CryptMng.dll
646  11:05:13 PM  Iface.exe:1672   QUERY INFORMATION  C:\Program Files\Panda Security\Panda Internet Security 2008\CryptMng.dll
647  11:05:13 PM  Iface.exe:1672   CLOSE              C:\Program Files\Panda Security\Panda Internet Security 2008\CryptMng.dll
648  11:05:13 PM  Iface.exe:1672   QUERY INFORMATION  C:\Program Files\Panda Security\Panda Internet Security 2008\CryptMng.dll
649  11:05:13 PM  Iface.exe:1672   QUERY INFORMATION  C:\Program Files\Panda Security\Panda Internet Security 2008\CryptMng.dll
650  11:05:13 PM  Iface.exe:1672   OPEN               C:\Program Files\Panda Security\Panda Internet Security 2008\CryptMng.dll
651  11:05:13 PM  Iface.exe:1672   CLOSE              C:\Program Files\Panda Security\Panda Internet Security 2008\CryptMng.dll
652  11:05:13 PM  TPSrv.exe:1940   OPEN               C:\PROGRAM FILES\PANDA SECURITY\PANDA INTERNET SECURITY 2008\CRYPTMNG.DLL
653  11:05:13 PM  TPSrv.exe:1940   READ               C:\PROGRAM FILES\PANDA SECURITY\PANDA INTERNET SECURITY 2008\CRYPTMNG.DLL
654  11:05:13 PM  TPSrv.exe:1940   CLOSE              C:\PROGRAM FILES\PANDA SECURITY\PANDA INTERNET SECURITY 2008\CRYPTMNG.DLL
655  11:05:13 PM  TPSrv.exe:1940   OPEN               C:\PROGRAM FILES\PANDA SECURITY\PANDA INTERNET SECURITY 2008\DATA\EXTERNAL\Cr_29770338
656  11:05:13 PM  TPSrv.exe:1940   OPEN               C:\PROGRAM FILES\PANDA SECURITY\PANDA INTERNET SECURITY 2008\DATA\EXTERNAL\Ex_39798084
657  11:05:13 PM  Iface.exe:1672   QUERY INFORMATION  C:\Program Files\Panda Security\Panda Internet Security 2008\MiniCrypto.dll
658  11:05:13 PM  Iface.exe:1672   QUERY INFORMATION  C:\Program Files\Panda Security\Panda Internet Security 2008\MiniCrypto.dll
659  11:05:13 PM  Iface.exe:1672   OPEN               C:\Program Files\Panda Security\Panda Internet Security 2008\MiniCrypto.dll
660  11:05:13 PM  Iface.exe:1672   CLOSE              C:\Program Files\Panda Security\Panda Internet Security 2008\MiniCrypto.dll
661  11:05:13 PM  TPSrv.exe:1940   OPEN               C:\PROGRAM FILES\PANDA SECURITY\PANDA INTERNET SECURITY 2008\MINICRYPTO.DLL
662  11:05:13 PM  TPSrv.exe:1940   READ               C:\PROGRAM FILES\PANDA SECURITY\PANDA INTERNET SECURITY 2008\MINICRYPTO.DLL
663  11:05:13 PM  TPSrv.exe:1940   CLOSE              C:\PROGRAM FILES\PANDA SECURITY\PANDA INTERNET SECURITY 2008\MINICRYPTO.DLL
664  11:05:13 PM  TPSrv.exe:1940   OPEN               C:\PROGRAM FILES\PANDA SECURITY\PANDA INTERNET SECURITY 2008\DATA\EXTERNAL\Cr_30055742
665  11:05:13 PM  TPSrv.exe:1940   OPEN               C:\PROGRAM FILES\PANDA SECURITY\PANDA INTERNET SECURITY 2008\DATA\EXTERNAL\Ex_32577751
666  11:05:13 PM  Iface.exe:1672   CLOSE              C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
667  11:05:13 PM  Iface.exe:1672   OPEN               C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
668  11:05:13 PM  Iface.exe:1672   READ               C:\Program Files\Panda Security\Panda Internet Security 2008\privdata.dat
669  11:05:13 PM  Iface.exe:1672   OPEN               C:\Program Files\Panda Security\Panda Internet Security 2008\randkey.dat
670  11:05:13 PM  Iface.exe:1672   READ               C:\Program Files\Panda Security\Panda Internet Security 2008\randkey.dat
671  11:05:13 PM  Iface.exe:1672   CLOSE              C:\Program Files\Panda Security\Panda Internet Security 2008\randkey.dat
```

Figure 9 - File activity monitor during attempted upload of confidential data.

The following table summarizes the results of my tests:

Software suite security issue	McAfee Internet Security Suite	Norton Internet Security 2008	BitDefender Internet Security 2008	Panda Internet Security 2008
Confidential data echoed back on screen	Yes	Yes	No	No
Confidential data found on process space	Yes	Yes	No	No

Note: McAfee and Symantec were notified in writing about the vulnerabilities in their products on June 2nd and June 3rd 2008 respectively. Both companies have been extremely nice and responsive on following up on the findings. McAfee has attempted to fix some of the issues in their newest version located at beta.mcafee.com. I wish to thank both companies for their prompt responsiveness and handling of the issue.

References

1. “Does Your Security Suite Also Protect Your Privacy?” (tinyurl.com/3fqsbu)
2. Notmyfault - download.sysinternals.com/Files/Notmyfault.zip
3. Pmdump - www.ntsecurity.nu/toolbox/pmdump
4. Winhex - www.x-ways.net/winhex/
5. Idapro - www.hex-rays.com/idapro/
6. Filemon - download.sysinternals.com/Files/Filemon.zip

Shrikant Raman is a security researcher and has over 10 years of experience in information security and privacy issues. He is currently a manager with a leading financial services company in their information security program office and is responsible for identifying any potential threats and vulnerabilities in any new products and offerings among other things. Prior to that, he was a manager at Ernst & Young's Advanced Security Center at Houston performing numerous security and vulnerability testing for clients. Shrikant is also a part time instructor for "The Hacker Academy" teaching classes in hacking and computer forensics. He can be reached at sraman at gmail dot com.

VB2008

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

I Digital Security Forum

7 November-8 November 2008 - Lisbon
www.segurancadigital.org/en/

RUXCON 2008

29 November-30 November 2008 - Sydney, Australias
www.ruxcon.org.au

The Fourth International Conference on Availability, Reliability and Security (ARES 2009)

16 March-19 March 2009 - Fukuoka, Japan
www.ares-conference.eu/conf/

To add your security event to our calendar get in touch by sending an e-mail to general@net-security.org

A proactive approach to data breaches

by Scott Mitic

We live in a world where our personal information is increasingly exposed and vulnerable. The amount of new information stored on paper and online is growing at an alarmingly fast pace. From government agencies to financial organizations to medical offices, countless servers, laptops and databases hold records with your name, Social Security number, financial data, shopping habits, and more. When you consider that the loss or theft of personal data soared to unprecedented levels in 2007, this is causing a major problem that the security industry has been called on to resolve.

A million pieces of data

There are currently over one billion terabytes of data out there, at least two billion file cabinets, 135 million web servers, five billion instant messages, and the world's information continues to grow at the rate of 30 percent each year. Click streams, electronic transactions, cell phones, electronic toll devices, video cameras—these are all "digital breadcrumbs" about consumers that can be used to piece together the various elements of someone's identity.

As awareness of identity theft increases, most people understand that they need to be careful when handing over their sensitive data. We're told to choose smart passwords, delete suspicious-looking emails, use security software to fight viruses, and verify the authenticity of anyone who asks for our personal information. To a degree, we can significantly lower our changes of becoming an identity theft victim by following a few simple rules.

However, today our data is everywhere, making it impossible to be completely risk-free. Personal information resides in homes, offices, computers, doctor's offices, government databases, online databases, and company files. It is stored on credit reports, warranty information, police records, real estate deeds, in marketing and retail databases, financial transactions, driver's records, and a myriad of other places. Every time we hand over a credit card number, send an email, fill out a medical form, or pay a bill, we hand over valuable information and trust that the organizations we do business with will securely process and store our data. At this point, protecting our identity is out of our control.

Unfortunately, sensitive information is not always treated with the greatest of care. Major data breaches occur every day at banks, corporations, and government agencies. Companies like AOL and Google have enormous amounts of information about consumers – but are they keeping it safe? And are they acting in the consumer’s best interests in the way they store, share, and use that information?

The growing data breach problem

The Department of Justice's recent indictment of 11 people for hacking into the databases of nine major U.S. retailers, OfficeMax, Boston Market, Barnes & Noble, Sports Authority, Forever 21, Marshalls, and T.J. Maxx, and the theft of more than 40 million credit and debit card numbers highlights the need for better data loss protection. While the financial consequences of such a breach are bad enough, the damage to reputation and the resulting loss of business can be even more devastating.

This high-profile incident is just one of many corporate data breaches that potentially exposed American consumers to identity theft.

Every time we hand over a credit card number we hand over valuable information and trust that the organizations we do business with will securely process and store our data. At this point, protecting our identity is out of our control.

Other recent examples include:

• A U.S. Transportation Security Administration vendor reported that a laptop, containing unencrypted personal records of 33,000 customers seeking to enroll in the company's Registered Traveler program, was stolen.

• A laptop containing the names, birthdays and Social Security numbers of more than 26 million military veterans was stolen from the home of an employee at the Department of Veterans Affairs (VA).

• A data intrusion into the Hannaford supermarket chain’s network exposed nearly 4.2 million credit and debit cards and led to 1,800 reported cases of fraud.

• Harvard University notified 10,000 applicants that their Social Security numbers and other personal data may have been accessed by hackers through a file-sharing site.

These breaches highlight an ominous problem as commerce increasingly moves to the Internet, where there are countless opportunities for identity thieves to use stolen personal information for financial gain. And the problem is not limited to online shoppers. Today, most brick-and-mortar retailers store data on computers that connect to the Internet. For organizations that maintain large databases of personally identifiable information, data breaches raise the possibility of identity theft and other violations of privacy.

Fortunately, in many data breach cases, the consequences - financial and otherwise - of the breach are slight. However, many lead to identity theft, which is one of the fastest growing crimes in the United States. More than 27 million Americans became victims of identity theft between 2003 and 2007, and it took millions of hours and dollars to repair the damage. In the United States, an identity theft currently occurs once every two seconds.

Red flagging, shoulder surfing, dumpster diving, midnight mailing, skimming, phishing, pharming, vishing and data breaches have all become part of our daily reality. Identity theft is no longer a paranoid concern – it’s a burgeoning epidemic.

These numbers continue to grow, as corporate and government leaders continue to take a "reactive" approach to protecting identities. Most state data breach laws require companies to send notices to all persons whose data is likely to have been compromised. The problem with this approach is that the action comes after the breach. Companies are spending millions of dollars on notifications, but failing to invest in security mechanisms to prevent fraud.

The changing face of storing and sharing

As businesses expand globally, corporations are storing data in more places than before, including on laptops, mobile devices, Internet, laptops, and personal computers.

This means that corporate data is being used, and exposed to risk, in more places and in more ways than ever before. For identity thieves, this opens a whole new world of opportunity to use sophisticated techniques and schemes that take advantage of security vulnerabilities.

As this information increasingly comes online, the problem is exacerbated. Personal data is now available to more individuals and organizations, including government agencies, private corporations, and even identity thieves. Even pieces of data you might think is private, like unpublished phone numbers, are now being bought and sold online by data brokers. Privacy experts say many of the methods for acquiring such information are illegal, but business continues to thrive.

While providing access to your phone records is not as dangerous as providing access to your Social Security number when it comes to financial fraud, your phone records say a lot about your personal associations.

For example, a fraudster can learn who you talk to—your mom, your boss, your doctor, your accountant—and use that information to collect even more information about you, and eventually commit a crime.

The data breach problem isn't expected to turn around anytime soon. Companies are relying on security experts to help them fight back against fraudsters.

Security professionals today are in a unique position to help organizations move from a reactive approach to a proactive, comprehensive approach in resolving the data breach problem plaguing consumers and enterprises.

Under the current model for storing and sharing consumer information, consumers carry the burden of risk while companies benefit from a surplus of information about us. Today, via a simple Google search, people have access to endless amounts of information about us. For example, Senator Ted Stevens recently ordered his staff to steal his identity.

They came back not just with digital breadcrumbs on the senator, but also with information on his daughter's rental property and his son's activities. "For $65 they were told they could get my Social Security number," he said. Stevens' experiment proved what privacy advocates have been saying for years: all it takes to get someone's personal information is Internet access, some spare time, and a little cash.

Initially, most consumers didn't understand what kind of personal data would be of interest, how it would be used, or why they should care. However, nowadays people are increasingly aware of the risks and benefits, and are started to take notice. The broad availability of our personal data is causing huge shifts in marketing—companies are moving away from random direct mail toward truly personalized, cell-phone presented, and geo-targeted advertisements.

The bottom line

A recent survey indicates that 79 percent of consumers cite trust, confidence, damage to reputation, and reduced customer satisfaction as consequences of major security and privacy breaches suffered by the business or government organizations that they deal with.

Because they don't trust these organizations to adequately protect their data, the majority say that they want more control over their private information.

As a result, more and more people are signing up for the “Do Not Call” list, and demanding opt-in marketing and better privacy controls from the companies with whom they do business.

Today, a shift is afoot in identity theft protection, as more people move away from reactive credit monitoring and toward a more proactive approach—credit freezing. This trend has many implications in different areas, such as financial management, reputation management, and more.

With the proliferation of the web, mobile devices and social networking sites, what will our identity look like in a few years? Today, identity thieves are after our Social Security, bank account and credit card numbers, but what will they be targeting in the future? How can we best manage our reputations? These are all questions that organizations are looking to security professionals to address.

The bottom line is that traditional security measures are ill-equipped to measure up to consumer and regulatory compliance demands. Simply stated, companies need to do more to protect consumers against data intrusions. For this reason, security is becoming a top IT priority.

“Five or six years ago, security represented seven-tenths of one percent of all IT spending, which is a very large number,” said Art Coviello, president of RSA, the security division of EMC. “Last year, it was one-and-a-half percent—more than double. And that’s one-and-a-half percent of a much larger number than five years ago. At the trajectory we’re on, it will double again probably within three or four years.”

Businesses are faced with a plethora of new data monetization opportunities and an even greater range of IT and security issues to resolve, including open data stores, data aggregation transparency, protection of information, data portability, user permissions, retention, and purge policies.

A recent Department for Business Enterprise and Regulatory Reform biennial security survey showed that businesses are indeed taking data security seriously. In fact, 77 percent of companies said they regard protecting customer information as a top priority. However, only eight percent said they encrypt data stored on laptops.

The fact that most people still do not feel that their data is secure demonstrates the need for a new approach to security. Businesses are faced with a plethora of new data monetization opportunities and an even greater range of IT and security issues to resolve, including open data stores, data aggregation transparency, protection of information, data portability, user permissions, retention, and purge policies.

For security professionals, these trends have a massive impact, as they portend an increase in investment—and job security. As organizations move from a reactive to a proactive approach to data protection, the security industry as a whole is seeing unprecedented opportunities and support for improving the ways companies protect consumer information in all stages of its lifecycle, as well as the different formats in which data is stored.

The question is—how will we rise up to this challenge?

This article was written by Scott Mitic, CEO of TrustedID, the leading provider of identity theft prevention solutions. To learn more about how to protect yourself and your family from identity theft, please visit www.trustedid.com.

Compliance does not equal security but it's a good start

by Jack Danahy

As more and more organizations have moved credit card transactions and point of sale infrastructure to the Internet, credit card providers and customers have become increasingly exposed to attacks on their private information. Recognizing the growing market for stolen credit card credentials and foreseeing the potential damage from the accelerating pattern of theft and abuse of consumer data, major credit card issuers and banks partnered to develop the Payment Card Industry Data Security Standard (PCI).

The standard, issued in January of 2005 sought to increase security and privacy among applications brokering credit card transactions and to drive an increased awareness of best practices in the development and deployment of these applications and infrastructures. To date, much of the interest and activity around PCI has been driven by requirements for third party assessment of compliance, the potential application of fines, and the aggressive timelines. Over the past three years, PCI has evolved and been clarified, and many of the deadlines that were put in place for vendors to meet have since passed, including the most recent in June of this year.

Looking back at the results of this effort, it is obvious that one goal of PCI - the increased awareness of the importance and vulnerability of customer data - has been accomplished. State privacy laws, Federal regulation, and industry best practices, are each reflecting on the purpose, recommendations, or framework of the standard. However, that being said, frequent public losses of credit card data, massive breaches with corresponding damage, and an ongoing lack of consistency in interpretation of PCI and its requirements is still occurring. As a result, the debate surrounding PCI has moved from whether or not such a standard is necessary to whether this standard goes far enough in helping to make data secure.

PCI requires all organizations that store, process or transmit credit card information to demonstrate compliance with twelve categories of data security requirements, ranging from application-specific characteristics to infrastructure and deployment management criteria like networking, access control, and audit policies. While compliance audits often leave many of the details open to interpretation, PCI provides some guidance and a minimal set of requirements for critical areas designed to improve the state of protection for the privacy and security of customer data.

There has been an unfortunate and unintended consequence as a result of the focus on PCI compliance - the simple assumption that by being compliant, you are also secure. The decline in emphasis on security in favor of focus on compliance has created a climate in which passing an audit or satisfying a regulator is deemed to indicate a sufficient focus and effort on doing what's necessary to protect critical assets. This, as is seen on a regular basis through breaches, is a dangerous assumption. The reality is that PCI compliance is by no means synonymous with security.

Real and practical security is about balancing risks and costs. Organizations deploy the amount of security they feel is necessary to prevent catastrophe, while working within budgets, and within acceptable thresholds of risk. This is because security, more than almost any other technical discipline, suffers from the 90/10 rule. 90% security can be had for roughly 10% of the cost of 100% security. (To make matters worse, it would be virtually impossible to find a single credible security expert who would attest to the existence, much less the attainability, of 100% security.) Therefore, organizations must understand and balance the risks among systems, attack types, and potential damage, with the very tangible cost of prevention and the always-elusive factor of likelihood.

REAL AND PRACTICAL SECURITY IS ABOUT BALANCING RISKS AND COSTS

This is the landscape and reality of security, a system of checks and balances to ensure the best protection that is attainable in a real environment of investment and return. The current imposition of requirements arising from PCI compliance has thrown this unstable system into further chaos. The reason is that there has been the introduction of a new risk, the risk of being “out of compliance” with PCI. In many organizations, this risk is more pressing, and more persistent, than actual breach risks, because of an axiom once coined by Dr. Hugh Thompson, who said, “You may or may not be hacked, but the auditors will always come.”

This confusion between PCI compliance and security is natural. Many of PCI requirements look like security measures, and the overall goal of PCI is to decrease the likelihood of one class of security events. It is a common result to have PCI required areas over-emphasized, and a lack of the natural balance between investment of security and reduced external risk. However, this line of thinking leads to a mistaken perception - when all the dollars have been spent, that full compliance will lead to full security and that the areas outside of PCI are of less importance to overall organizational security, which is, in most cases, patently untrue.

What is required is a more measured approach to PCI compliance in light of the overall security of the organization. It should never have been an exercise on its own, but rather an advancement and acknowledgement of internal security practices already in place, to secure the credit card related applications and infrastructure. A report on PCI compliance should be a lens through which a particular set of threats and countermeasures are assessed, however, it should not be an independent goal, with an expectation that fuller security strength is achieved within it.

Letter of the standard vs. spirit of the standard

While PCI is still a work in progress, it has been the driving force to a healthy debate about data security best practices. Even the PCI Council acknowledges that the standard requires refinement and that it is a growing and changing document. Containing relatively dynamic content, it is naturally subject to individual interpretation, and that interpretation is helping to better define the “best” in “best practices", and to raise the level of discourse of a variety of important security topics.

Conflicts in interpretation, and in some cases enlightened self-interest, lead to ongoing debates and clarification. These debates often center on the contention between the exact language of the standard and the inferred spirit of the standard.

A recurring example of this exists in the recently clarified Section 6.6 of PCI DSS Version 1.1. The section originally stated:

"Ensure that Web-facing applications are protected against known attacks by applying either of the following methods:
• Having all custom application code reviewed for common vulnerabilities by an organization that specializes in application security
• Installing an application layer firewall in front of Web facing applications"

To an external observer, the letter of the standard is clear. If you analyze all application code, you are in compliance, or if you install an application layer firewall, you are in compliance. At the most superficial read, this resolves the issue, and many organizations have pursued one or the other, exclusively, in order to meet compliance guidelines.

This is unfortunate for their overall security, because, in fact, the key to this section of the standard is in the first sentence. The section is requiring application protection from all known attacks, and the two named approaches can or should be used in that pursuit. A clarification, issued in April of 2008, has done little to clear the issue among most organizations.

If organizations view the real purpose of PCI as protection of data, and then recognize that this is a natural and important part of their security program, then the interpretation becomes more clear, and much more useful. One such translation is: "Ensure that Web-facing applications are protected against known attacks. Based on the type of application, and the type of vulnerability, one or more of the following approaches should be used to ensure appropriate coverage."

Different types of vulnerabilities require different protective measures, and addressing this mix of requirements requires a palette of technologies.

Complying with the requirement to prove a negative

Part of the issue with many industry regulations is they require organizations to prove they have not done something wrong, irresponsible or insecure with their customer data. This requires the fundamentally difficult task of proving a negative. As an example, PCI requires that certain data never be stored and that other data always be encrypted. Even large retail and financial organizations, with dedicated security personnel, may not immediately realize that data has been mishandled until a security breach has occurred. In some public cases, data thieves had access to millions of credit card numbers for months or even years before the organization was capable of detecting the problem.

While establishing a network perimeter and monitoring traffic will always be essential components of any security policy, organizations cannot assume that this will ensure their security, unless they also understand what is happening to their data. This is why PCI released a "Data" security standard. Protection of data requires that organizations identify, measure, and track data progress in the application building blocks. It is this requirement that leads to recommendations for the additional protections in Section 6.6, and specifically to the recommendations for examination of those building blocks: the source code.

There are a number of reasons organizations deploy software in which data is mishandled. Sometimes, software has been developed by outsourced or off-shore partners with insufficient definition of security requirements. Similarly, in internal development, many organizations have not embedded secure development as criteria in the accepted software development as criteria in the accepted software development lifecycle. As a result, additional effort and additional inspection is necessary to provide security with peer status to other common software development concerns such as quality, performance, and functionality. To complicate the issue even further, simply developing secure code is not a sufficient response. Although developing secure code is necessary, and would be a significant step in the right direction, most enterprise applications are not under active development, but rather, are comprised of legacy, open source, or reused code that is unexposed to the quality assurance (QA) process. There is additional effort necessary to examine these components, external to software development lifecycle performance.

With the emergence of Web services and other SOA features, legacy code modules are further exposed when they are repurposed to serve needs and clients never envisioned by the original designers. The result of this leverage is an exposure of increased application vulnerabilities that were not even considered vulnerabilities when the code was originally developed.

Discover, prioritize and secure

Compliance audits typically test if existing network-based approaches (intrusion detection systems (IDS), intrusion prevention system (IPS), or firewall applications) are properly working. These tools do not speak to the security of the data but rather to the permeability and security of the network. The appropriate and necessary evaluation must include application analysis, and should provide information to security analysts, managers, auditors, developers, and executives. The investigation must look beyond the pipes that connect the application, but actually within the applications themselves.

Buried within millions of lines of code that can comprise an organization's backbone, undiscovered vulnerabilities lurk and continually pose a threat. According to NIST (National Institute of Standards and Technology), more than 93 percent of reported vulnerabilities are software vulnerabilities that can expose organizations to risk of an attack. Most organizations lack the resources necessary to proactively locate and remediate these threats. Organizations need a way to analyze the software itself, allowing security risk managers and auditors to identify those threats so they can be isolated or eliminated.

While examining source code can be difficult, the good news is that today's software analysis tools make the process easier. By shortening the time required to locate software vulnerabilities, identification and verification can be manageable. By creating a system where organizations can get actionable results in hours, not days, security analysts can gain valuable insight into the location of software vulnerabilities. They can then ascertain the nature of the vulnerabilities, the risks and impacts if those vulnerabilities if they were exploited, and ultimately, offer remediation advice to the developer. Thanks to compliance regulations such as PCI, often-ignored software vulnerabilities are finally becoming an area of focused attention. By identifying and managing these risks, organizations can meet and even exceed compliance security requirements while also protecting customer data and corporate reputations.

PCI compliance is obviously a goal for concerned organizations, and it provides a new view into applications and their security. It provides organizations with concrete guidelines for the protection of credit card account data, and brings a sense of business perspective to typically abstract issues of risk and loss.

Organizations are often quick to complain that PCI is both too specific, and too vague, but organizations need to stop using excuses for their lack of proper security. The PCI standard can be used to improve a company's security postures, and can generate interest and support for security measures beyond the usual IT community. It is not, however, an appropriate objective or sole outcome for security programs and practitioners. It should be one view, one snapshot, of a set of business practices that exist in harmony with broader and more encompassing security measures. It is not a roadmap to a secure organization, but it is an important, and illuminating, signpost along the way.

Jack Danahy is founder and CTO of Ounce Labs (www.ouncelabs.com) as well as one of the industry's most prominent advocates for software security assurance. Jack is a frequent speaker and writer on information security topics and has been featured many times in print, TV, trade, and national security events. He holds multiple patents and has patents pending in systems analysis, software security assessment, kernel security, secure remote communications, systems management and distributed computing.

Secure web application development

by David Rook

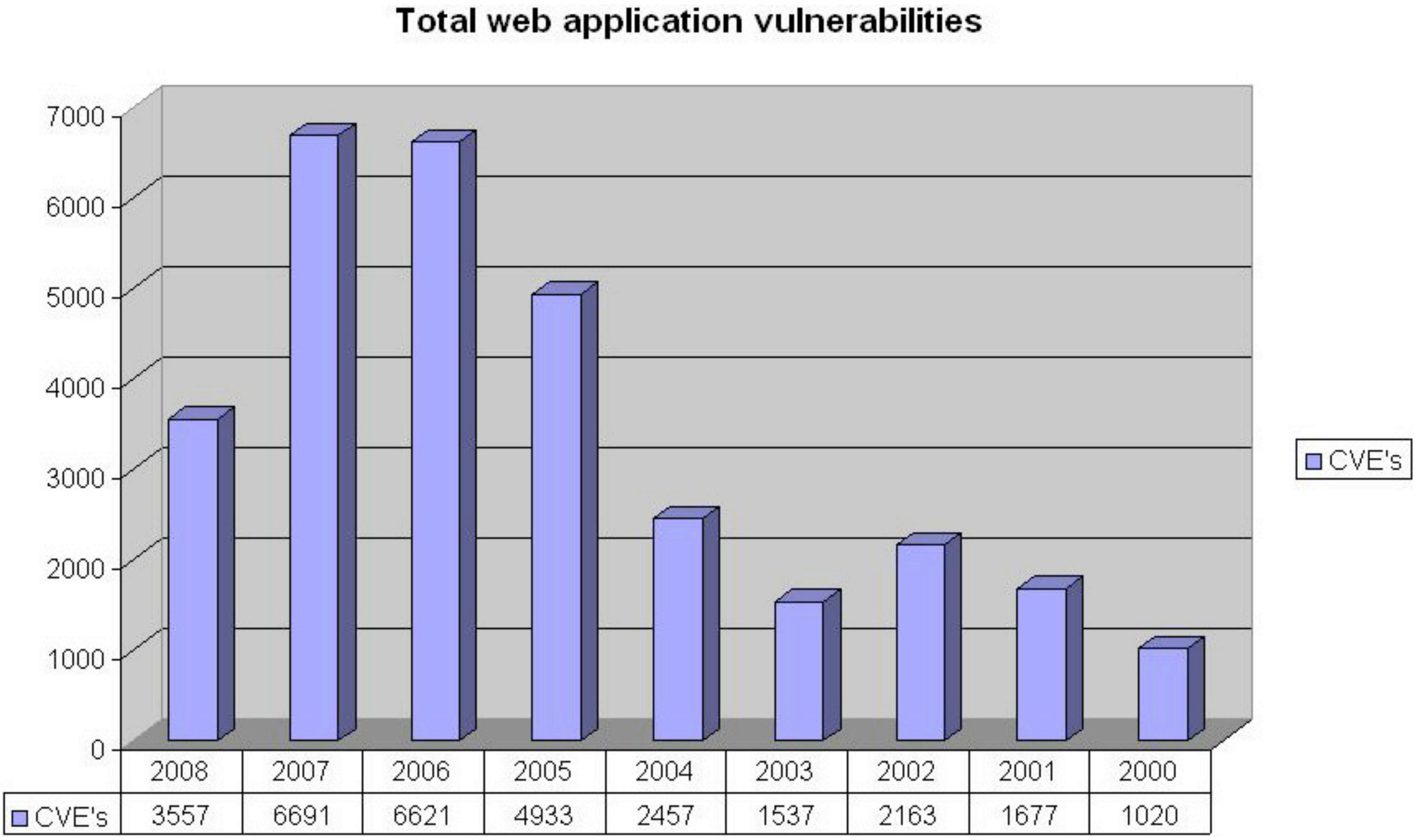
Why? That one word question is probably the most common question I have to answer when I talk about secure web application development. Why should we build security into our development lifecycle? Why use threat modeling? Why do I have to sit through security awareness training? Application security is not black magic, and I hope this article will answer the "Why?" questions and help you integrate security into existing software development lifecycles.

Software has always had security holes and this will probably always be the case. From the Morris Worm through to vulnerabilities in SCADA systems we have witnessed vulnerabilities in every type of software imaginable. Traditionally a company's biggest security expenditure and concerns would be at the operating system and network layers. This has led to substantial investments at these layers which has improved security to a level where they are no longer the weakest link in the security chain. Attackers will always target the weakest point and in 2008 that means the application layer.

I have never been one to try and scaremonger people into addressing information security issues, I prefer to present facts about security issues and give economic and business impact assessments of them. With that statement in mind I want to show you that web application vulnerabilities really are becoming the attacker's choice in 2008.

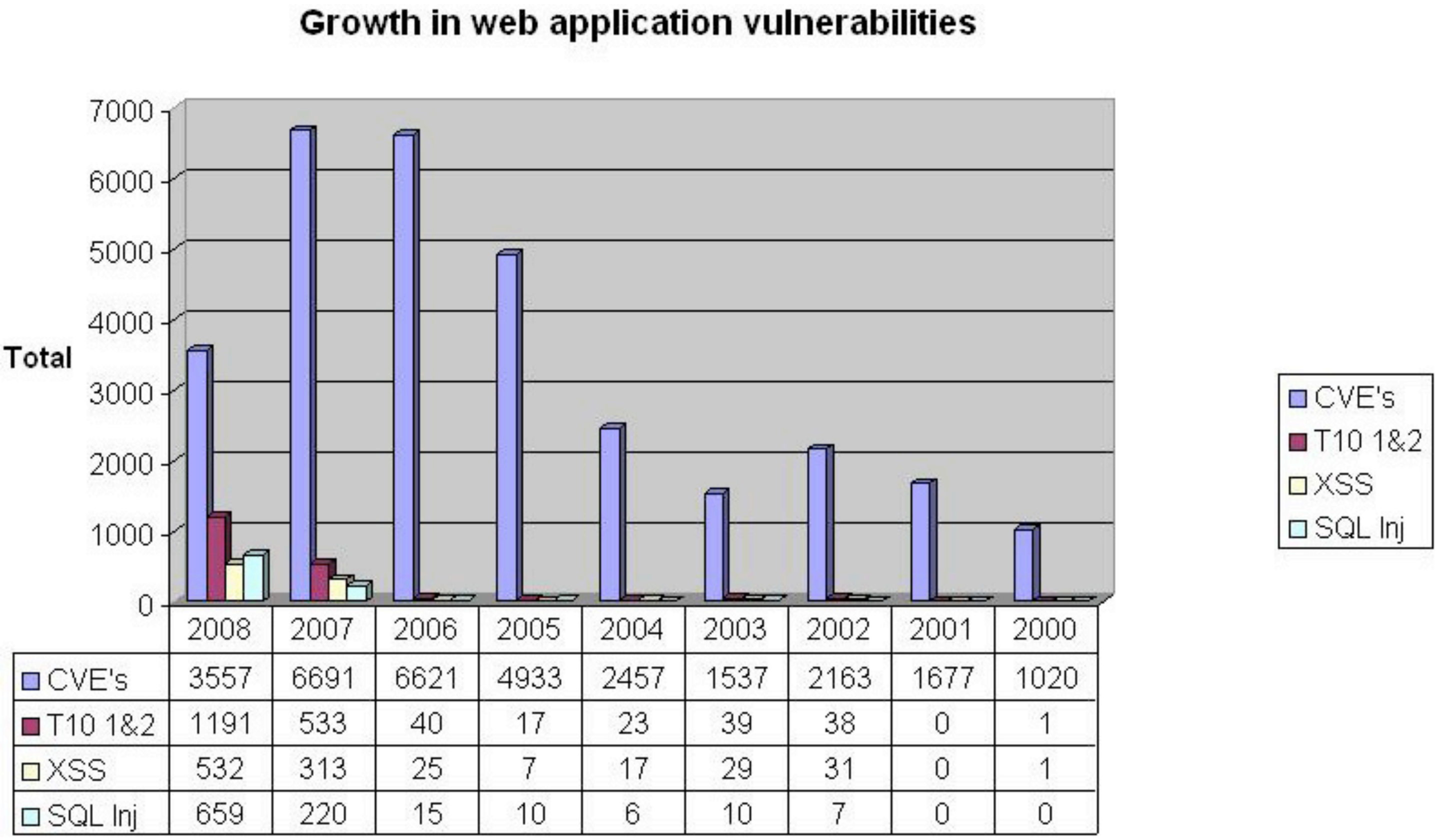
The first graph on the following page shows the amount of CVE (Common Vulnerabilities and Exposures) numbers issued between January 2000 and August 2008. A CVE number is usually issued to publicly known security vulnerabilities and they provide a good indicator for plotting trends in the number of security issues discovered annually.

You can see that the number of CVE numbers issued has risen dramatically from 2003 through to 2007. The CVE statistics database allows you to specify the types of vulnerabilities you wish to filter by.



	2008	2007	2006	2005	2004	2003	2002	2001	2000
□ CVE's	3557	6691	6621	4933	2457	1537	2163	1677	1020

To demonstrate the increase in web application vulnerabilities I have provided a second graph below. The second graph shows what percentage of the total CVE's map to common web application security issues, namely Cross Site Scripting and SQL injection:



	2008	2007	2006	2005	2004	2003	2002	2001	2000
□ CVE's	3557	6691	6621	4933	2457	1537	2163	1677	1020
■ T10 1&2	1191	533	40	17	23	39	38	0	1
□ XSS	532	313	25	7	17	29	31	0	1
□ SQL Inj	659	220	15	10	6	10	7	0	0

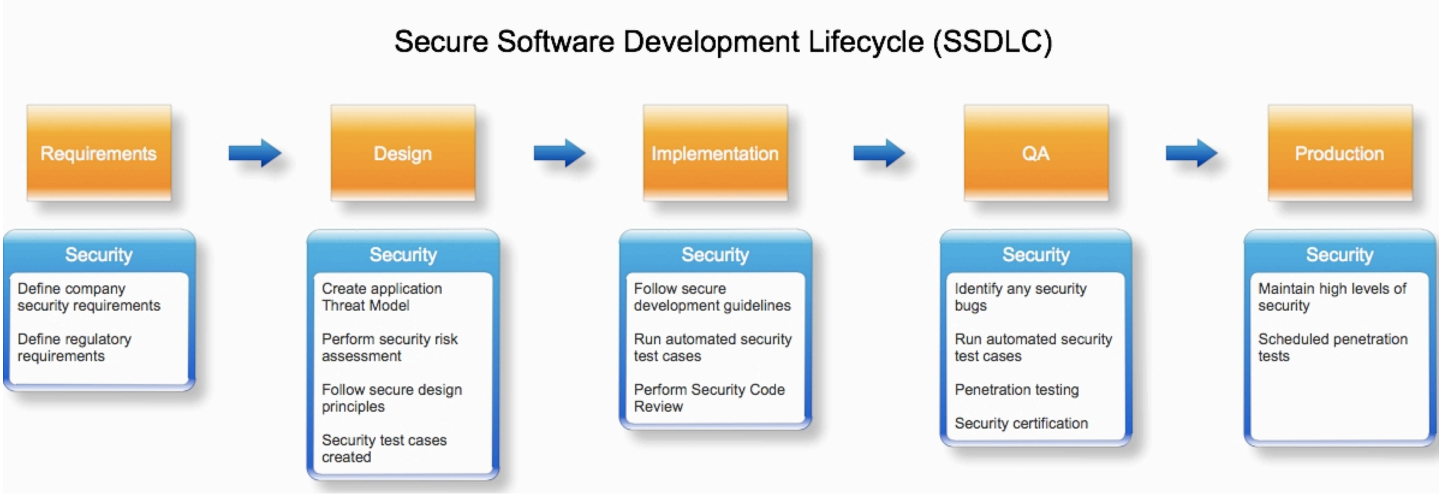
This increase in web application vulnerabilities can be seen by selecting many other web application security issues. I have selected Cross Site Scripting and SQL Injection as examples because they the top two issues in the OWASP Top Ten and have experienced a dramatic increase. You can clearly see the growth in Cross Site Scripting (17% of 2008 CVE's at the time of writing) and SQL Injection (20% of 2008 CVE's at the time of writing) over the past 3 years. In 2005 Cross Site Scripting and SQL injection accounted for 0.20% of the annual total. You can view and query the CVE statistics at nvd.nist.gov.

How to improve web application security

The best way to build secure applications is to integrate security in to the Software Development Lifecycle (SDLC). This will create an SSDLC (Secure Software Development Life Cycle) which will serve to increase the security of applications and reduce the cost of fixing security bugs. The common approach to security in the development lifecycle is to treat security as a separate process.

This approach to development will perform security testing at the end of the development lifecycle meaning that fixing any bugs found at this stage comes with a high cost.

Below you can see an example of an SSDLC:


Secure Software Development Lifecycle (SSDLC)
Requirements
Design
Implementation
QA
Production
Security
Define company security requirements
Define regulatory requirements
Security
Create application Threat Model
Perform security risk assessment
Follow secure design principles
Security test cases created
Security
Follow secure development guidelines
Run automated security test cases
Perform Security Code Review
Security
Identify any security bugs
Run automated security test cases
Penetration testing
Security certification
Security
Maintain high levels of security
Scheduled penetration tests

This diagram shows how security can be added to each step of a traditional development process.

Every development should have security considered at the requirements stage just as functional and business needs are. The requirements need to be relevant to the technology being used and clear enough for designs to be written from them. The security requirements should ensure that any applicable industry (for example PCI DSS) and company requirements are built into the application. Treating security requirements in the same manner as functional and business requirements will ensure that security is treated as an important component of the application design.

The design should clearly indicate how the application will meet each of the security requirements laid out in the requirements phase. The application design should focus on core security principles such as least privilege and fail-safe defaults. The security decisions taken at the design phase must be clearly documented and implement an open design. This will ensure that the application isn't following the principle of security through obscurity.

A security professional should be able to identify potential security issues at this stage before any code is written, for example if the application plans to use a deprecated encryption algorithm this can be identified here. Identification of these types of issues at the design stage has an obvious cost benefit for the development.

The security professional should follow a well defined process to identify any security issues in the proposed design. A common approach would be threat modeling which is the approach I take at this stage of a development lifecycle. Threat modeling will decompose the application so the reviewer can identify important information such as entry points, trust levels and most importantly the potential threats to an application.

The identified threats will be given a risk score and an associated test case to ensure the potential threat is tested for and subsequently eliminated during the implementation phase.

The implementation phase will begin once a design and threat model have been produced and agreed. The implementation phase will be largely hands off from a security professional’s point of view. The developers will be following secure development guidelines that have been created based on security best practice and industry examples such as Microsoft's SDL.

The guidelines should be built around company security policies and standards making it clear to architects and developers how to build security into the application and adhere to the relevant policies and standards. The developers should also utilize the test cases that have been developed as part of the application threat model. The application threat model was created at the design phase and test cases will have been produced for the threats that have been identified. The test cases should be run by the developers during the implementation phase which can identify potential security issues at a point in the lifecycle which has a low cost associated with code fixes.

Once the developer has finished writing the application the security code review must be completed before the application moves on to the QA phase. The security code review can take a few different paths and I have detailed my personal approach in this article. I find it useful to have a checklist to begin the security code review process; my personal checklist has over 50 individual checks covering the following areas:

- Input Validation
- Authentication
- Authorization
- Configuration Management
- Sensitive Data
- Session Management
- Cryptography
- Parameter Manipulation
- Exception Management
- Auditing and Logging

I haven't listed all of the individual checks here but a good example of a security code review checklist can be found in the Microsoft Threats and Countermeasures guide. The checklist will allow the reviewer to perform a repeatable code review for every application that is reviewed. The checklist should immediately identify any areas of the application which might not meet the required levels of security.

Once the checklist has been completed a manual code review should be conducted. One could argue that an automated approach should be used but I feel automated code reviews need a manual review to compliment them and understand the code in context. I prefer to perform manual code reviews starting from the areas of risk identified by the application threat model. An obvious starting point would be the entry points that are listed in the application threat model. The review of these entry points should consider what data do these entry points receive, how the application performs validation on the data that has been received and what happens if an error occurs when malicious data is received through this entry point.

This isn't an exhaustive list but you can see the kind of questions a reviewer will need to ask and receive an answer for during a security code review. Once the security code review is complete a report should be produced detailing how the code meets these high level secure development principles:

- Configuration Management
- Data Protection in Storage and Transit
- Authentication
- Authorization
- Session Management
- Data Validation
- Error Handling
- Auditing and Logging

I will also include a section in the report detailing any of the relevant checklist items that this application has an answer of "no" for. Some of the checklist items won't be relevant to every application. The final section of the report should detail any security issues that have been identified by the security code review.

The issues should be explained clearly and be accompanied by a risk rating so the business can evaluate the risk associated with not addressing the issue.

The application can enter the QA phase once the security issues that were identified have either been accepted or removed. The QA phase shouldn't find major security issues, major security issues should have been identified in the requirements and design phases and at worst in the security testing during the implementation phase.

The testers should run the same test cases that the developers executed in the implementation phase and verify that the application passes these tests. Security testing during this phase should become as natural as functional testing and easily become part of existing testing procedures. A suitable member of the security team should perform a penetration test against the development during this phase prior to the development receiving its security certification.

Once the application has been through this process it should have a good level of security built into it. A schedule should be produced to ensure that this application isn't "forgotten" once it goes live.

The applications we develop today may not protect against the future threats we are likely to face and with this in mind the application must be re-visited on a regular basis (As an example I recommend revisiting applications that process sensitive data every 6 months) for a threat model review and penetration test.

The two things we are trying to achieve by doing this is to answer two questions:

1) Does the threat model still accurately depict the security of the application?

2) Does the application protect against current threats?

Experience has shown me that the best way to ensure that the code is written securely is to implement a good security awareness program. There isn't a shortcut to getting this right but you have to spend time with developers explaining what the threats their applications are likely to face are and how they can mitigate them. Security awareness training for application developers can take many forms and isn't as simple as training developers to write secure code. I appreciate that statement might sound a little odd so I have listed a few points below which I feel comprise a good security awareness program for developers:

1) Understand common vulnerabilities (i.e. OWASP top ten) and how to mitigate them.

2) Train developers to use security testing tools and to interpret the results of testing.

3) The re-use code that has been security certified.

4) Understand how security fits into the development lifecycle.

5) How to conduct security focused code reviews.

Once developers have been through an awareness program which covers issues like the ones mentioned above then the implementation phase should produce secure code. A mistake often made by companies is assuming developers automatically know how to develop securely and not providing sufficient training based on this assumption.

If you take secure development seriously and adopt the principles I have outlined in this article you should begin to produce secure applications. In 2008 anyone who develops web applications cannot bury their heads in the sand and think they won't appear on hacker’s radars.

With approaches such as Google hacking, you might become someone's target sooner than you think.

David Rook is a Security Analyst for Realex Payments in Dublin, Ireland. Realex Payments is a leading Europen Payments Service Provider enabling thousands of businesses across Europe to accept payments in multiple currencies across multiple channels; including websites, call centres and mail order. David is the creator of securityninja.blogspot.com and is a contributor to several OWASP projects including the code review guide. David has presented at IT security conferences on the topic of secure application development.

Cute Password Manager Pro (www.net-security.org/software.php?id=721)

Cute Password Manager (CPM) is a free form filling software that auto fill userID and password. CPM stores your web logins on your local machine with 256-bit AES encryption and performs a true "one click login" for you. It is a fast, easy and secure password manager. One 'Master Password' is all that is needed to access all your passwords and private information.

audit daemon (www.net-security.org/software.php?id=702)

The audit package contains the user-space utilities for creating audit rules, as well as for storing and searching the audit records generate by the audit subsystem in the Linux 2.6 kernel. It also has a basic Intrusion Detection plugin based on audit events capable of IDMEF alerting using prelude.

IPSecuritas (www.net-security.org/software.php?id=599)

IPSecuritas lets you easily setup IPSec VPN connections to another host or network over the Internet, while securing your data by encryption and authentication. This way, you can easily and cheaply access your office network from any point of this world, always knowing your communication is safe and protected from others.

ntop (www.net-security.org/software.php?id=36)

ntop is a network traffic probe that shows the network usage, similar to what the popular top Unix command does.

Avoiding a ‘keys to the kingdom’ attack without compromising security

by Stephane Fymat

In Europe, very few people have heard of Terry Childs. In California, everyone has. Childs is the City of San Francisco’s disgruntled network manager who reset all administrative passwords to the routers for the city’s FibreWAN network and held the city administration to ransom. He refused to hand over the passwords which effectively gave him complete control of the network, locking out all other employees and preventing anyone else from administrating it.

As San Francisco legal teams try to get to the bottom of how the now notorious Childs was able to gain so much control, IT managers around the world are working out how to prevent the same thing happening to them.

The complexity of corporate IT systems require users to memorize more and more passwords: surveys have found that 36 per cent of users have between six and 15 passwords to remember, a further 18 per cent have more than 15 unique identifiers to memorize.

Research from the Burton Group suggests that the average user can spend up to 15 minutes every day logging on to separate application – which adds up to 65 week days spent entering user IDs and passwords each year.

Almost every user has personally experienced password frustration: the inability to remember the details for an important application when they needed it and the delay in getting the password reset by the IT help desk. Gartner estimates that 25 to 35 per cent of calls made to IT help desks are password related at an estimated cost of around £15 - £20 a call, adding millions to the support bill at larger companies.

Aside from the lost productivity, the excessive administrative overhead and the user frustration, passwords can actually present a significant security risk. In an effort to jog their memories, users will often create passwords that are easy-to-figure out - such as derivatives of names and birthdays - making it all-too-easy for hackers to gain access to enterprise applications and data.

Concerns about ineffective password systems and lax password security that enables unauthorized users to breach enterprise networks have caused corporate regulators to take a tougher stance on password security. The Sarbanes-Oxley Act for example, includes specific clauses on password security. Nonetheless, there are people, including Bill Gates, who question their benefit and long term future.

But the problem does not lie with passwords themselves - it is how they are managed and the lack of best practice in how they are deployed. The latest generation of enterprise single sign-on technologies (ESSO) overcomes the inherent weaknesses of passwords. ESSO eliminates the need to remember - and therefore the risk of forgetting - and is the most effective antidote to the problem of password overload.

ESSO enables users to sign in once with a single password and access all their applications, databases and systems. They no longer need to remember or enter individual passwords for all those applications, so they gain immediate access to corporate information in a more secure, controlled environment. ESSO automates the process of password entry by responding to each log-in prompt without user intervention. New passwords can be automatically generated when old ones expire, and the user ID and password for every application can be stored in a secure central repository.

Quite aside from the very quantifiable savings that can be made in help-desk costs, the benefits of ESSO to the enterprise include simplified administration, improved enterprise security and greater user productivity, all while retaining the ability to achieve compliance with regulations on data protection, privacy and corporate governance.

New passwords can be automatically generated when old ones expire, and the user ID and password for every application can be stored in a secure central repository

Why isn't it more widely used?

ESSO has often been seen as too costly and labour intensive to ever be truly attainable. But the latest advancements in the technology mean that its time may finally have come. Traditionally, one of the biggest criticisms of ESSO has been that it makes an organization vulnerable to a single point of attack. The reality is that ESSO provides a higher degree of security. There is no user involvement so password quality rules can be more easily enforced, for example. Password length and complexity and the frequency at which they are changed can be greatly increased making them much more difficult for a hacker to decipher. Since users do not need to remember each password, unique, complex alphanumeric combinations of any length, case or format can be created for each application, database or account log-in. Mathematicians have proved that if the length of a password is increased from 8 to just 9 characters, the time to crack the password is increased to 447 years.

Even in the unlikely event of a hacker cracking the password, they would still need access to a workstation with ESSO software on it, or alternatively install software on a workstation themselves. Even then it would require specific knowledge about how to install and configure the ESSO software with the target organization's directory.

But the problems associated with passwords are not limited to the fallibility of users' memories and the determination of hackers.

The Terry Childs incident illustrated another problem that has largely passed under the radar at most companies, who place an enormous amount of trust in their IT staff and system administrators. There was only one administrative account on many systems at San Francisco. Childs had open access to system passwords, and so was able to change them without authorization and lock out his colleagues. It's not an uncommon scenario – but it is an unavoidable and unnecessary one.

The most advanced ESSO software now includes shared and privileged user management capabilities. This enables all administrative passwords to be encrypted and stored in the enterprise's central directory. Administrators must check out a password from the directory in order to use it - and can be approved or denied based upon the administrator's role and manager's approval within an identity management system. If approved, the software will log the administrator on to the network device and check the password back in automatically – the administrator never knows the password. The software will also keep a history of passwords for each network device. If network devices must be restored from backup, the then-current password can be retrieved.

Had this system of shared management capability been in place at the City of San Francisco, Terry Childs would never have been able to hold the City administration to ransom in the way that he did. The lesson from San Francisco is that there is a need for an effective alternative to basic password systems, which offers much greater control and security around access to enterprise networks. The number of application passwords that must be managed in many enterprises today is untenable, undesirable and unsafe.

The bottom line is simple: passwords no longer provide adequate protection. ESSO is a proven solution that removes the burden from both end users and administrators, and simultaneously hardens the network against attack through strengthened password policies. The Terry Childs incident highlights the need for greater control over administrative passwords – and the role that ESSO can play in protecting organizations against sabotage by insiders.

If we are to avoid a repeat of what happened to the City of San Francisco, widespread adoption of ESSO with shared and privileged user management needs to be seriously considered.

Stephane Fymat is the VP of Business Development and Strategy at Passlogix (www.passlogix.com), the developer of the v-GO Sign-On Platform, an enterprise single sign-on platform with successful installations in hundreds of organizations of all sizes and in all industries around the world.

The insider threat

by Sacha Chahrvin

The demands of the modern workforce are changing rapidly. It's now a mobile business world and we expect to conduct our work whenever we want, wherever we want.

Laptops now outsell desktops, wireless is outpacing wired and your average smartphone can do almost anything – even if you only use it to make calls and as an occasional alarm clock on a business trip.

Not so long ago - when businesses were solely run out of an office - it was easy for employers to keep track of their staff and know that everything from the stationary to their confidential information was kept under one roof.

Nowadays, staff can work wirelessly and remotely, and as business becomes more global we have to adapt to the fact that employees expect to work with a myriad of different appliances and gadgets – many of which are capable of storing anything from customer databases to family albums.

The trouble with all this mobility is that it's not secure, data seems to fly through the air between devices – it's no longer tangible. Because of the demands of modern working practice, it is becoming increasingly difficult for IT managers to adequately protect company information. The standard anti-virus and network access control is not enough nowadays. Mobility, in all its weird and wonderful forms, jeopardizes business security - and it's a growing problem.

Recent research has revealed that UK companies trail behind those in Germany and the US in the implementation of policies to prevent data leakage. It also showed that UK end users are less likely to know what type of information is confidential and rarely receive training on data policies.

There is a growing concern that IT networks are becoming too vulnerable to threat from the very thing that they are trying to incorporate – the remote device. The proliferation of iPods, smartphones, PDAs and USB sticks mean that most employees now have personal devices that can store huge amounts of data.

These devices are virtually impossible to trace and can be connected to a laptop or PC with ease. Incidents of employee data theft is constantly growing: 1.6 million personal details were stolen from Monster.com, 800,000 were stolen from GAP and the UK Government saw a whopping 15 million individual records stolen in 2007.

From these figures it’s clear that this type of threat does not focus on any specific industry, it can happen to any organization at any point.

And it’s not just your standard Blackberry or USB stick that pose a risk. Over 26,000 different USB products currently exist - from coffee warmers to network adaptors. And these devices are to become more sophisticated and more readily available – there are already 10GB appliance available in the shops for under £30.

A survey of more than 1,000 UK workers found that 60 per cent admitted to theft of confidential documents, customer databases, business contacts or sales leads. Sixty-three per cent said there were no restrictions on using personal portable devices such as USB memory sticks in the workplace. So how do IT managers start to manage the security threats that are raised from these devices?

Vulnerability assessment

It is important to assess where the business is vulnerable. For some companies it is often a certain group of employees that use mobile devices on a regular basis, such as a sales team. Pinpointing areas in the business such as this, where there is a much greater chance of hardware being lost or stolen for example, means that you can focus your plan of action accordingly.

It is important to assess where the business is vulnerable.

Policy

The UK government has recently been roundly criticised over its handling of sensitive data following a new report by the Joint Committee on Humans Rights. Its recent incidents of data loss were considered to be "symptomatic of lax standards."

Many of the other large companies that appear in the news who have experienced an incident of data leakage have undoubtedly already got a data security policy in place, just like the government. But the fact is that a policy document buried in the hard drive and a few well-placed posters lecturing on the ‘enemy within’ are fairly pointless. Data loss is either on purpose or by accident, so there needs to be a concerted effort, through training and seminars, to convey the importance of data protection and the legal implications of data theft.

Reduce and limit access to data

Restricting who can access what information can help to control the movement of important data. The easier data is to copy, the harder it is to control so, making sure that the right levels of access are being granted to the right people is important. Encrypting data on mobile devices is also a useful measure.

Controlling data

In the US, many companies do not allow staff to enter the workplace with personal devices that have storage capacity. This is becoming an increasingly common way for businesses to be proactive in stopping employees from being tempted to copy data onto their MP3 player or mobile phone. But it is not fail-safe. Investment in technical controls in order to monitor and prevent data being copied and printed without a trace should be the key ingredient of the strategy in managing the threat of data loss.

Endpoint data security enables businesses to allow staff to carry sensitive data in laptops and USB sticks without making data access inflexible and protracted. And this is the balance that IT departments are looking for. The workforce demands easily accessible data at the touch of a button, and the IT department would ideally like sensitive data to be totally secure - which would be impractical for modern working. Additional password authentication will help control who accesses certain systems, and endpoint security software can secure the company's hardware from theft, or malicious attack through a USB port.

Mobile devices that have become so integral to our personal and business lives are a reaction to the fact that our personal and working lives have become so much more mobile over recent generations. Just as manufacturers have adapted to this shift with devices and gadgets that help us run our busy lives it is important that we adapt to protect the information that we now carry around with us.

It is not necessarily a struggle for IT security to keep up with all these gadgets and devices, but it is a struggle for them to keep up with how we choose to use those items. Educating employees to try and alter their habits is vital as long as it coincides with the implementation of user friendly security measures such as endpoint security, two-factor password authentication or even James Bond style tracking technology for the most forgetful!

Sacha Chahrvin is the Managing Director at DeviceLock UK (www.devicelock.com), a worldwide leader in endpoint device control security.

Security videos

The Vulnerability Economy

(www.net-security.org/article.php?id=1157)

Jeff Moss, the founder of DEFCON and Black Hat, discusses the unfolding of the vulnerability economy. Nowadays, instead of exposing high profile zero-day vulnerabilities at conferences, many researchers opt for selling their discoveries on a growing market.

DTrace: The Reverse Engineer's Unexpected Swiss Army Knife

(www.net-security.org/article.php?id=1167)

In this video, made at Black Hat Europe, security engineer David Weston illustrates his research related to DTrace. Created by SUN and originally intended for performance monitoring, DTrace is one of the most exciting additions to OS X Leopard and is being ported to Linux and BSD.

It offers an unprecedented view of both user and kernel space, which has many interesting implications for security researchers. Many of the features of DTrace can be leveraged to discover new exploits, unobtrusively monitor malware and even protect against buffer overflow attacks.

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

Web application security: risky business?

By Rob Faber

There's an overwhelming number of opportunities for you to create your own web portal, or do business on the Internet using a variety of web services. On the other hand, since securing web applications is still not as widespread as it should be, your online business may be vulnerable to attacks. The ease with which a web application can be broken is staggering, and there should be more attention given to some of the known weaknesses, insecure programming problems and misconfigurations that plague web applications.

A couple of months ago, I was surfing the Internet in search for a holiday home where I could vacation this summer. I visited several well known websites to explore their offerings, but when searching one of them in particular I encountered problems after accidentally typing in a specific string. I made a mistake, but the result instantly caught my attention.

A simple search of the sort that most people make will be performed innumerable times each day, and this is where the story, and this article, begins.

The site I'm referring to is still in use, and is a real-life example of how vulnerable such websites can be. Websites like this one usually host a web front-end that will has a connection to a database. This database contains, and conceals, your private data, such as your credit card information, postal address, and order information. If your website is not properly secured, all of these details could be within easy reach of an attacker. This article will focus on the common vulnerabilities of web applications and, in particular, on the very popular platform on which they are built - PHP.

Security is still not always incorporated

Web applications are almost everywhere and continue to win the hearts of IT and business managers for a number of reasons. Yet, the latest figures have revealed that about 85 percent of all web applications contain one or more weak spots that can be targeted. Why?

Web application attack techniques are fairly easily to understand, and there are lots of useful white papers, vulnerability databases, and other tools that can automate harvesting and attacks. This makes it easy to investigate a website to try to discover the flaws and weak spots in the application's logic and manipulate application input. Compared to other kinds of more sophisticated attacks, the level of knowledge required to attack many web applications is not extensive. What is more, firewalls are not enough to stop an attacker because, in most cases, you will have to allow inbound HTTP/S traffic, and the underlying service is still within reach via standard port 80. Combined with easily accessible web development platforms like LAMP (Linux / Apache / MySQL / PHP), most web applications are assembled by developers who have little experience when it comes to secure development.

The HTTP protocol itself does not implement a robust authentication and authorization mechanism as part of its overall architecture. All of these factors together create a big problem, and can unwillingly reveal sensitive information to the public, or worse. A common misunderstanding for customers using a web application: using SSL won't solve your problems!

Most common problems with web security

There are resources that we can use to get an overall view of the most common (critical) flaws found in web applications. According to OWASP (The Open Web Application Security Project) which is an open-source application security project, some of the main web application security issues are:

- Invalidated (user) input
- Remote code execution
- SQL injection
- Broken access control
- Broken authentication and session management
- Format string vulnerabilities
- Cross Site Scripting (XSS)
- Username enumeration
- Session Hijacking
- Cookie manipulation
- Buffer overflows
- Improper error handling
- Insecure storage
- Application denial of service
- Insecure configuration management

That is quite a long list, and yet it is just the tip of the iceberg. Each of these types of attack represents a variety of vulnerabilities.

OWASP's goal is to inform, and to make web developers and administrators aware of the risks. It is crucial that you learn about known weaknesses and the types of attack that can occur. It is also important to note that web applications can be subjected to many more forms of attack than those listed above.

Let us have a look at a few very popular web application attacks, namely SQL injection. The prime focus will be on PHP applications, and providing an example of a vulnerable website. The examples will relate to PHP coding because of its popularity on the Web. The more important message concerns the concepts that apply to any programming languages.

The case: hack the way in

Although web servers work with the HTTP protocol, database servers understand the SQL language better. The web server will connect to the database on numerous occasions to get to the useful information stored inside it. When users authenticate themselves, the web application collects important information such as an e-mail address and password, or a user ID and password. The application takes these parameters and creates an SQL query that will be used to get information from the database. The web server's connection to the database might be established just once, every time communication is necessary, or maintained for a longer period in connection pools.

The web server will use its own system account name and password to authenticate itself to the database. The web server then passes these credentials on to the backend database server (which is found in the SQL statement). The database accepts the statement, executes it, and then responds with a result. It is up to the application on the web server how it handles this response from the database.

SQL injection is still popular as it allows an attacker to retrieve crucial information from a web server's database.

Depending on the application's security measures, the impact of this attack can vary from basic information disclosure to remote code execution and total system compromise. SQL injection vulnerabilities can arise in any application parameter that influences a database query. This includes URL parameters, POST data, and cookie values. Needless to say, these aspects need to be tested to determine if a vulnerability is present.

The discovery of this flaw in the website I visited was a complete coincidence. The website has several ways of allowing for user-input which is then used to build a query which is submitted to the underlying database, thereby retrieving and presenting that information to you. We can distinguish between different types of SQL injection. It is possible to directly manipulate the URL of the specific webpage, and it will look like this on the address bar of your browser:

http://www.housebooking.org/search.php?nperson=5&holiday=1w&month=06&year=2008&refid=1234567

Within the PHP code, this information will be used to make a query of the database. However, before discussing this further I first want to present you with another fact.

HTTP methods

HTTP defines the different methods of prescribing the desired action to be performed on a resource. Some of these methods are: GET, HEAD, POST, PUT, and DELETE.

When web applications are attacked, the two methods that are used most extensively are GET and POST. There are some important differences between these methods and which can affect an application's security. The GET method is designed for the retrieval of data. It can be used to send specific parameters to the requested resource by incorporating it into the URL string. URLs are displayed on-screen in the address bar of the browser, and are logged, and can be found in the browser history. They are transmitted to another site in the referrer header when external links are followed.

For these reasons, the query string should not, under any circumstances, be used to transmit sensitive information like login IDs.

The POST method is designed for the performance of actions. Request parameters can be sent both in the URL query string, and in the body of the message. These parameters will be excluded from logs and do not show up in the referrer header. Because the POST method is designed for performing actions, if a user clicks the “back” button of the browser to return to a page that was accessed previously, the browser will not automatically reissue that request. Instead, it will warn the user.

This prevents users from unwittingly performing an action more than once (for example buying an item in a web store). For this reason, POST requests should always be used when an action is being performed.

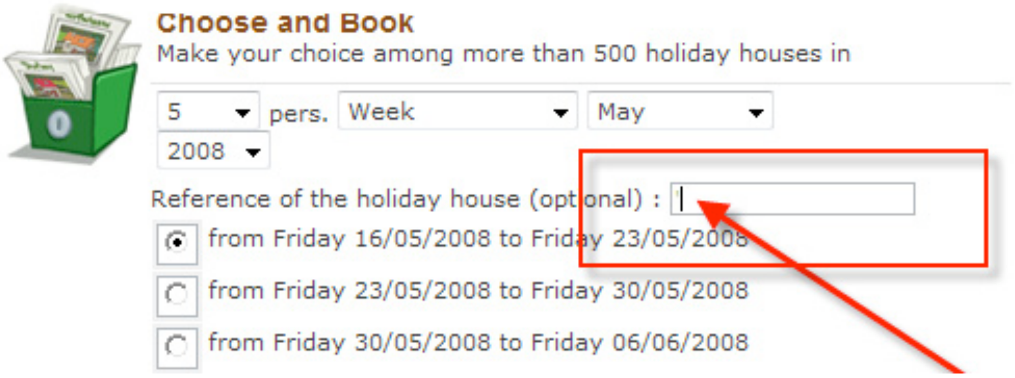
In our example, imagine you searched for a specific holiday home with the reference number “1234567”, in June 2008 for 5 people for 1 week. The “refid” presented here will probably be “$refid” in php code. But what if the developer didn’t check this kind of input properly? Let us assume that he did not. In these circumstances it would be possible to directly manipulate the URL in the address bar, and not input a reference number for “$id” but something else instead. For example:

http://www.housebooking.org/search.php?nperson=5&holiday=1w&month=06&year=2008&refid=letsputinsomebogushere

Or something like this:

http://www.housebooking.org/search.php?nperson=5&holiday=1w&month=06&year=2008&refid={12345 OR refid = 98765}

As you can imagine, the results of doing this can be devastating. Another possibility of SQL injection is to manipulate the input data that is submitted by typing characters in an input box. In this scenario, there is no proper input validation in place, so it is possible to extend or manipulate the query string that is sent to the database.



Choose and Book

Make your choice among more than 500 holiday houses in

5 ▾ pers. Week ▾ May ▾ 2008 ▾

Reference of the holiday house (optional) :

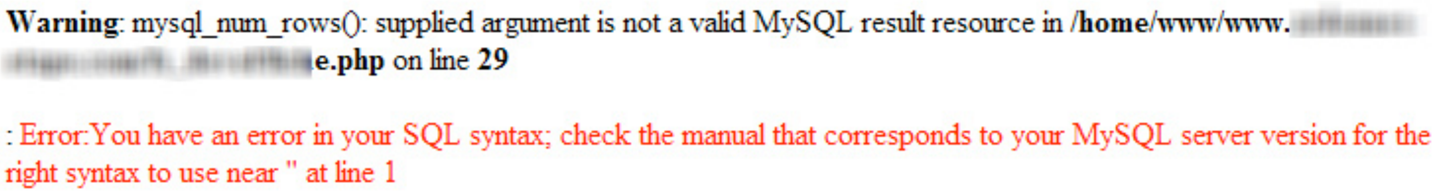
● from Friday 16/05/2008 to Friday 23/05/2008

○ from Friday 23/05/2008 to Friday 30/05/2008

○ from Friday 30/05/2008 to Friday 06/06/2008

Figure 1 - The search input box.

A brief investigation provides us with some clues about the behavior of the application. After typing in a wrong reference number, the application in this case had no proper method of error handling and returned the message you will find in figure 2.



Warning: mysql_num_rows(): supplied argument is not a valid MySQL result resource in **/home/www/www.[redacted]e.php** on line **29**

: Error:You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 1

Figure 2 - The error message.

By now I became very curious, and refreshed my knowledge about SQL injection techniques. How does this work? A PHP code can contain SQL statements that are used to retrieve information from the MySQL database. This code is part of the HTML code and is surrounded by brackets: <?php ..... ?>. In this way it is obvious that this is the PHP code which needs to be executed. Most of the time, the PHP code will be something like this:

```
<form action="retrieveobject.php" method="POST" />
<p>Objectcode: <input type="text" name="objectcode" /><br />
<input type="submit" value="code" /></p>
</form>
<?php
$hsql = "SELECT * FROM houses WHERE objectcode = '{$_POST['objcode']}";
$result = mysql_query($query);
?>
```

Basic SQL injection

In these circumstances, the application returned an error code that is not sanitized, suggesting that the MySQL query was incorrect. It seems that some characters had a negative effect on the whole SQL query. The script will only work normally when the input does not contain any violating characters. In other words, when submitting a normal input code, the query would be something like this:

```
$hsql = "SELECT * FROM houses WHERE objectcode = 'xyz'";
```

In my holiday home example, this is not what happened. Thus, the question arises: is this application vulnerable to attack? The answer is a definite yes! The next step is then to manipulate the query string with the so-called “control characters”.

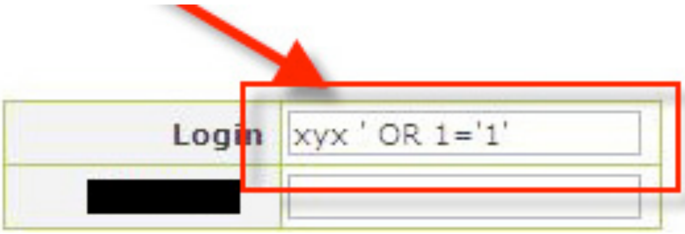
As you will have noticed, the query input is between “ ‘ “ and “ ‘ “. What would happen if we could change this and include something that extends the query and would dramatically change the results? We would then be able to retrieve information from the database as we wished. By typing the following in the input box:

```
xyz ‘ OR 1=’1’ /*
```

we can directly manipulate the results. Consider the query that is used behind the scenes. The developer expects that, at this point, a visitor will provide some valid input such as a particular house’s code. This code is then submitted (within the complete query) to the database. If we can extend this line with valid SQL code, and there is no correct user input validation, the concatenated query string is handed over to the database engine. In the previous example, this would look something like this:

```
$hsql = "SELECT * FROM houses WHERE objectcode = 'xyz' OR 1=’1’";
```

The final element of the input string “/*” are also control characters. Everything that follows these will be ignored (they are treated as comments).


Login xyx ' OR 1='1'

[redacted]

Figure 3 - The manipulated input box.

As the "OR" condition is always true, the mysql_query function returns records from the database. Now that we have this result, it is clear that an attacker can browse the data retrieved from the database with two goals in mind: to discover interesting information, and if possible get the layout of the different tables in the database. The result of what I tried was more detailed customer information.

One step ahead

If it is possible to get a database’s layout, or a glimpse of the underlying structure, this would help a hacker to harvest even more information. One of the techniques utilized in the area of SQL injection is to use the “UNION” operator. In the previous examples, only information about holiday homes was presented, which is not very interesting to someone who wants more sensitive information.

What if we could manipulate this query and try to get information about other underlying tables in the MySQL database? What if the database contains other tables, which contain information about holiday home owners, customers, or IDs and passwords?

The work can be done with the UNION operator. We can use the UNION operator within SQL to “glue” two separate SELECT queries together. There are some restrictions. The numbers of columns of both queries and the field type (string, integer) have to be the same.

Such a query would look like this:

```
"SELECT housename, code, housedescription FROM houses WHERE objectcode =
'1234567' UNION SELECT username, userid, password FROM users WHERE use-
rid = 1";
```

In this way, it is possible to retrieve very sensitive information from the database. Naturally, you could automate this process by writing a script and then get all user IDs with their corresponding information.

Remote code execution with SQL injection

The previous examples are only intended to directly hit your database in search for useful information. But what if an attacker wants to gain access and use your machine as an attack platform? This can also be accomplished by using SQL injection techniques.

In this type of attack, the attacker's intention would be to get control of a machine and to, for example, upload some specific tools (files) to the SQL Server and then execute a tool like netcat. Netcat can then be used to set up a connection from the inside out. Most firewalls accept this kind of traffic from inside the perimeter.

Countermeasures

We have touched upon some known vulnerabilities of web applications in this article, with the focus being on SQL injection and input validation. The conclusion is that it is insufficient to have a firewall solution in place, because the attacks are aimed at the web application's logic and the database server or middleware. SSL and the little "lock" provide no guarantees. Instead, every malicious action undertaken against the application will be hidden.

The only possible solution to overcoming this kind of vulnerability, or to mitigate your risks, is to really test your code thoroughly. This can be done by reviewing the code and also by having the application attacked in a brute force manner using fuzzing techniques. Incorporate secure programming principles from the very beginning of a project.

Finally, sanitizing the input is crucial. In this way, it is almost impossible to perform SQL injection in the first place.

Bear the following in mind for PHP:

- Avoid connecting to the database as a superuser or as the database owner. Always use customized database users with the bare minimum of privileges required to perform the assigned task.
- If the PHP magic_quotes_gpc function is on, then all the POST, GET, COOKIE data is escapes automatically.
- Keep things up to date and patch, patch, patch.
- PHP has two functions for MySQL which sanitize user input: addslashes (an older approach) and the mysql_real_escape_string (the recommended method). This function appears in later PHP versions, so you should first check if this function exists, and whether you are running the latest version of PHP 4 or 5.

Finally, please take note of the following. If you are interested in reading more about this topic, there are a number of excellent white papers, books and other initiatives for you to investigate. At the very least, please visit the OWASP portal (www.owasp.org). Information about PHP security can also be found in places like phpsec.org/php-security-guide.pdf.

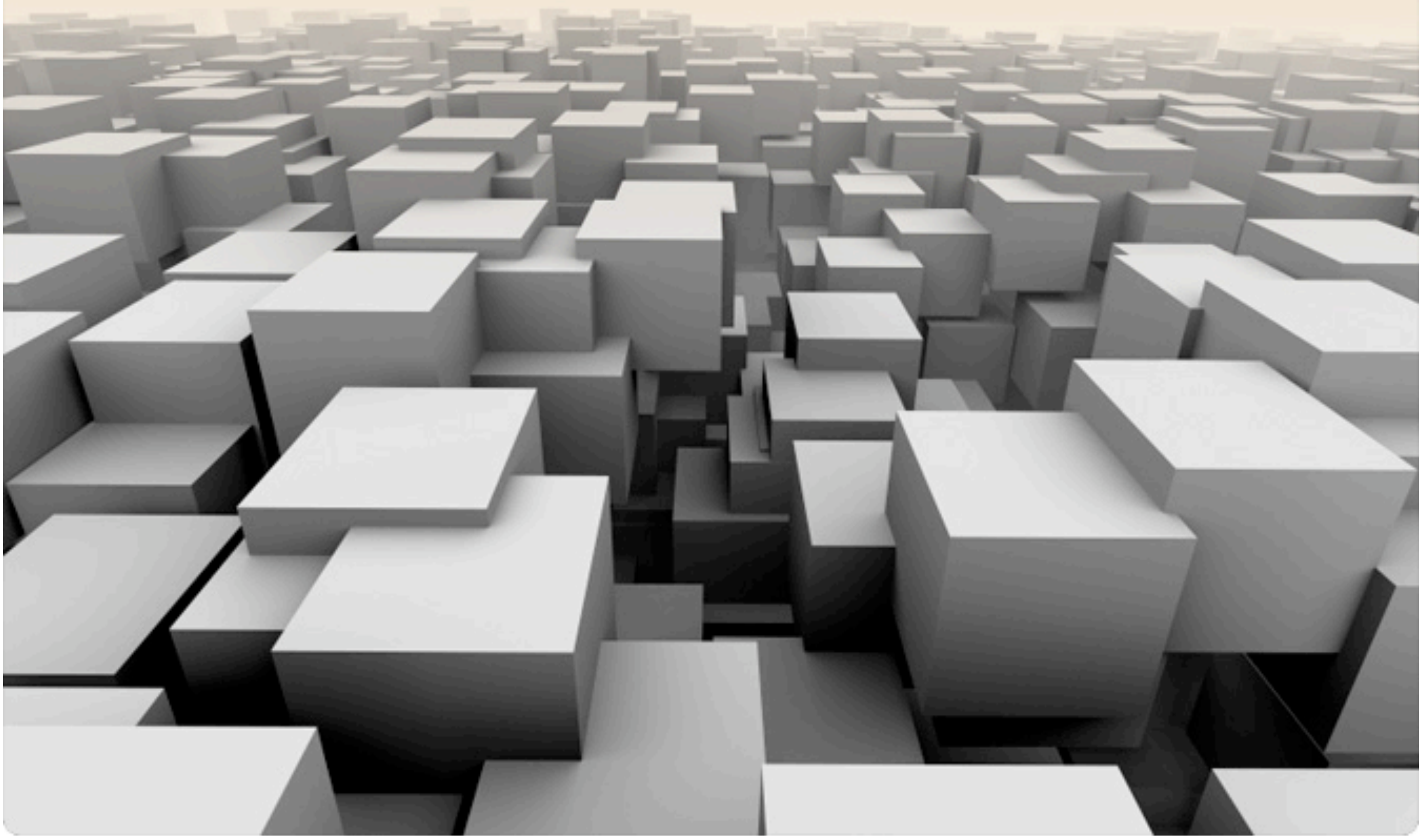
Conclusion

I presented a real example of known vulnerabilities. As a professional - having no doubt at all - I informed the owner and organization behind the website used in this article immediately. While respecting the ethics of the my profession I handed over all information and didn't go any further as described in the article. I described the vulnerabilities and advised them to solve this problem as fast they could. Sadly, months after I reported this, some vulnerabilities still exist...

Rob P. Faber, CISSP, CEH, MCTS, MCSE, is an information security consultant. He currently works for Atos Origin, a global company and international IT services provider based in The Netherlands. His specialization and main areas of interest are Windows platform security, ethical hacking, Active Directory and identity management. He maintains his own weblog at www.icranium.com. You can reach him by e-mail at rob.faber@atosorigin.com or find him on the LinkedIn network.

Enterprise application security: how to balance the use of code reviews and web application firewalls for PCI compliance

By Ulf Mattsson



Organizations handling credit cards feel pressure building as the deadline for PCI Requirement 6.6 compliance [1] has passed and well documented breaches have heightened the public and regulatory agencies' concerns about how well companies are securing consumer-specific information.

Despite some initial advances, sensitive information is still frequently stolen. Internal threat an issue, magnified by extended partnerships which ultimately lead to more tasks will be performed outside company facilities. In increasingly complex technical and business environments, no one security approach can deal with all the new and innovative intrusions. However, the lack of a security silver bullet doesn't mean data security is impossible. It simply means that businesses have to take a multi-pronged approach to data security.

This article is based on a project case study in protecting an enterprise application environment, including web-oriented applications. The article is PCI 6.6-oriented and compares the use of Web Application Firewalls (WAF) or code reviews for web-facing applications. It also addresses code scanning that is not web related. Extending the code reviews into the non-web applications, we also briefly discuss other types of protections. Other articles already discussed how to protect from SQL Injection into the database, or internal threats, including a DBA that impersonates a user.

The section "Protecting the data flow" includes a few pointers to resources discussing protection of the enterprise data flow. The code review section is longer since this is an evolving area from a PCI perspective focusing on WAF and complementary code scanning.

This article will compare WAF and web-based code reviews, and point to resources [15] discussing the whole data flow, which then involves much more than C/C++ code scanning.

The part concerning code analysis is not web-oriented, but it's about C/C++/Java source code scanning, though it has some general parts.

The case study - company ABC

The case study from company ABC recommended using both WAF and code reviews. Internal and external users are accessing sensitive client and credit card data via web based applications and other types of applications. ABC is considering Web applications as #1 focus of an attack. ABC reviewed recent research that shows that the majority of cyber attacks are performed at the Web Application level. ABC considers that their e-business Websites are at immediate risk of being hacked. ABC's primary issues are PCI compliance, and a concern about the escalating threat against financial data from organized crime and insiders.

Time is a critical factor in selecting solutions to prevent breaches. ABC is a security aware organization that will need both short term and long term solutions to this problem. The case study from company ABC will analyze and identify an approach to develop and maintain secure systems and applications, including selecting suitable static-analysis code scanning tools for application development. ABC positioned different approaches to prevent data theft (and the attack-paths to the data – different types of apps, databases) including WAF, data protection (encryption, hashing, tokenizing) and C++ code scanning. The solution for ABC is based on the conclusion that every layer of defense is critical. A holistic and layered approach can provide the best level data security and the sooner sensitive data gets encrypted, the more secure the environment. ABC is planning an enterprise data protection approach and protects data across the information life cycle.

ABC acknowledges that secure development will take a long time to implement, partly based on expensive and time-consuming manual code reviews. The short term solution is based on protecting the external web-facing applications with a WAF combined with data encryption in files and databases. This will give ABC a quick and cost effective data security implementation that will meet PCI requirements in this area. ABC is complementing this with a medium term solution including code reviews and scanning of internal code non-web applications. ABC also identified a long term project that will include penetration testing and scanning and review of the web application code base.

Payment Card Industry (PCI) Requirements

PCI Requirement 6 - Developing and maintaining secure applications

Payment Card Industry (PCI) Data Security Standard (DSS) Requirement 6, Develop and maintain secure systems and applications. PCI 6.6 itself has two halves, “code review” (the act of finding/fixing vulnerabilities) and “application firewalls” (device designed to thwart website attacks) that merchants may choose between.

Fixing custom application code is not easy

Requirement 6 is about “developing and maintaining secure applications and systems.” Requirement 6.1 requires that vendor-supplied security patches be applied within one month of release. Securing and fixing custom application code is not as easy as downloading a patch from your favorite software vendor.

Web application vulnerabilities must be identified, fixes developed, tested, and deployed. In short, you're on your own for the entire process. Setting aside the fact that these two options should not be perceived as competitive, rather complementary, the Council is giving merchants the choice acknowledging budget constraints.

PCI Requirement 6.6 mandates the following:

PCI DSS version 1.1 Requirement 6.6: Ensure that web-facing applications are protected against known attacks by applying either of the following methods. Having all custom application code reviewed for common vulnerabilities by an organization that specializes in application security. Installing an application layer firewall in front of web facing applications.

Testing procedure for web-based applications

PCI DSS version 1.1 Requirement 6.6 Testing Procedure: For web-based applications, ensure that one of the following methods are in place as follows. Verify that custom application code is periodically reviewed by an organization that specializes in application security; that all coding vulnerabilities were corrected; and that the application was re-evaluated after the corrections. Verify that an application-layer firewall is in place in front of web-facing applications to detect and prevent web-based attacks.

The confusion stems from the interpretation of the requirement. First, let's clear up some high-level misconceptions. Requirement 6.6 is not just for “level ones.” It does not specify service providers or merchants nor specify either source code reviews or web-application firewalls.

Complying with Requirement 6.6

Requirement 6.6 is about protecting web applications, plain and simple. Given our modern threat landscape, it is no wonder that PCI Requirement 11.3.2 dictates “application penetration tests” be performed after every “significant change.” Meaningful web application security management requires frequent assessments as code and threats evolve continually.

Requirement 6.6 is about developing a repeatable methodology that connects the “Find” (the vulnerability detection) process to the “Fix” process for the systematic, efficient elimination of vulnerabilities from web applications. Additional practical PCI guidance can be found at [16].

MEANINGFUL WEB APPLICATION SECURITY MANAGEMENT REQUIRES FREQUENT ASSESSMENTS AS CODE AND THREATS EVOLVE CONTINUALLY

Testing procedure for web-based applications

PCI DSS version 1.1 Requirement 6.6 Testing Procedure: For web-based applications, ensure that one of the following methods are in place as follows. Verify that custom application code is periodically reviewed by an organization that specializes in application security; that all coding vulnerabilities were corrected; and that the application was re-evaluated after the corrections. Verify that an application-layer firewall is in place in front of web-facing applications to detect and prevent web-based attacks.

The confusion stems from the interpretation of the requirement. First, let's clear up some high-level misconceptions. Requirement 6.6 is not just for “level ones.” It does not specify service providers or merchants nor specify either source code reviews or web-application firewalls.

Complying with Requirement 6.6

Requirement 6.6 is about protecting web applications, plain and simple. Given our modern threat landscape, it is no wonder that PCI Requirement 11.3.2 dictates “application penetration tests” be performed after every “significant change.” Meaningful web application security management requires frequent assessments as code and threats evolve continually.

Requirement 6.6 is about developing a repeatable methodology that connects the “Find” (the vulnerability detection) process to the “Fix” process for the systematic, efficient elimination of vulnerabilities from web applications. Additional practical PCI guidance can be found at [16].

What does PCI 6.6 mean

The ultimate goal is to ensure secure web applications. For applications developed or customized in-house, the following process must be continually performed: Identify vulnerabilities (find), correct them (fix), and test to confirm that the correction is effective (prove). Find, fix, prove, find, fix, prove.

PCI quarterly network scans – too little too late

The quarterly network scans will find some SQL injection and catch missing basic input validation, but generally they cannot find application-level vulnerabilities. Web applications need to be checked on a continuous basis, these quarterly network scans should not be relied on to tell you if your Web apps are vulnerable.

Vulnerabilities must be detected, communicated, and corrected

A long term goal of Requirement 6.6 is the establishment of a web application vulnerability life-cycle – leading to the effective elimination of risk. Vulnerabilities must be detected, communicated, and corrected. This can be done through various measures including, black box testing (run-time assessment), white box testing (source code review), binary analysis, static analysis, remediation by developers or web application firewalls.

Runtime assessments, source code reviews, binary and static analysis to find vulnerabilities in web applications

There is a misconception that all detection techniques try to achieve the same end goal and compete for the same budgetary dollars. The fact of the matter is that each testing ideology brings different benefits to the table at different prices, almost all of which are complementary and help paint a complete picture of application weaknesses. While Vulnerability Scanners are required for PCI DSS section 11.3 and can be used for section 6.6, WAF helps organizations meet 8 of the 12 PCI DSS requirements. That's eight PCI DSS requirements that WAF helps meet versus just two that vulnerability scanners can help meet.

Requirement 6.6 option 1 – application code reviews

The application code review option does not necessarily require a manual review of source code. Keeping in mind that the objective of Requirement 6.6 is to prevent exploitation of common vulnerabilities (such as those listed in Requirement 6.5), several possible solutions may be considered. They are dynamic and pro-active, requiring the specific initiation of a manual or automated process. Properly implemented, one or more of these four alternatives could meet the intent of Option 1 and provide the minimum level of protection against common web application threats:

1. Manual review of application source code
2. Proper use of automated application source code analyzer (scanning) tools
3. Manual web application security vulnerability assessment
4. Proper use of automated web application security vulnerability assessment (scanning) tools.

Requirement 6.6 option 2 – application firewalls

PCI Requirement 6.6 can be quickly met through installing a web application firewall. In the context of requirement 6.6, an "application firewall" is a web application firewall (WAF), which is a security policy enforcement point positioned between a web application and the client end point. This functionality can be implemented in software or hardware, running in an appliance device, or in a typical server running a common operating system.

PCI Auditors seek evidence of due care

The PCI Council has not asserted itself as an authority on application security. It leaves the verification of compliance to approved auditors.

What the PCI Auditors seek is evidence of due care. Automated tools alone only cover roughly half of the web Application Security Consortium's Threat Classifications. If an application is worth protecting, test it thoroughly with both automated and human means. Web applications are continually changing, as is the threat landscape. Test the application in production as frequently as is meaningful, for example, with each code change. Vulnerabilities identified become a known liability and must be managed. Vulnerabilities must be communicated clearly and effectively to groups tasked with remediation.

Testing custom application code must be done methodically, and retesting must follow the same processes where possible. Patch development, validation of remediation, and corrections will be simplified if you follow a consistent methodology.

Application Layer Attacks

Web Application Attacks

Buffer overflows, SQL injection and Cross Site Scripting

Buffer overflows and SQL injection are not new, but attackers still manage to make effective use of them to gain access and administrative privileges to databases. Intrusion prevention systems are of use in dealing with buffer overflows. SQL injection is a popular method of attack, since modern databases utilize SQL - Structured Query Language - to enable users to access and manipulate data stored in a database. The basic procedure for a SQL injection exploit is to provide a valid request at the beginning followed by a single quote and a ";" with an additional request appended which contains the actual command the attacker hopes to implement. By piggybacking the "bad" code onto the good code it is possible to trick an incorrectly configured database into carrying out unauthorized executions.

Cross site scripting occurs when a web application gathers malicious data from a user. The data is usually gathered in the form of a hyperlink which contains malicious content. The user will most likely click on this link from another website, instant message, or simply reading a web board or email message. Usually the attacker will encode the malicious portion of the link to the site in HEX (or other encoding methods) so the request is less suspicious looking to the user when clicked. After the data is collected by the web application, it creates an output page for the user containing the malicious data that was originally, in a manner to make it appear as valid content from the website.

THERE ARE MANY VARIABLES IN APPLICATION SECURITY TESTING, SO YOUR MILEAGE WILL VARY

Finding vulnerabilities in web-facing applications

Regardless of your classification as a Merchant or Service Provider, if you have a web-facing application, it must be assessed. This will be far more exhaustive than a network vulnerability scan, and will require authentication to access the majority of application functionality. This testing requires human expertise to exercise the application, validate findings, and test for logical vulnerabilities and other threats a testing tool cannot identify.

Vulnerabilities in custom application code

Vulnerabilities in custom application code can be found in a variety of ways. The Web Application Security Consortium [12] has classified 24 different types of attacks targeting web applications. Half of those threats (13 technical vulnerability classes) can be identified at some level of effectiveness through automated means, including run time code testing as well as source code analysis. As with any detection technology, there is a certain signal-to-noise ratio. Human validation is required to separate true vulnerabilities from false findings. There are many variables in application security testing, so your mileage will vary. There are 24 threat classifications, with two current appendices (HTTP Response Splitting and Cross Site Request Forgery), which have not yet been formally ratified into the WASC Threat Classification document.

Fixing vulnerabilities

PCI Requirements 11.3.2 and 6.6 require this. For context, reread PCI requirement 6.1. Proving you have installed a patch to commercial applications and operating systems is easy. Proving you have corrected a weakness in custom application code is a little more complicated

This is where having a consistent testing and reporting methodology will come in handy. If you own the web application code - fix it. If you do not own the code, or have valid business case or cost restrictions that are impediments to fixing the raw code - correct the vulnerability through other methods (e.g., a web application firewall).

Ensure the mitigation correct the vulnerability in practice and in writing

After significant investment in managing the web application vulnerability lifecycle, an auditor (SOX, PCI, or any other auditor) needs documentation to prove the fix worked. Ensure the mitigation applied does in fact correct the vulnerability in practice and in writing. The PCI 6.6 compliance process of “Find, Fix, Prove” can be simplified further. If the “Find” process is done with sufficient precision and creates proper documentation, the “Find” process can be done in a continual or ongoing manner - and will in turn document proof of the “Fix” actions as they occur. Auditors like to see trends, especially when they involve continual detection and removal of vulnerabilities - this makes proving due care very easy.

With a clear understanding of PCI Requirement 6.6, compliance is not only achievable, but can provide great value to web application owners and users. This requirement creates a need for visibility into the life-cycle for vulnerability detection and correction, and will serve to mature web application security. Applying metrics to the efficiency of detection, the cost of producing vulnerable code, and the associated costs of correction will only serve to advance the goal of total web application security.

Traditional network firewalls

A WAF is different from other firewall approaches. Traditional firewalls which perform packet filtering only cannot monitor and block by user, which is required for compliance. Also, without a white list security model, this type of solution cannot protect against parameter tampering, session hijacking and cookie poisoning attacks, among others. The bottom line is that network firewalls do not understand enough information about the application and it’s state over time to provide adequate application security functionality.

1st generation Web application firewalls

Reverse proxy only Web application firewalls introduce latency, because they terminate traffic and require changes to the network, DNS and the application itself, as discussed in [33]. They may even break applications in the event of large traffic loads.

Application delivery solutions with application security add-ons

Layer 7 content switches and first generation Web app firewalls share something in common: generally they both mandate deploying reverse proxies to modify and manage traffic.

As a consequence, many application delivery vendors acquired Web app security technology and integrated it into their content switches. However, these joint solutions have retained all of the challenges of legacy Web app firewalls. For example, they often rely on manually defined white lists to validate Web requests. They protect session IDs by signing cookies and obfuscating URLs—intrusive measures that often have unexpected consequences. Combining Web application security and delivery also introduced many new challenges. The extensive regular expressions and content parsing in Web security significantly degrades the performance of application delivery products, upwards to 50%. And lastly, most application delivery vendors do not specialize in Web security, so they do not regularly research new application threats or automatically update security policies.

Selecting a defense strategy

Holistic security - protecting the enterprise data flow

Management accountability

Different options to protect payment card data is discussed in [18]. Protecting the enterprise data flow is discussed in [15] and [17] is looking at security beyond PCI. Scanners identify vulnerabilities. If those vulnerabilities are not fixed, but still known, management is accountable. We know that it often takes months to fix vulnerabilities in the application.

WAF provides a unique solution: it prevents the vulnerability from being exploited, allowing time to fix the code – thus eliminating the accountability issue. The easiest way to protect against these sorts of exploits.

Data at rest encryption - configuration files, log files, web pages and data

Web Server Applications are typically accessing configuration files, log files, web pages and data. It is recommended to secure these files by encrypting each type of files using different encryption keys. With a mature encryption solution, all files are encrypted yet access can be granted to any of these types based on the user's role.

Complementing application scanning tools

Vulnerability scanning tools cannot verify cryptographic storage at all. Code scanning tools can detect use of known cryptographic APIs, but cannot detect if it is being used properly or if the encryption is performed in an external component. Like scanning, testing cannot verify cryptographic storage. Also do not use weak algorithms, such as MD5/SHA1. Favor safer alternatives, such as SHA-256 or better. Code review is the best way to verify that an application encrypts sensitive data and has properly implemented the mechanism and key management. Please see OWASP 2007 item is A8 – INSECURE CRYPTOGRAPHIC STORAGE [4] for more information. Database scanning, i e scanning for database vulnerabilities like default/weak passwords, improper configuration, patch level, etc. since application security is very dependent on database authentication (based on the transparency requirement), for instance a weak password would be critical. This is related to PCI 2.1, 2.2, 6.1, 6.2 and 8.5.

VULNERABILITY SCANNING TOOLS CANNOT VERIFY CRYPTOGRAPHIC STORAGE AT ALL

Protecting the Data Flow

Limit the exposure of sensitive data bytes inside applications and the LAN

Many applications have no need to view all bytes in every data field that is passed through. One approach to protect this information in application memory and in transit is to use masking or partially encrypt sensitive fields to hide the not needed bytes from exposure [7]. This can be enabled by using some mode of AES encryption algorithm that is providing full or partial format preserving encryption or preservation of length or data type. This allows arbitrary fields to be encrypted to a given target format. This alleviates the need to change the database, and minimizes the application end point changes to a minimum. Some of these types of AES encryption modes may not be approved for use when permanently storing PCI data.

Validate the encryption mode with a certified PCI assessor

It is always a good practice to check if the AES encryption mode is approved for the type of use that you are planning. You may check with a certified PCI assessor. Please see [6] and [13] regarding merchants, risk management and other considerations. I'd also check the list that is approved by NIST [5]. In Special Publication 800-38A, five confidentiality modes are specified for use with any approved block cipher, such as the AES algorithm. The modes in SP 800-38A are updated versions of the ECB, CBC, CFB, and OFB modes that are specified in FIPS Pub. 81; in addition, SP 800-38A specifies the CTR mode.

Finding Vulnerabilities

Source-level analysis is clearly still required

An application firewall is an excellent solution for protecting against knowable front-end attacks, and can be the only solution for applications where deeper analysis is not possible or permitted. Having said that, source-level analysis is clearly still required, because a majority of customer credit information exposures occur because of issues with access control, authorization, and data storage/transmission.

These problems are, and will continue to be, outside the capability of a firewalling technology to distinguish.

Vulnerability scanners and application code reviews

Manual code reviews have negative aspects like time, cost, skills required etc. but may detect bugs that are hard to find in other ways. Tool-based code reviews may detect a different set of bugs. There are a few different code review alternatives discussed above.

Performance and frequency

Scanning web sites in production can disrupt website performance. Applications, especially Web applications, change frequently, so the target of vulnerability scanning and code review is a moving target, and new vulnerabilities can be introduced at any time. In many cases the application can change before a review cycle has been completed.

Attacks change frequently

Attacks, especially Web attacks, also change frequently. A few years ago, no vulnerability scan or code review would have found response splitting problematic. Then a paper describing response splitting attack techniques required developers to send the same code back to review.

Source code may not be available

For many applications the source code is not readily available or understood – and, in some cases, cannot easily be changed by the organization using the Web application. This could be either because the application is a third-party application or because the original developers of a legacy application are no longer available to explain what they did.

Code reviews are often not security oriented

One of the problems with manual code reviews; they are more often done for functionality purposes. It is expensive and time-consuming process to go through manual code reviews like the OWASP-based source-code review.

On the code review side, just about all forms of testing options are still on the table. Black and white box, with or without automated scanning assistance, and that kind of flexibility is a good thing. The catch is the person/firm doing the testing “must have the proper skills and experience to understand the source code and/or web application, know how to evaluate each for vulnerabilities, and understand the findings.” This goes for tool use as well. That’s going to be the little bit fuzzy part since our industry is new and doesn’t really have formalized certification or education processes. It’ll be up to the merchant to prove the case to their auditor or bank.

A massive legacy code base

We not only develop code at a staggering pace, we have a massive legacy code base. While many leading organizations follow secure software development life-cycles, and many more will be adopting at least some level of code scanning over the next few years thanks to PCI 6.6, it’s naive to think that even the majority of software will go through secure development any time soon.

On top of that, we are constantly discovering new vulnerability classes that affect every bit of code written in the past. And, truth be told, no tool will ever catch everything, and even well-educated people still make mistakes.

Senior skills needed

Manual code reviews and manual assessments of scan results are only as good as the reviewer. Skill sets vary widely and can be very expensive. Manual code fixes are only as good as the developer. Often, manual code fixing introduces new vulnerabilities.

Penetration tests

Application vulnerabilities can be a significant class of vulnerabilities and no scanner or WAF can identify. Application vulnerabilities can be introduced by bad design or programming.

The best way to find those vulnerabilities is by a penetration test. Penetration tests should be performed by a security expert and can be better than code review in finding problems from the overall system view.

Comparing WAF with scanners and code review

Web applications and WAF

Companies need to do security code reviews, specifically designed for web applications coding errors and web application vulnerabilities. Then the feedback from the review process – which requires automated tools to integrate into the Web application development design templates and scripts and tools.

Web applications are a special breed of living code - always online, always accessible, always being modified, and always subject to attack. Diligent web application security demands frequent assessment/attack research and findings targeting specific web applications are posted daily.

WAF - immediate protection and without changing the application

A WAF can be deployed to provide immediate protection and without changing the application. Vulnerability scanners and application code review are both still require developers to manually fix code – this takes time and isn't always possible. WAF's Dynamic Profiling technology automatically profiles applications and user behavior, automatically provides accurate protection for web applications and cardholder data, and automatically adjusts as applications and user behavior change to provide continuous protection of web applications and cardholder data, and can be used to provide valuable information to developers to improve the application under normal cycles.

A WAF is a complement to the development processes

A WAF is useful, and complementary to building security into the development processes. The WAF is probably the best choice in many situations. The WAF is the best first step as it can provide an immediate solution for immediate threats.

It can also provide new immediate solutions as other methods uncover issues over time, or as new attack methods evolve over time. Even if the customer is aware of web-threats and develops his web-application with security in mind, the customer is aware of the PRESENT threats, not about FUTURE threats. Today's secured applications will not necessarily stay secured tomorrow. There is a great opportunity for a feedback loop in both directions from WAF to code review and/or pen testing and/or scanning solutions.

WEB APPLICATIONS ARE A SPECIAL BREED OF LIVING CODE - ALWAYS ONLINE, ALWAYS ACCESSIBLE, ALWAYS BEING MODIFIED, AND ALWAYS SUBJECT TO ATTACK

Deploy a WAF and build a plan for a long term code review

A WAF can help to balance different options. One issue is that PCI puts two very different techniques against each other. Organizations are going to choose only one technique to achieve compliance, where, in reality, they should be using both.

Looking past that, the wording works very well for web application firewalls, in the sense that most organizations are likely to choose to deploy a WAF rather than go through a very long and very expensive process of code review.

WAF is an aid to Web application developers

WAF provides critical information on usage patterns and changes in usage patterns that can GUIDE code review teams and point out problems so they can fix any underlying logical issues in the application.

After WAF is deployed, code review and code fixing can proceed at a controlled pace

WAF secures web applications and cardholder data without incurring the time and cost to bring 3rd party consultants or maintaining a separate dedicated group to review code.

After WAF is deployed, code review and code fixing projects can proceed at a controlled pace, reducing risk of errors and reducing the extra costs of emergency-mode development.

The basic premise is that we need to assume that any browser that connects to our applications is completely compromised. Attacks like cross-site request forgery are just too difficult for the average browser/application to defend against. A big part of the problem is that the browser is a multi-session tool. Unlike most of our client/server desktop applications, the browser needs the ability to mix content from multiple sources, often in a single window. It's how the whole darn Internet works.

Some organizations don't understand what an application firewall does or how does, to use it, and may use a network scanner as a substitute for an app firewall.

Selecting a WAF solution

WAF selection criteria

The clarification provides more depth on what is required of a solution in order to meet Option 2 for Section 6.6. Several vendors views this clarification as a positive step for the industry as there have been frequent misleading claims by solutions attempting to claim application security functionality where none in fact exists. The new guidance provides a step in the right direction in defining the specific functionality that Web application security comprises.

An important part of the guidance stresses the need for a solution to provide specific application security functionality, saying:
"Increasingly, WAF technology is integrated into solutions that include other functions such as packet filtering, proxying, SSL termination, load balancing, object caching, etc. These devices are variously marketed as "firewalls," "application gateways," "application delivery system," "secure proxy," or some other description. It is important to fully understand the data-inspection capabilities of such a product to determine whether the product could satisfy the intent of Requirement 6.6."

Only a WAF in blocking mode to satisfy PCI 6.6 requirements

Be aware that simply buying expensive WAF hardware does not meet this requirement. Configuring that application-layer firewall to fix known vulnerabilities is required, and entails the risk of mis-configuration, and potentially blocking legitimate traffic to your website -- but you must configure the WAF in blocking mode to satisfy PCI 6.6 requirements that the vulnerability has been corrected.

PCI require a sophisticated Blocking WAF

And the list they provided is quite detailed and extensive requiring a sophisticated product, no marginal network security device with a few webappsec checks is going to cut it here. Of course the catch here is the device must be configured to "block" the attacks, not just alert on them. That's going to be the most challenging part in my estimation as this is not a trivial process. An issue that's not been brought to the front is what happens from a PCI perspective if an organization chooses code review (or if the clarification allows for pen test / scanning in the future) and that review turns up an issue requiring a long fix cycle.

WAF critical requirements

A "sophisticated WAF" should search for REQUEST vulnerabilities and should look for REPLY vulnerabilities (look for forbidden patterns). These capabilities are very different from IDS/IPS and network sniffers.

Soft appliance, a hardware appliance or any combination WAF should be able to be deployed as software, a soft appliance, a hardware appliance or any combination of the three. This will enable the WAF to be a completely "green" solution, coupled with deployment flexibility, make it an ideal choice for shared hosting and virtual server environments. A WAF should also be able to operate as an in-line gateway or out-of-band monitor.

Latency issues with traditional application firewalls

Most application firewalls, whether they are implemented as separate reverse-proxy server machines, co-located with the application on the same host machine, or co-located with network firewall machines, generally operate in real-time, intrusively in-line with the applications they protect. This introduces latency while the application firewall examines the traffic, logs the activity, alerts IT Operations and/or network firewalls to suspected attacks and passes traffic on to the application. Additional latency is introduced when HTTPS traffic is examined.

For instance, secure socket layer ("SSL") protocols used in HTTPS are terminated and decrypted prior to examination; in some implementations, traffic is additionally encrypted again before passing traffic on to the Web, application, and/or database servers for final HTTPS termination. Application firewalls are not configured to take advantage of security events or behavioral anomalies detected elsewhere in the environment in the same approximate timeframe, although correlation with those events is a typical practice when auditing the forensics of events via log files, long after the events have occurred.

Web application firewalls combined with an escalation system

Automated, synchronized threat monitoring and response between the application level and database level provides a highly effective protection against both external and internal attacks. An escalation system [14] can solve most of the latency issues with traditional application firewalls by dynamically switch Web application firewalls between different protection modes is described below.

I DON'T THINK STAND-ALONE EXTERNAL WAFS WILL EVER BE EFFECTIVE ENOUGH TO PROVIDE US THE SECURITY WE NEED FOR WEB APPLICATIONS

The Future of WAF Technology

Neither approach can solve the web application security problem

It's increasingly clear that no matter how good we are at secure programming and no matter how effective our code scanning and vulnerability analysis tools are, neither approach can “solve” our web application security problem.

Need to change how we view WAFs

I don't think stand-alone external WAFs will ever be effective enough to provide us the security we need for web applications. Rather, we need to change how we view WAFs. They can no longer be merely external boxes protecting against generic vulnerabilities; they need tighter integration into our applications.

Web application firewalls, applications, databases and file systems combined with an escalation system

Think of it as a combination of a web application firewall, an agent on the application server watching activity (what a user clicks on, where data goes) and a database agent or passive monitor watching all SQL activity, see [19] and [2].

A Multi-layer security advisory framework

A Multi-layer Security Advisory System provides a framework to effectively deal with threats of some classes of attacks. The warning system has 5 risk-of-attack-levels (Threat Levels) which when triggered, initiate specific actions by local servers within the same policy domain. Information about data security events is collected from sensors at different system layers (web, application, database and file system). The Threat Level is propagated to systems that are connected within a data flow. The Threat Level will also adjust for time of day, day of week, and other factors that are relevant.

A score-card to keep track of usage abnormalities

A score-card is maintained for each subject (user or service account/proxy-user, IP address, application, process) and object (database column, file) with a history of processing sensitive data. The score-card summarizes current and historical information about data access patterns for each entity (subjects and users). The score-card also includes a ‘fingerprint’ that reflects historical deviation from acceptable access patterns at the level of s/i/u/d (select/insert/update/delete) operations.

A high score-card value will initiate more extensive analysis before releasing data to the subject. The dynamic and automatic altering of the protection policy between multiple system layers includes modifying the protection policy of data at one or several of the system layers.

The modification is performed based on a result of the prevention analysis. The score-card can also keep track of when a remote system need to reconnect to the central system to renew or recharge it's capability to encrypt and decrypt data. The policy may allow the local system to only operate stand alone for a certain time or processing a fixed number of crypto operations between each host connection and central password renewal. This behavior will act like a rechargeable key box and can automatically shut down the local access to sensitive data in case the local system is stolen, cloned or compromised in some other way, see [3].

We link in to track activity through the application stack and can alert on things like a user seeing credit card numbers they've never had access to before, or activity that resembles XSS. So it's some of what you talked about, but really looking more at an end-to-end user transaction and seeing if that violates policy or not.

Multi-layer system for privacy enforcement and monitoring of suspicious data access behavior A method for controlling data access in a data-at-rest system includes executing a link intrusion prevention analysis between multiple layers of the data-at-rest system, introducing a privacy policy at enforcement points that span multiple system layers, and dynamically altering the privacy policy.

Selecting a code review approach

Web development and code management

Web development with .NET, C#, Java, PHP, JavaScript, AJAX, is covered in [8] and [9] and OWASP [10]. Code reviews of managed code (.NET environment) from Microsoft is covered in "How To: Perform a Security Code Review for Managed Code (Baseline Activity)" at [11]. One of the code scanning tools mentioned below for general application development using Java, C/C++ and other languages.

Security Development Lifecycles

Microsoft's Trustworthy Computing Security

Related to this is Microsoft's Trustworthy Computing Security Development Lifecycle (SDL) initiative. SDL describes requirements for different phases in development, with the main goal to reduce the number of vulnerabilities in software products. For the Implementation Phase it's said in [23] that:

Apply coding and testing standards

Coding standards help developers avoid introducing flaws that can lead to security vulnerabilities. Testing standards and best practices help to ensure that testing focuses on detecting potential security vulnerabilities rather than concentrating only on correct operation of software functions and features. Fuzzing supplies structured but invalid inputs to software application programming interfaces (APIs) and network interfaces so as to maximize the likelihood of detecting errors that may lead to software vulnerabilities.

Apply static-analysis code scanning tools and code reviews

Tools can detect some kinds of coding flaws that result in vulnerabilities, including buffer overruns, integer overruns, and uninitialized variables. Microsoft has made a major investment in the development of such tools (the two that have been in longest use are known as PREfix and PREfast) and continually enhances those tools as new kinds of coding flaws and software vulnerabilities are discovered.

Code reviews supplement automated tools and tests by applying the efforts of trained developers to examine source code and detect and remove potential security vulnerabilities. They are a crucial step in the process of removing security vulnerabilities from software during the development process.

Separate code reviews as a way to enhance security

Both PCI DSS and SDL mention separate code reviews as a way to enhance security.

In addition SDL mentions the use of static-analysis code scanning tools. Such tools often assists during code reviews, but may also be applied during normal development.

Static/dynamic analysis and other definitions

There are two principal types of program analysis. Static, or compile-time, analysis is aimed to investigate a program's run-time properties without actually executing it. Normally this is performed by source code inspection, but binaries may also be used. Dynamic, or run-time, analysis is performed when observing the program at execution. Testing, debugging and performance monitoring are examples of dynamic analysis.

Example of a very simple static analysis

An example of a very simple static analysis would be searching for specific words like strcpy using a file-search utility like grep. The goal would be to identify where unsafe functions are used.

A search for strcpy will however also list values like strcpy_s. If strcpy is part of a comment, this will also be presented as a valid occurrence. Such output is called a false positive, i e something reported as a vulnerability though not. False positives are a big issue in static analysis since they give the user more data to evaluate than necessary. Each program construction reported as a vulnerability must be considered and reviewed.

Suppose strcpy is renamed to something else, for instance with a macro like '#define mycopy strcpy'.

In this case a word search for strcpy won't list any occurrence of mycopy, though strcpy really is used. This is called a false negative, i e a real vulnerability that hasn't been presented.

An ideal analysis tool

An ideal analysis tool would have no false negatives and no false positives, only true vulnerabilities would be presented. That is however not realistic. Instead they are often somewhat related. A lower false positive rate means a higher false negative rate, and vice versa.

Free open source static-analysis tools

There are different free open source static-analysis tools, as described further below. These are only marginally better than the simple word-search as above. They search for specific unsafe calls like strcpy as listed in an internal database, and when found they present the position and a general comment about the problem. This handling gives a lot of false positives, but also a lot of false negatives since they only look for some calls. Such simple tools are of limited value.

More advanced tools

More advanced tools try to interpret the context of the word, based on the programming language used. This is called semantic analysis. The better this analysis is, the fewer false positives there will be. The free open source tools do perform some semantic analysis, meaning at least some false positives will be skipped.

In addition to look for certain language-specific words, advanced tools also look at the general program context. An intra-procedural analysis only looks at what happens within a specific function. This may be inaccurate, for instance when external entities like globals are used. An inter-procedural analysis tries to consider all parameters of the function, and the interaction of functions. This is much more complicated than intra-procedural analysis, considering different possible parameter values and paths for execution. Related to this is flow-sensitive and path-sensitive analysis, which try to consider the program flow and the different paths possible.

Inter-procedural analysis may in some cases not be impossible

Even if supported by the tool, inter-procedural analysis may in some cases not be possible. If there are third-party libraries for which source code isn't available, or there are yet unimplemented functions, the tools can't inspect what happens inside these calls. This may result in false negatives produced.

Tools often try to simplify analysis, to get better performance

Tools often try to simplify analysis, to get better performance. Doing such as inter-procedural and flow-sensitive analysis could consume considerable resources. A tool may for instance only consider min/max values when handling integer input. Such simplifications are also a source of false negatives. In general, a tool will never be totally accurate, but the better it performs different types of advanced analysis, the more vulnerabilities it will find.

Using static-analysis tools during development

There is a wide range of static-analysis tools, from simple syntax checkers to advanced tools performing semantic, inter-procedural and flow-sensitive analysis. The advanced tools are also getting more advanced for each version, applying new types of analysis and vulnerability detection rules.

Examples of what tools may look at are:

- resource leaks
- references to NULL pointers
- use of uninitialized data
- buffer array overruns
- unsafe use of signed values
- use of resources that have been freed
- concurrency handling.

Without doubt a tool capable of such analysis would be valuable during development. A tool may however be used in different configurations.

TOOLS OFTEN TRY TO SIMPLIFY ANALYSIS, TO GET BETTER PERFORMANCE

The questions are when, where and by whom should the tool be applied? There are different options:

1) When the code is written by the developer

First option would be to run the tool when the code is being written. In this case it's the developer that runs the tool, in the local IDE used. Later versions of commercial tools also support a mixed handling where there are local instances, but still some central repository for handling overall metrics, for instance to see if coding skill is evolving over time.

There are both advantages and disadvantages with the local approach:

+ It's easier and faster to handle a bug if caught directly when the code is written; the programmers know their own code best, and the code is in current focus.
+ Handling a bug locally means it's not propagated to the central repository, thereby affecting other developers.
+ Running a tool and interpreting the output will educate the developers in secure coding. Tools have contextual help that explains a given vulnerability and how to handle it.
– Tools are often licensed per user, one tool instance per developer could mean a large total cost for the tool.
– Running a tool too early could mean an unnecessarily high number of false positives. Tools are less precise when the code is in an initial phase, and inter-procedural analysis doesn't really apply when many code pieces are missing (later versions of commercial tools however claim to be better in this aspect).
– Each developer must be trained in using the tool. Interpreting tool output is for senior developers, with appropriate security knowledge. Marking a valid bug as a false positive could mean the weakness is lost.
– Each developer workstation needs a local installation of additional software.

2) At build time, when scheduled builds are performed

A second option would be use a tool integrated in a central build process. The scan is performed when the total application is built. This is an option often used with commercial tool offerings.

+ A central configuration means a group of senior developers may evaluate tool output before it's distributed to responsible developer, the analysis could be transparent to developers.

+ Providing error reports to responsible developers means education in secure coding still applies.
+ Server installations minimizes the number of software deployments necessary.
+ Reports are easily generated concerning overall secure coding metrics, and may be accessed by everyone.
+ The tool is executed a bit later in the development process, not until the code has been checked into the build system. This will reduce false positives caused by premature code.

– The tool license cost may still be based on number of users, or it may be some general server license. The licensing cost could still be high.
– Bugs are not handled directly, but if the build is performed often the code is still current and not that hard to modify.
– Errors are propagated to the central repository, thereby affecting other developers until corrected.
– A specific group of reviewers may become a constrained resource. They will however likely become tool experts after some time, speeding up the process.

3) At certain code review checkpoints, by a security oriented code reviewer

A third option would be to use the tool as an assistant during code reviews, to be performed at certain project milestones like before product release.

+ The tool license cost should be smaller, since only a few security oriented users will have the tool. License could however also be based on code size and other aspects.
+ The tool is executed late in the development process, which will reduce false positives caused by premature code.
+ Senior security oriented developers are evaluating output before it's distributed to responsible developer, the analysis could be transparent to developers.
+ Distributing error reports to the developer in charge means education in secure coding will still apply. Errors have been filtered by the code reviewer.
+ Reports may be generated concerning overall secure coding metrics.
– Bugs are not handled directly, but rather late in the development process. Fixing an error will take longer time and be more costly, code may not even be current for developer when bug is presented.
– Errors are propagated to the central repository, thereby affecting other developers until corrected.
– The security reviewer may not know the code, which could slow down the interpretation of the tool output.
– The group of reviewers will likely become a constrained resource, possibly handling different projects. They will however become tool experts after some time, speeding up the process.

All these three cases could apply for ABC development

All these three cases could apply for ABC development. The first case seems like an attractive way to go; the errors are handled directly by the developer and won't affect others, the developers will become more skilled as time goes by. But based on the cost involved to pursue such a configuration, it's absolutely necessary to first verify how good the tool is in the ABC environment. A similar high cost would also be true for the second configuration.

A specific platform library, where system calls are hidden

ABC has a specific platform library, where system calls are hidden. There are different types of wrapper functions and classes used throughout the code. Vital parts like encryption and SSL are implemented in external libraries. All of this means inter-procedural analysis will be important for a ABC analysis tool. Until a tool is tested in the ABC environment, it's not possible to say how good it will be, and it shouldn't be used on a large scale.

Since there is much code already developed in ABC

Since there is much code already developed in ABC, and all this code is about to have a code review, the third option could be a good starter. This will give an indication of general code quality, and should minimize the initial cost for the tool.

If the tool used is found to be very helpful, and the tool budget allows, the tool may later be propagated into the whole developer community either as option one or two.

ABC is a highly security oriented organization

Another consideration is that ABC is a highly security oriented organization. A higher degree of false positives could be accepted for a ABC scan, since this normally also means a higher percentage of errors identified. But this also means there will be more time spent with the tool output, each error reported must be evaluated.

This is a big issue in a time-constrained environment, which is a reality for many developers. If using the first configuration, an option would be to restrict the vulnerability rule set for local development, and then have a more thorough rule set for a central build or security review.

Tool selection criteria

A static analysis tool would certainly be valuable during development, no matter where it's applied in the development chain. But all tools will of course not be equally good. There are different things to consider when selecting a static analysis tool, especially for a commercial tool. Some considerations will depend on how the tool is going to be used.

1) Multiple language support

The tool must support the programming languages used for development. This is a requirement no matter where the tool is applied. The main languages used for ABC development are C/C++ and Java; support for these is a basic tool requirement. But ABC also has some code built with for instance C#, PL/SQL and T-SQL. Support for these additional languages would be a further advantage, though not a requirement.

2) Capability in finding vulnerabilities

The principal task for the tool is to find code vulnerabilities. Strong capability in this area is another major requirement. This is a requirement no matter where the tool is applied. This ability is two-fold. There should be a high rate of true errors identified compared to the total number of errors, but there should also be a low rate of false positives. These are often a bit contradictory; a lower false positive rate often means a higher rate of missed errors.

Being a security oriented organization, a higher degree of false positives could be accepted for a ABC scan if this means a lower false negative rate. The target for a security oriented organization must be to have the smallest amount of bugs possible, even if this means time for analysis will be extended.

3) Integration into development environment

If the tool is to be used as part of normal development operations, it's important that the tool integrates smoothly into the development environment used, for instance Visual Studio. If necessary to run a separate tool, it will likely be less often used than if closely integrated, and additional time must be spent on handling two different output lists. If used in a central build environment, the tool must of course integrate into what's used there. Since ABC is a multi-platform product, there must be support for at least one UNIX version and Windows.

4) Management of true errors and false positives

Tools normally provide some explanation why an error has been reported. The more specific this explanation is, the easier and faster it will be to evaluate if the reported item really is an error, or if it's a false positive. Good explanations are also important for educational purposes.

When an error has been fixed it shouldn't be listed any more. It should also be easy to mark en error as a false positive when this has been decided, and this mark should be consistent (saved in-between invocations). Otherwise it will be necessary to mark it as a false positive for each execution, and much time will be spent on the same errors. Related to this is the possibility to have different types of reports generated, for instance providing trends in number of errors reported. This may be useful for education and management.

5) Management of rule set

Tools are often based on some rule set, where different errors are grouped. There will always be false positives produced. It's important that it's possible to tweak the rule set, to adjust the amount of errors and/or false positives reported. A recommendation is to start with a small subset of rules in the beginning, to not get overwhelmed by tool output, and then step-by-step extend the rule set used. In the end a security oriented organization like ABC must be using an extensive list of rules.

Related to this is the complexity to add internal rules, to extend the default rule set. This is for advances users, but may be necessary in some situations, like when using external libraries or to catch certain error types. Writing an extension could mean writing a separate C/C++ library, or using some internal tool language format.

6) Price

Assuming the tool budget isn't unlimited, price may be an important parameter for a tool. If using one copy of the tool per developer, cost may easily be very high since the tools are often licensed per user.

License cost may often be selected either as an annual fee, or a perpetual one with some maintenance cost.

USING A COMMERCIAL TOOL IS RECOMMENDED

Concerning tool selection, there are three paths to go from here:

- Use free tools only
- Select a commercial tool based on trials with all four possible vendors
- Select a single commercial tool, and either buy a single license or perform trial.

1) Use free tools only

Using a free tool for C/C++ like Flawfinder doesn't seem to be an option. Especially since ABC has a platform library, which is the only place for unsafe calls as for instance listed in Flawfinder. Free tools could possibly be used as education sources, learning the unsafe functions if not known already. The GPL license type must also be considered.

Using Microsoft's PREfast should be added to normal development process. All existing ABC C/C++ code should be scanned with PREfast, and before new code is being checked in, it should have been scanned with PREfast (since compilation time will be longer when using PREfast, it probably shouldn't always be used). Code scanning with PREfast will however be restricted to the Windows environment, some UNIX specific parts in ABC won't be handled.

Looking at Java, the FindBugs tool should be a good choice. It has a LGPL license, and is even used in the Fortify commercial tool. All existing ABC Java code should be scanned with FindBugs, and before new Java code is being checked in, it should have been scanned with FindBugs.

2) Select a commercial tool based on trials with all four possible vendors

Using a commercial tool is recommended. A commercial tool will be more advanced concerning inter-procedural analysis than PREfast, and is expected to find more vulnerabilities. The C/C++ code is likely where most bugs will be found in ABC, being less secure programming languages than for instance Java.

The choice of a commercial tool is however not that clear. Based on the public tests available, there doesn't seem to be any major differences concerning bug-finding capability. Different ranking is rather based on tool management.

A general recommendation is to test the tool in the own environment, and most vendors support trials. An environmental test is maybe even more important for ABC, with its platform library and different types of wrapper functions/classes. Strong ability in inter-procedural analysis is important.

Conclusion

The case study from company ABC recommended using both WAF and coding reviews. ABC is planning an Enterprise Data Protection approach and protect data across the Information Life Cycle. The primary issues are PCI compliance, and a concern about the escalating threat against financial data from organized crime and insider threats. Time is a critical factor in selecting solutions to prevent breaches.

WAF and data encryption will give ABC a quick and solid data security solution to start with. ABC is complementing this with a long term approach including code reviews and scanning. ABC positioned different approaches to prevent Data Theft (and the attack-paths to the data – different types of apps, databases) including WAF, data (protection (encryption, hashing, tokenizing), C++ code scans. Implementation of Secure Development may take time and ABC are looking at PCI 6.6 as a part of their short term year's budget and install appliances. The appliance is easier to implement and will cost less, based on research done by ABC.

WAF is the most effective mechanism to immediately address security issues since the security rule set can be adjusted to stop new attack types without the time required to change the application code. WAF can protect custom applications, 3rd party applications, and legacy applications – even in cases where the organization does not control the source code (as for SAP, Oracle, PeopleSoft web applications and portals) and where the people who understand the application are no longer accessible. The solution for ABC is based on the conclusion that every layer of defense is critical. A holistic and layered approach can provide the best level data security and the sooner sensitive data gets encrypted, the more secure the environment. Early data encryption will protect sensitive data at rest and while it’s moving between the applications and databases and between different applications and data stores. An effective code-scanning tool would definitely be useful in ABC development. Being a security oriented organization, it’s very important to minimize the number of bugs.

The use of code scanning tools is also mandated by Microsoft’s SDL. No matter what tool used, this should be accompanied with code reviews, appropriate testing including such as fuzzy testing, code standards that are followed, and proper education. No matter what tool configuration selected, manual code reviews, education, coding standards and proper testing must also be applied.

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.

References and suggested reading

[1] Information Supplement: Requirement 6.6 Code Reviews and Application Firewalls Clarified, Feb 2008, https://www.pcisecuritystandards.org/pdfs/infosupp_6_6_applicationfirewalls_codereviews.pdf

[2] The Future Of Application And Database Security, December 2007, http://securosis.com/

[3] Multi-layer system for privacy enforcement and monitoring of suspicious data access behavior, February, 2006, United States Patent Application 20060259950

[4] OWASP 2007 item is A8 – INSECURE CRYPTOGRAPHIC STORAGE, PROTECTION section and in the VERIFYING SECURITY section, http://www.owasp.org

[5] NIST (http://csrc.nist.gov/CryptoToolkit/modes/). In Special Publication 800-38A

[6] http://usa.visa.com/merchants/risk_management/cisp_merchants.html

[7] Data type preserving encryption, November 2000, United States Patent 7,418,098

[8] NIST – Application Vulnerability Scanners

https://samate.nist.gov/index.php/Web_Application_Vulnerability_Scanners
[9]OWASP Tools
http://www.owasp.org/index.php/Phoenix/Tools
[10] The code review project at OWASP
http://www.owasp.org/index.php/OWASP_Code_Review_Project
[11] How To: Perform a Security Code Review for Managed Code (Baseline Activity),
http://msdn.microsoft.com/en-us/library/ms998364.aspx
[12] Web Application Security Consortium, http://www.webappsec.org/
[13] IT Security is a news and information publication,
bhttp://www.itsecurity.com/meet-experts/expert-biography-ulf-mattson-100206/
[14] Protegrity's Defiance Threat Management System, http://www.protegrity.com
[15] Protecting the enterprise data flow, http://www.ulfmattsson.com
[16] The PCI Knowledge Base is a Research Community,
http://www.knowpci.com/index.php?option=com_adsmanager&page=show_ad&adid=25&catid=1&Itemid=97
[17] Data Security for PCI and Beyond , http://papers.ssrn.com/sol3/papers.cfm?abstract_id=974957
[18] Payment Card Data - Know Your Defense Options,
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1126002
[19] A Multi Layered Approach to Prevent Data Leakage, Help Net Security (HNS), November 2007
http://www.net-security.org/article.php?id=1092
[20] Software that makes software better, Economist, Mar 2008
http://www.economist.com/research/articlesBySubject/PrinterFriendly.cfm?story_id=10789417
[21] Payment Card Industry (PCI) Data Security Standard v1.1, Sep 2006
https://www.pcisecuritystandards.org/
[22] Payment Card Industry (PCI) Data Security Standard v1.1 - Security Audit Procedures, September 2006
https://www.pcisecuritystandards.org/
[23] Trustworthy Computing Security Development Lifecycle, Microsoft, Mar 2005
http://msdn2.microsoft.com/en-us/library/ms995349.aspx
[24] Code Scanners, False Sense of Security?, Network Computing Report, Apr 2007
http://www.networkcomputing.com/channels/security/199000936
http://www.networkcomputing.com/channels/security/198900460
http://www.fortify.com/products/sca/
[25] Fortify SCA 5.0 Extends App Protection, Redmond Developer News, Nov 2007
http://reddevnews.com/news/devnews/article.aspx?editorialsid=855
[26] Fortify Software Extends Leadership in Detecting…, Fortify press release, May 2007
http://www.fortify.com/news-events/releases/2007/2007-05-14.jsp
[27]Source-Code Assessment Tools Kill Bugs Dead, Secure Enterprise, Dec 2005
http://www.klocwork.com/company/releases/12_05_05.asp
http://www.neohapsis.com/publications/articles.html
[28] Coverity and Klocwork code analyzers drill deeper, InfoWorld, Jan 2006
http://www.infoworld.com/article/06/01/26/73919_05FEcode_1.html
[29] DHS Funds Open-Source Security Project, eWeek, Jan 2006
http://www.eweek.com/c/a/Security/DHS-Funds-OpenSource-Security-Project/
[30] Scan Project Rung 1 status, Coverity, Mar 2008
http://scan.coverity.com/rung1.html
[31] Closing Security Holes with Application Scanners, Enterprise Systems, Jul 2007
http://esj.com/news/article.aspx?EditorialsID=2714
[32] Klocwork Inc, The Wall Street Transcript, Nov 2007
http://www.nohau.se/page.asp?page=artall
[33] Imperva, http://www.imperva.com/