IBM X-Force® 2010 Trend and Risk Report

March 2011

IBM

Dedication

The IBM X-Force® 2010 Trend and Risk Report is dedicated in memory of our friend and colleague Bryan Williams who passed away during this effort. His knowledge and focus on the changing threat landscape of virtualization is documented in this report. Bryan was a highly valued member of the IBM X-Force team since the early days and his contribution to the team, security and IBM are too numerous to list. He will be greatly missed.

Contributors

Producing the X-Force Trend and Risk Report is a dedication in collaboration across all of IBM. We would like to thank the following individuals for their rapt attention and dedication to the publication of this report.

Contributor	Title
Amrit Williams	Director, Emerging Security Technology
Bryan Williams	X-Force Research and Development, Protection Technologies
Carsten Hagemann	X-Force Software Engineer, Content Security
Colin Bell	Principle Consultant, AppScan OnDemand Services
David Merrill	STSM, IBM Chief Information Security Office, CISA
Dr. Jens Thamm	Database Management Content Security
Harold Moss	Emerging Tech & Cloud Computing Technical Architect
Jay Radcliffe	Senior Threat Analyst, MSS
Jeffrey Palatt	Manager, Emergency Response Services
John Kuhn	Senior Threat Analyst, MSS
Jon Larimer	X-Force Advanced Research, Malware
Leslie Horacek	X-Force Threat Response Manager
Lisa Washburn	Global Product Mgr, IBM Security Services—Threat/Cloud
Marc Noske	Database Administration, Content Security
Mark E. Wallis	Senior Information Developer for IBM Security Solutions
Matthew Ward	Senior Product Manager—Tivoli Security
Michelle Alvarez	Team Lead, MSS Intelligence Center(aka Eagle Eyes)
Mike Warfield	Senior Wizard, X-Force
Ory Segal	Security Products Architect, AppScan Product Manager
Patrick Vandenberg	Manager, Rational Security & Compliance Marketing
Ralf Iffert	Manager X-Force Content Security
Ryan McNulty	IBM Managed Security Services & SQL Querier Extraordinaire
Scott Moore	X-Force Software Developer and X-Force Database Team Lead
Shane Garrett	X-Force Advanced Research
Steven Bade	STSM Security Architect and Strategist
Tom Cross	Manager—X-Force Strategy and Threat Intelligence
Wangui McKelvey	X-Force Marketing Manager

About X-Force

The IBM X-Force® research and development teams study and monitor the latest threat trends including vulnerabilities, exploits and active attacks, viruses and other malware, spam, phishing, and malicious web content. In addition to advising customers and the general public about emerging and critical threats, X-Force also delivers security content to help protect IBM customers from these threats.

Contents
Section I

Contents
Section II, III and IV

Navigating the report

Welcome. This year we have made some helpful improvements to the format and content of the Trend Report. These improvements are aimed at enabling readers to take the findings a step further. We understand that computer and network security is about focusing on awareness of the threat and helping to protect the systems and networks from these threats. But then what? As an organization matures in its stance on computer security and known threats, how can they begin to develop a deeper focus towards improvement?

We asked ourselves that question and determined the answer was to provide to our readers a deeper understanding of what we experience and have learned from the breadth of capabilities that is IBM Security Solutions.

For this report we have divided the content into four sections.

- Threats
- Operating Secure Infrastructure
- Developing Secure Software
- Emerging Trends in Security

We start by talking about the threats that our systems and networks are facing, because we have to begin by understanding the problem we are all working to solve. Once a threat is understood, we can work towards realistic technology controls and educational awareness to help secure our enterprise and systems. In both the **Operating Secure Infrastructure** and **Developing Secure Software** sections we not only discuss threats but provide logical advice on how to help improve or detect those threats in your environment. In the **Emerging Trends in Security** section, we take a forward look into emerging technologies that are pressing into discussions as future business concerns.

We believe this new layout better organizes the material we want to present and helps you the reader focus on what is most important to your organization.

Section I–Threats

In this section we explore topics that comprise "Threats" and describe the attacks aimed at the enterprise that security specialists face. We address the malicious activity observed across the spectrum by IBM and how we go about helping protect networks from those threats. In addition, an update on the latest attack trends as identified by IBM.

Executive overview

The second decade of the twenty first century is underway and technology continues to permeate every aspect of our work and personal lives. At IBM we call this the Smarter Planet and we are continuously helping our customers to take advantage of a world that's more interconnected, intelligent, and instrumented. As much as these innovations can increase our efficiency and ability to instantly connect on a global scale, so too can the risks and dangers of a connected world become more sophisticated and difficult to contain.

To prove the point, the confluence of this innovation recently showed its face in several authoritarian countries, where technology and political activism have united to empower people in sharing a voice and making change on a global scale. More generally, we have seen a rise in hactivism across the globe, where attackers are no longer motivated simply by self-recognition or financial gain, but by political change and protest.

The second half of 2010 also marked a highly visible precedent in the industrial and manufacturing space. The multi-faceted and highly customized Stuxnet worm shook up the SCADA world by proving how security vulnerabilities can cripple a factory or production site. No longer is just e-commerce, personal, or corporate data at risk, but the very infrastructure that powers our factories and energy sector can be exposed for exploitation.

On a smaller scale, mobile devices continue to multiply in the workplace, helping increase the magnitude and complexity of risk in protecting the enterprise. In the emerging trends in security section, we look at several mobile vulnerabilities that may be an indicator of more to come. In the enterprise, and at home, web vulnerabilities targeting the browser continue to dominate the majority of weaknesses, demonstrating the importance of patch compliance and host protection. We discuss an interesting case study of how large complex organizations can benefit from centralized patch management.

In our **advanced persistent threat article,** we look at some of the most sophisticated adversaries our networks have ever faced. These types of low and slow coordinated attacks are often an indicator of highly cohesive and organized groups of attackers who use a variety of sophisticated attack techniques to inch their way into the enterprise.

Not only are attacks changing but so is the very technology that we utilize to carry this traffic. We take a quick look at how networks are scrambling to keep up with technology changes. At the mid-year point, we discussed a shift from IPv4 into IPv6 requirements and in this report, we discuss the oncoming advent of DNSSEC.

2010 was a pivotal year on many counts and has shown that understanding the trends of the security landscape is more critical than ever. IBM continues its dedicated effort to educate, inform, and discuss security topics and emerging trends with the community at large. Preparing organizations to not only understand the emerging threat landscape, but also to better understand the weaknesses of an organization's infrastructure.

2010 Highlights

Threats

Malware and the Malicious web

- IBM Managed Security Services (MSS) saw an upward trend in Trojan botnet activity during 2010. This growth is significant because despite increasing coordinated efforts to shut down botnet activity (as seen with the Mariposa, Bredolab and Waledec botnets), this threat appears to be gaining momentum.
- IBM's data illustrates the dramatic impact of a successful effort in early 2010 to shutdown the Waledac botnet, which resulted in an instantaneous drop off in observed command and control traffic.
- Zeus (also known as Zbot and Kneber), continues to evolve through intrinsic and plugin advances. The Zeus/Zbot family of botnets has been around for many years now and due to its extreme popularity with attackers, there are hundreds, or even thousands, of separate Zeus botnets active at any given time. The Zeus botnet malware is commonly used by attackers to steal banking information from infected computers.
- SQL injection is one of the leading attack vectors because of its simplicity to execute and its scalability to compromise large amounts of web servers across the Internet. There also appears to be a seasonal pattern: during each of the past three years, there has been a globally scaled SQL injection attack some time during the months of May through August.
- Obfuscation, whereby attackers attempt to hide their activities and disguise their programming, continued to increase over 2010 and shows no signs of waning.
- Compromise through PDF exploitation continues to be a favorite among attackers. In late April, a particular spam campaign contained an Adobe Acrobat PDF that used the Launch command to deliver malware. At the peak of the attacks, IBM Managed Security Services (MSS) received more than 85,000 alerts in a single day.
- The SQL Slammer worm first surfaced in January 2003 and became known as one of the most devastating Internet threats of the past decade. This worm continued to generate a great deal of traffic on the Internet in 2010.

Web content, spam, and phishing

- IBM Content security team identified that in the past three years, anonymous proxies have steadily increased, more than quintupling in number. Anonymous proxies are a critical type of website to track, because they allow people to hide potentially malicious intent.
- USA, India, Brazil, Vietnam, and Russia are the top five countries for spam origination in 2010.
- In 2010, spammers focused on content over volume. At the beginning of August, spammers began sending spam threats with ZIP attachments that contained a single EXE file that was malicious. By September, spammers began shifting to HTML spam to once again trick the end-user.
- There were a few months with ups and downs in the volume of spam seen over the year, however, the overall trends stayed flat and we have seen even less volume at the end of the year in comparison to the beginning of 2010.
- At 15.5 percent, India was the top country for phishing email origination in 2010, followed by Russia at 10.4 percent.
- In 2010, financial institutions continue to climb as the number one target for phishing attempts, representing 50 percent of the targeted industries up from the mid-year report when it was 49 percent.
- In 2010, more than three out of four financial phishing emails targeted banks located in North America. The remaining 22 percent targeted Europe.

Operating Secure Infrastructure

Vulnerabilities and Exploitation

- According to the X-Force database tracking, 2010 had the largest number of vulnerability disclosures in history—**8,562**. This is a 27 percent increase over 2009, and this increase has had a significant operational impact for anyone managing large IT infrastructures. More vulnerability disclosures can mean more time patching and remediating vulnerable systems.

- 49 percent of the vulnerabilities disclosed in 2010 were web application vulnerabilities. The majority of these were cross site scripting and SQL injection issues. However, as IBM X-Force has been saying for years, these vulnerabilities represent just the tip of the iceberg since many organizations develop third-party applications in-house that are never even reported publically and are not included in this count.
- Although vendors have been diligent in providing patches, at least 44 percent of all vulnerabilities in 2010 still had no corresponding patch by the end of the year.
- In early 2010, the term Advanced Persistent Threat (APT) became part of the everyday information security lexicon as a result of certain public disclosures and acknowledgement of a targeted series of attacks known as Operation Aurora. There has been much debate over this term and the underlying concepts within the information security community.
- During certain public disclosures in early 2010, and after attacks associated with Operation Aurora, the term APT began to take on a different meaning. In essence, APT became associated with any targeted, sophisticated, or complex attack regardless of the attacker, motive, origin, or method of operation.

Virtualization

- IBM X-Force notes that virtualization systems added 373 new vulnerabilities to the network infrastructure in the period between 1999 and 2009.
- A number of public exploits exist that demonstrate the risk from virtualization system vulnerabilities is real.
- Hypervisor escape vulnerabilities are the most common type of vulnerability that has been disclosed in server class virtualization systems.

Developing Secure Software

Web Application Vulnerabilities

- From the IBM® Rational® AppScan® OnDemand Premium Service we observed web application vulnerabilities comprising 49 percent of the total vulnerabilities reported in 2010, it is no surprise that developing secure software is harder than ever.
- In 2010 for the first time we now find that Cross-Site Request Forgery (CSRF) is more likely to be found in our testing than Cross-Site Scripting (XSS). This change is attributed to better detection techniques for CSRF and also a greater awareness of the risk. We find that organizations will tolerate having some outstanding issues with CSRF if the risk of exploitation is minimized. This is not the case with XSS and these issues are often quickly resolved.
- ASP.NET applications were clearly more susceptible to SQL injection than Java or PHP. The likely reason is that applications would typically use SQL Server as a backend database. SQL injection is better documented and easier to detect in this technology.
- As Web 2.0, AJAX applications, and Rich Internet Applications (RIAs) become more common, client-side JavaScript vulnerabilities may become more relevant, with a potential rise in the amount of such issues being exploited by malicious attackers.
- A recent IBM research study discovered that about 14 percent of the Fortune 500 sites suffer from many severe client-side JavaScript issues, which could allow malicious attackers to perform attacks such as
  - Infecting users of these sites with malware and viruses.
  - Hijacking users' web sessions and performing actions on their behalf.
  - Performing phishing attacks on users of these sites.
  - Spoofing web contents.
- Based on the dataset that we analyzed, we may extrapolate that the likelihood that a random page on the Internet contains a client-side JavaScript vulnerability is approximately one in 55.

Emerging Trends in Security

Mobile

- Mobile devices represent opportunities for sophisticated, targeted attackers. There are a number of vulnerabilities to target, and there is exploit information available.
- However, it is important to keep the vulnerability increases in perspective -- these do represent shared software components used by both mobile and desktop software. The vulnerability research that is driving these disclosures is not necessarily mobile-centric.
- Still, we aren't seeing widespread attack activity targeting mobile vulnerabilities today, because mobile devices do not represent the same kind of financial opportunity that desktop machines do for the sort of individuals who appear to create large Internet botnets.

Cloud security

- While security is still considered one of the major inhibitors to cloud adoption, organizations are increasingly adopting cloud-based technologies to address competitive market needs.
- Extending existing security policies and standards, leveraging sound physical security protections already in place, and assessing systems and applications for security weaknesses are examples of security design elements that should be included when establishing a secure cloud environment.

IBM Security collaboration

IBM Security represents several brands that provide a broad spectrum of security competency.

- IBM X-Force® research and development teams discover, analyze, monitor, and record a broad range of computer security threats and vulnerabilities
- IBM Managed Security Services (MSS) is responsible for monitoring exploits related to endpoints, servers (including web servers), and general network infrastructure. MSS tracks exploits delivered over the web as well as other vectors such as email and instant messaging.
- Professional Security Services (PSS) delivers comprehensive, enterprise-wide security assessment, design, and deployment services to help build effective information security solutions.
- Our Content security team independently scours and categorizes the web through crawling, independent discoveries, and through the feeds provided by MSS. In addition, the team actively monitors millions of email addresses to receive mass amounts of spam and phishing emails. This work provides optimal spam protection accompanied by the latest trends in spam and phishing emails.
- IBM has collated real-world vulnerability data from security tests conducted over the past three years from the IBM® Rational® AppScan® OnDemand Premium Service. This service combines application security assessment results obtained from IBM Rational AppScan with manual security testing and verification.
- IBM Cloud Security Services allows clients to consume security software features through a hosted subscription model that helps reduce costs, improve service delivery, and improve security.
- Identity and access management solutions provide identity management, access management, and user compliance auditing. These solutions centralize and automate the management of users, authentication, access, audit policy, and the provisioning of user services.
- IBM Endpoint Management Solutions combine endpoint and security management into a single offering that enables customers to see and manage physical and virtual endpoints—servers, desktops, roaming laptops, and specialized equipment such as point-of-sale devices, ATMs and self-service kiosks.

IBM Managed Security Services—A global threat landscape

IBM Managed Security Services (MSS) monitors several billion events in more than 130 countries, 24 hours a day, 365 days a year. The global presence of IBM MSS provides a first-hand view of current threats. IBM analysts use this wealth of data to deliver a unique understanding of the cyber threat landscape. This section focuses on Trojan botnet activity, SQL injection, obfuscation, PDF exploitation, and cross-site scripting activity—threats that are discussed throughout this report. The trend of these threats is vital to determining what direction the threat is taking and to understanding the significance of the threat to our networks.

Trojan Bot networks

IBM MSS saw an upward trend in Trojan botnet activity during 2010. This growth is significant because despite increasing coordinated efforts to shut down botnet activity (as seen with the Mariposa[1] and Bredolab[2] botnets), this threat appears to be gaining momentum. While there have been some successful shutdowns there are many botnets that, due to their resilient and sophisticated Command and Control (CnC) topology, remain largely unaffected by these takedown attempts. Another reason attributing to this growth is the availability of bot exploit toolkits such as WARBOT. This allows less than tech-savvy individuals to take advantage of the lucrative business of selling sensitive information on the black market.

Trojan Bot networks also continued to evolve in 2010. One of them, Zeus (also known as Zbot and Kneber), continues to evolve through intrinsic and plugin advances. The Zeus/Zbot family of botnets

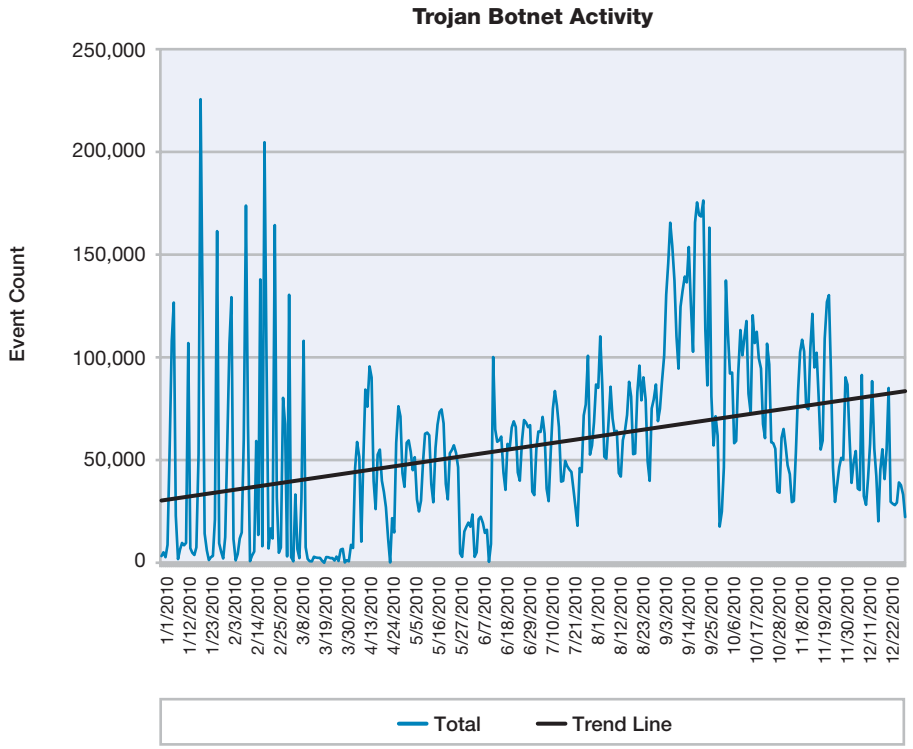


Figure 1: Trojan Botnet Activity

1 Massive Mariposa botnet shut down – http://www.net-security.org/secworld.php?id=8962
2 Bredolab botnet shut down – http://nakedsecurity.sophos.com/2010/10/26/bredolab-botnet-shut/

has been around for many years now and due to its extreme popularity with attackers, there are hundreds, or even thousands, of separate Zeus botnets active at any given time. The Zeus botnet malware is commonly used by attackers to steal banking information from infected computers.

Various bot networks based on Zeus are responsible for millions of dollars in losses over the last few years. For example, Zeus was reportedly responsible for stealing more than $1 million from customers of a single UK-based financial institution in July.[3] The continual arms race between attackers and defenders has botnet controllers finding stealthier ways to keep their bots under the radar. Zeus' merger with SpyEye, a very similar Trojan, is still in its infant stages. How this plays out over time is to be determined, but consolidation amongst Trojan botnets is expected to be an emerging trend.

In April, we saw a spike in malicious PDF activity associated with Zeus.[4] Attackers abused the "Launch" feature in Adobe Acrobat to distribute the Zeus botnet malware via email. The signature PDF_Launch_Program detects the network transfer of a PDF file containing an embedded action to Launch an executable program. Adobe Reader asks for user confirmation before actually launching the application, but certain versions of Foxit Reader do not and merely start the application without user confirmation. In cases where organizations have moved away from Adobe's implementation, this is of particular concern with regards to this attack.

Zeus' encrypted command and control activity is hard to detect. However, one of the signatures analyzed to assess this threat focuses on a type of behavior that Zeus might exhibit. The signature HTTP_Suspicious_Unknown_Content detects when a HTTP POST message results in a session where the content sent and received is not recognized as typical content, such as images or documents. Activity associated with this signature seemed to grow in intensity towards the latter half of 2010. Such activity could be normal or could indicate botnet activity. While this is a generic signature, we do believe that this activity is associated with Zeus. The section titled "Zeus botnet—facts, myths and understanding how these botnets operate" in the **2010 Mid-Year Trend and Risk Report** provides an in-depth explanation of Zeus and how readers can protect themselves from this threat.

There was also significant activity associated with the Waledac botnet at the start of the year up until early March and then the activity seemingly disappears for the rest of 2010. What could have caused this dramatic drop? We speculate that the cessation in activity is the result of "Operation b49".[5] This Microsoft led operation resulted in the takedown of a majority of this botnet in late February. Once a temporary restraining order was granted on February 22nd, much of the communication between Waledac's command and control centers and its thousands of zombie computers was cut off in a matter of days. In October, the U.S. District Court of Eastern Virginia ordered the permanent transfer of ownership of the 276 domains behind Waledac to Microsoft.[6] Does this mean that Waledac will never surface again? We may see activity, but probably not to the same magnitude that we observed prior to the takedown.

Another prevalent botnet is Pushdo (also known as Pandex and some components are known as Cutwail). This botnet generated noticeable activity across the IBM MSS network in 2010 though to a lesser extent than Waledac and Zeus. Pushdo, primarily used for spamming, had been observed launching Distributed Denial of Service (DDoS) attacks against certain SSL-enabled websites beginning in the first quarter 2010. The DDoS attack involved sending thousands of malformed SSL requests to the target hosts in an attempt to use up resources. To a business, this could directly impact revenue if services provided or product sales are interrupted during such an attack.

3 Targeted Attack Nets 3,000 Online Banking Customers – http://www.darkreading.com/smb-security/security/attacks/showArticle.jhtml?articleID=226600381
4 PDF-based Zeus attacks – http://www.iss.net/threats/PDFbasedZeusAttack.html
5 Cracking Down on Botnets – http://blogs.technet.com/b/microsoft_on_the_issues/archive/2010/02/24/cracking-down-on-botnets.aspx
6 R.I.P. Waledac – Undoing the damage of a botnet http://blogs.technet.com/b/microsoft_blog/archive/2010/09/08/r-i-p-waledac-undoing-the-damage-of-a-botnet.aspx

SQL injection

SQL injection is one of the leading attack vectors seen because of its simplicity to execute and its scalability to compromise large amounts of web servers across the Internet. A review of past X-Force Trend and Risk Reports reveals an interesting SQL injection trend. During each of the past three years, there has been a globally scaled SQL injection attack some time during the months of May through August. The anatomy of these attacks is generally the same: they target .ASP pages that are vulnerable to SQL injection. The surges that occurred during 2008 and 2009 are shown in Figure 2.

In 2008, attackers used a SQL CAST statement and some hex code to obfuscate the true injection string. The source of this attack was the Asprox botnet, and it was massively successful in compromising thousands of websites. In 2009, we observed the same attack methodology; the only difference was in the resulting payload. Asprox was again the source of this attack. However, it had varied success this time because of countermeasures that were deployed to thwart the attack.

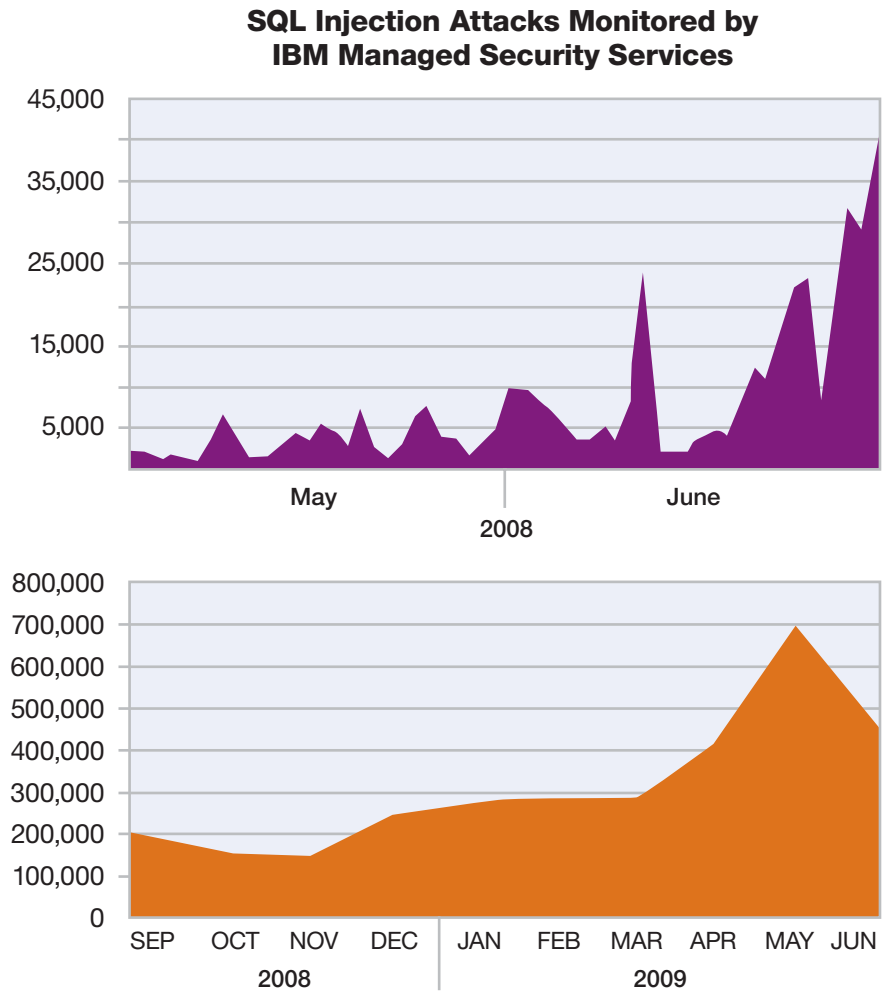


Figure 2: SQL Injection Attacks Monitored by IBM Managed Security Services

Figure 3 illustrates the significant SQL injection attack observed in 2010 as detected by the IBM signature SQL_Injection_Declare_Exec. The same attack methodology is used as in the previous two years, but some of the mechanics were changed. Attackers added leetspeak (1337) to the SQL statement to evade poorly written regex filtering. This statement, once decoded, contains another CAST statement resulting in two layers of obfuscation. While very similar to Asprox, this attack used slightly different techniques and therefore is known more popularly as the "dnf666" attack—so named because of a URL encoded inside.

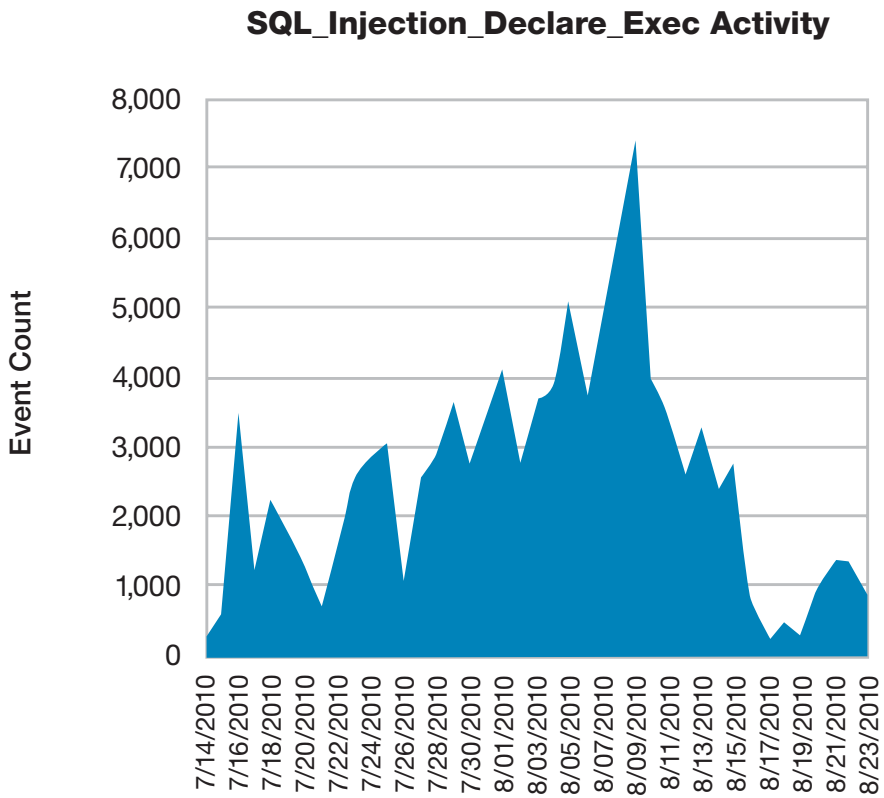


Figure 3: SQL_Injection_Declare_Exec Activity

Obfuscation

IBM MSS continues to track trends in obfuscation techniques used by attackers and toolkits. Obfuscation is a technique to hide or mask the code used to develop applications. New obfuscation methods are constantly evolving in an attempt to evade intrusion prevention systems (IPS) and anti-virus which often can't decode the web page or file to find the hidden attack. Through special detection algorithms incorporated into IBM Security Network IPS, we watch how patterns of use change by monitoring hits on these algorithms in our world-wide MSS deployments.

Obfuscation activity continued to increase during 2010 and shows no signs of waning. The most observed activity came from an event that triggers when a JavaScript 'unescape()' function with a large amount of escaped data is detected. This activity should be viewed with suspicion. It may be normal activity, or it could indicate the attempt to inject a large amount of shell code or malicious HTML and/or JavaScript for the purpose of taking control of a system through a browser vulnerability.

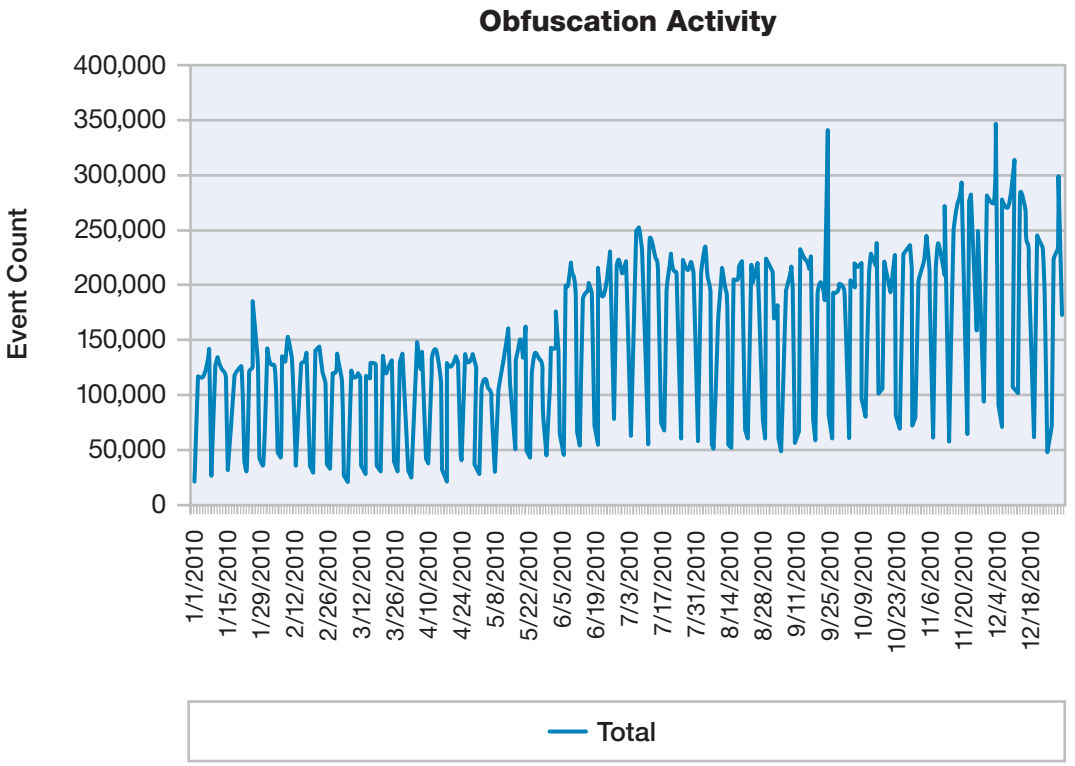


Figure 4: Obfuscation Activity

PDF exploitation

Compromise through PDF exploitation continues to be a favorite among attackers. Throughout 2010, our global Security Operation Centers witnessed surges of malicious traffic surrounding spam email. One notable increase occurred in late April, as shown in Figure 5. The emails of this particular spam campaign contained an Adobe Acrobat PDF that used the Launch command to deliver malware. At the peak of the attacks, IBM MSS received more than 85,000 alerts in a single day. The spam email was sent from various SMTP servers globally, which appeared to originate from the Zeus botnet.

There has been a small but steady rise in PDF exploitation since the beginning of 2010. There are numerous signatures that contribute to this assessment. Some of these signatures detect an unauthorized access attempt. For example, one signature detects a file with embedded corrupt JBIG2 data that could cause a buffer overflow in vulnerable versions of Adobe Acrobat and Adobe Reader. (Note: This is fixed in Adobe Acrobat/ Reader 8.1.3.) Other signatures may simply be looking for suspicious activity such as a PDF file containing a hex-encoded form of a filter name. This suggests malicious intent by concealing compressed content within the document.

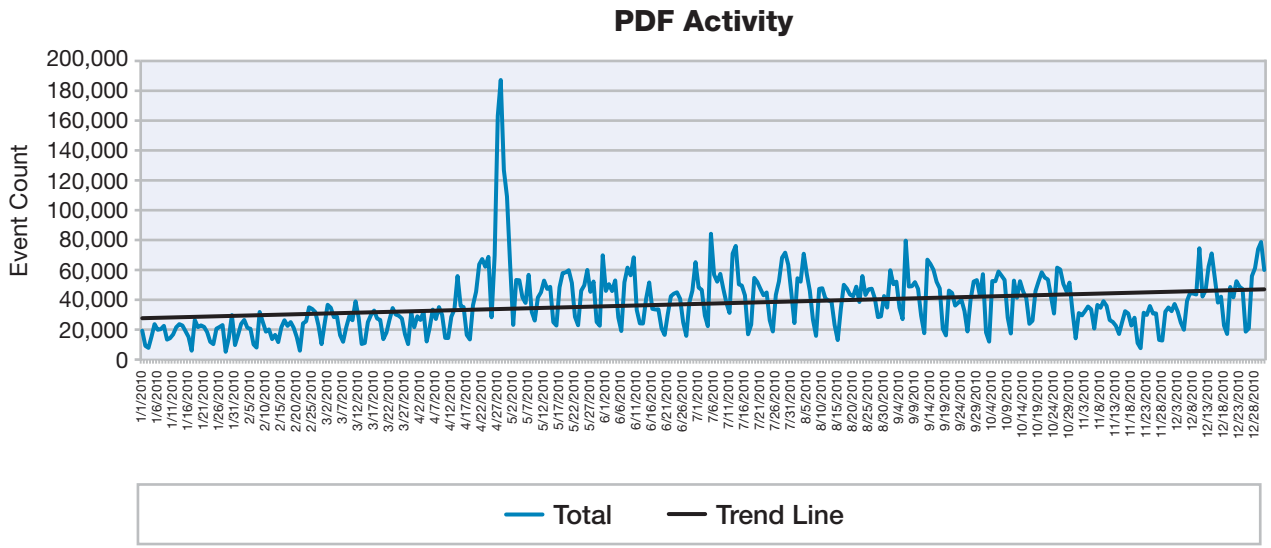


Figure 5: PDF Activity

Cross-site scripting

While cross-site scripting vulnerabilities continue to be one of the predominant types of vulnerabilities affecting web applications, activity targeting these vulnerabilities seems to have leveled off in 2010 as shown in Figure 6. Cross-site scripting allows attackers to embed their own script into a page the user is visiting, thereby manipulating the behavior or appearance of the page. These page changes can be used to steal sensitive information, manipulate the web application in a malicious way, or embed additional content on the page that can exploit other vulnerabilities.

Though the trend is flat, it does not mean that this threat is non-existent. From a Common Vulnerability Scoring System (CVSS) scoring perspective, these vulnerabilities do not typically rank as high or critical threats. IT and security professionals tend to deploy counter measures for the high-profile vulnerabilities first and, if resources allow, later address the low- to medium-rated issues. Attackers, therefore, will continue to take advantage of this window of opportunity in years to come.

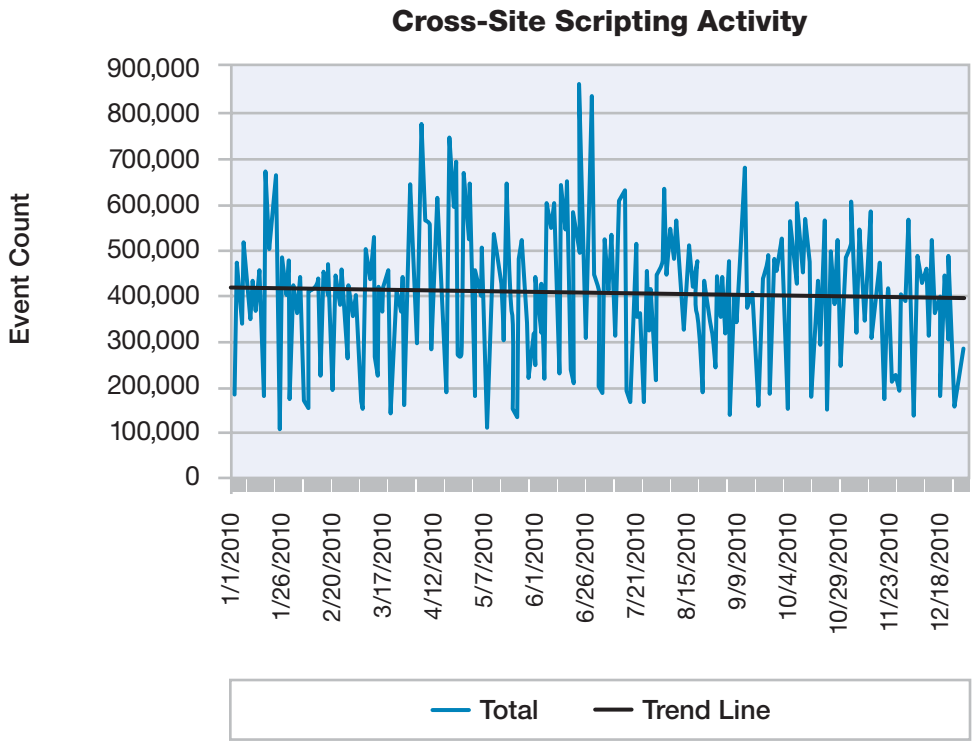


Figure 6: Cross-Site Scripting Activity

Industry trends

There is great interest in the general security community in knowing which industries are being targeted by what attack types. Our customer base is broad and reaches into a number of different industries. However, to identify a valid trend across a particular industry, we needed to establish a methodology with an acceptable sample size for analysis. For each attack category, we only assessed activity where a specific criterion was met in a given industry. A minimum number of affected customers and a minimum number of devices deployed amongst those customers was required prior to making an assessment.

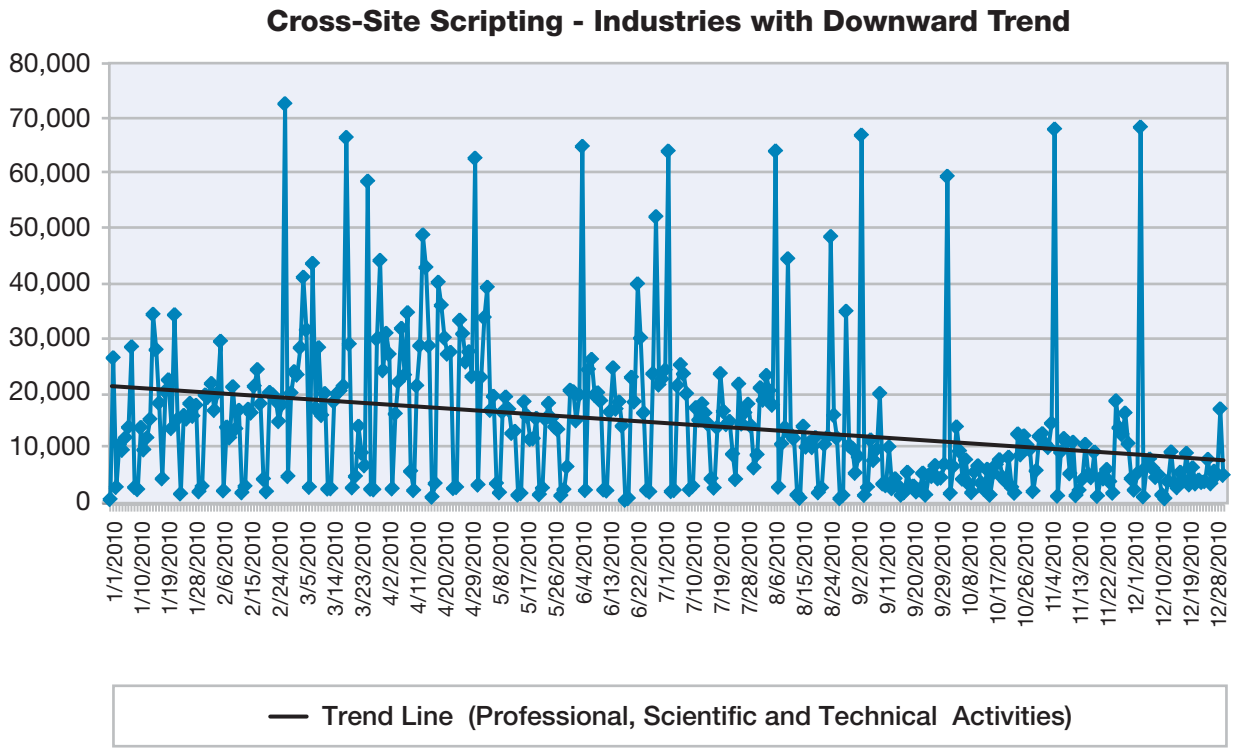


Figure 7: Cross-Site Scripting – Industries with Downward Trend

What did we see? Generally speaking, we did not see any significant discrepancies across different industries regarding the varying attack types compared to overall customer trends. Attack trends across all industries were relatively uniform.

What can be deduced from this? While some attacks are targeted, many exploits in circulation simply don't discriminate. A financial organization may be just as vulnerable to the latest botnet or PDF exploitation as an educational institution. Whether or not an organization is vulnerable to attack has much more to do with the protection measures that they have in place.

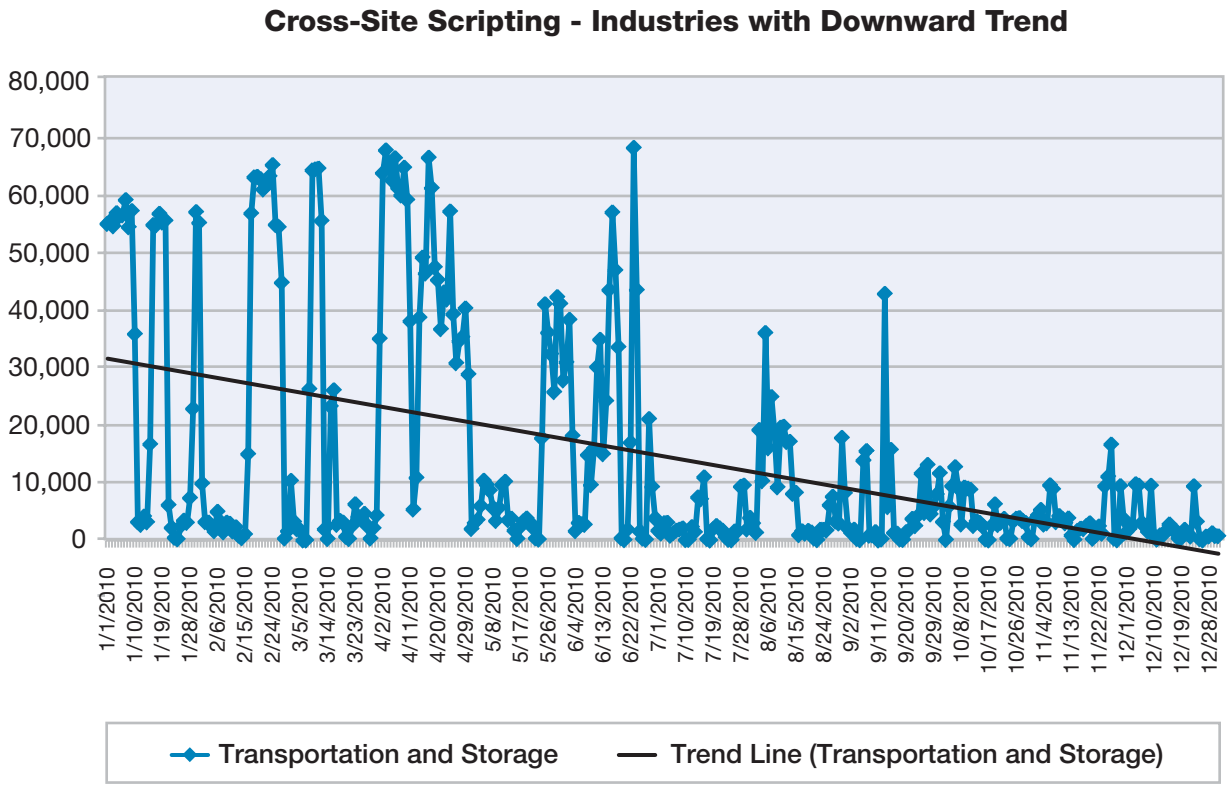


Figure 8: Cross-Site Scripting – Industries with Downward Trend

The only exception to our findings of consistent trends among the industries was in the cross-site scripting category. As shown in Figure 6, the overall trend for cross-site scripting was relatively flat and several industries followed this trend. As shown in Figures 7 through 9, a few industries saw a slight downward trend in this attack category including:

- "Professional and Scientific"
- "Wholesale and Retail Trade"
- "Transportation and Storage"

A decrease in cross-site scripting activity may indicate greater attention to addressing these types of vulnerabilities. **As noted later in this report, the IBM Rational AppScan on Demand Premium service that tracks web application vulnerabilities has also seen a steady decline in the instances of cross-site scripting reported vulnerabilities since 2007.** Part of this decline is attributed to a greater awareness of the risk associated with cross-site scripting.

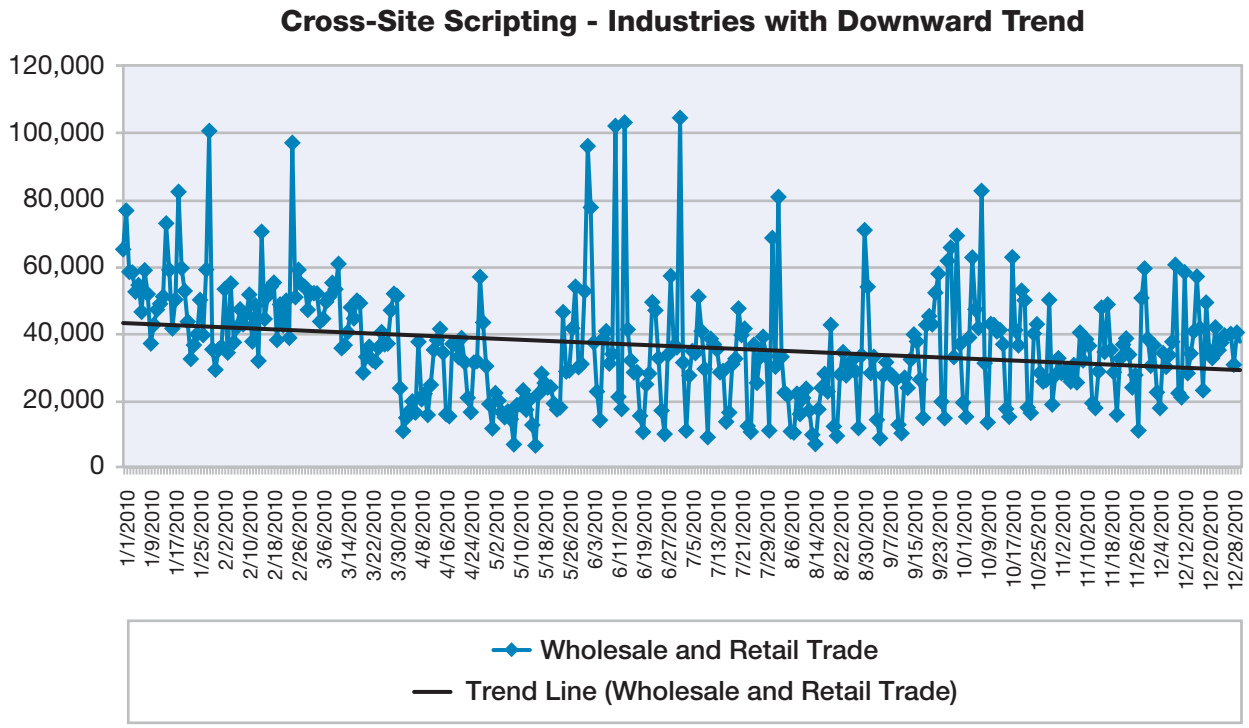


Figure 9: Cross-Site Scripting – Industries with Downward Trend

Top high-volume signatures—IBM MSS

Table 1 to the right, shows the placement of the top MSS high volume signatures and their trend line for 2010.

The top high volume signatures seen across the MSS network reveal some interesting aspects of life on the Internet today and are a reflection of the longevity of certain threats. For example, the SQL Slammer worm[7] first surfaced in January 2003 and became known as one of the most devastating Internet threats of the past decade. Despite the downward trend in 2010, this worm still exists and continues to propagate as evidenced by the top ranking signature, SQL_SSRP_Slammer_Worm shown in Table 1. SQL Slammer targets a buffer overflow vulnerability in the Resolution Service in Microsoft SQL Server 2000 or Microsoft Desktop Engine (MSDE) 2000 installations. This issue was patched by Microsoft in 2002. The fact that there is such a huge volume of activity associated with SQL Slammer seven years after it first surfaced probably suggests a need for better patch management.

Rank	Event Name	Trend Line
1	SQL_SSRP_Slammer_Worm	Down
2	SQL_injection	Down
3	PsExec_Service_Accessed	Slightly Up
4	SSH_Brute_Force	Slightly Down
5	JScript_CollectGarbage	Up
6	HTTP_Unix_Passwords	Slightly Up
7	SMB_Mass_Login	Down
8	SMB_Empty_Password	No Change
9	SQL_Empty_Password	Up

Table 1: Top MSS high volume signatures and trend line

7 SQL slammer traffic on the Internet significantly declined in March 2011 shortly before publication of this report. For more information on this topic, please see the Frequency-X blog. (http://blogs.iss.net/index.html)

Targeting SMB Servers

Two of the top signatures protect against threats targeting server message block (SMB) servers. The SMB_Empty_Password detects when a successful connection with no password is made to an SMB server. If this connection is from outside the network, consider the information on your server as compromised. The SMB_Mass_Login signature detects an excessive number of granted NetBIOS sessions originating from the same IP address. This may indicate a stolen account being used in a scripted attack. The existence of these signatures in the list highlights a possible lack of basic security with SMB shares. If attackers are attempting to connect to SMB servers with no password, this signifies that this method of attack continues to be fruitful for attackers. Recent threats, such as the Conficker and Stuxnet malware, use SMB shares to spread across networks.

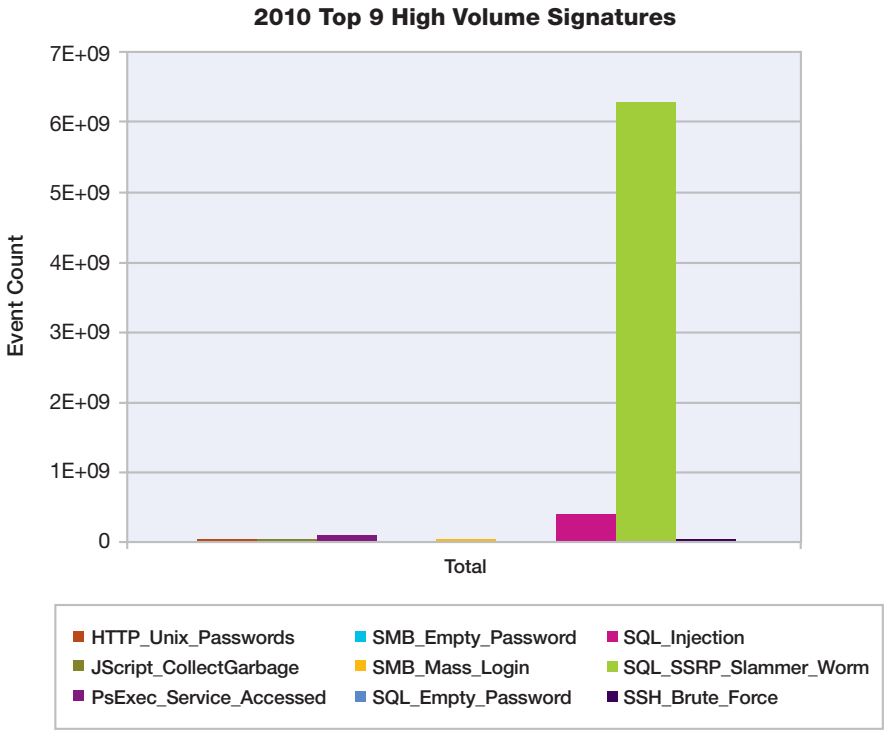


Figure 10a: 2010 Top 9 High Volume Signatures

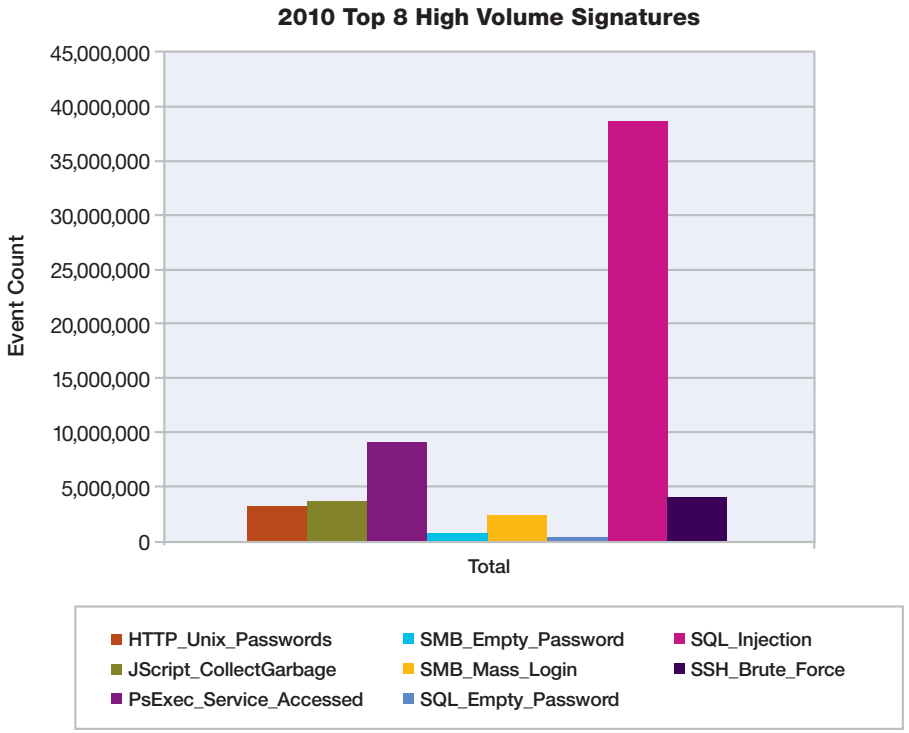


Figure 10b: 2010 Top 8 High Volume Signatures

SQL injection—high volume

Our heuristic SQL signature had the second highest volume seen in 2010. This is not surprising because SQL injection attacks against web applications are very common. IBM MSS has observed a seasonal surge in SQL injection attacks during the months of May through August for the past three years as discussed in the section **IBM Managed Security Services**—A global threat landscape. The other SQL signature noted in Table 1, SQL_Empty_Password, detects when a successful connection with no password is made to an SQL server. As with the SMB_Empty_Password signature, these types of connections should be considered suspicious if made from outside the network.

PsExec—A remote administration tool

The signature in the third spot, PsExec_Service_Accessed, is notable in that PsExec is a legitimate application. It is a command line based remote administration tool. However, worms and advanced threats also take advantage of PsExec. The "Here you have" worm, for instance, includes a PsExec tool that allows it to copy itself onto other computers over the network. If this application is used in your organization, you should ensure that best security practices are employed.

Brute force attacks & scans

SSH_Brute_Force is another interesting signature in this list. A brute force attack involves an attacker trying to gain unauthorized access to a system by trying a large number of password possibilities. This signature detects an excessive number of SSH Server Identifications from an SSH server within a specified timeframe. Through this type of attack, a malicious individual may be able to view, copy, or delete important files on the accessed server or execute malicious code. Organizations can help mitigate brute-force attacks by disabling direct access to root accounts and using strong usernames and passwords.

We provided an in-depth view on this topic in the **2010 Mid-Year Trend and Risk Report** where we explain the nature of a Darknet. A Darknet is a black-hole network whose addresses are not allocated to any active legitimate device or service on the Internet. When an attacker attempts a brute-force attack on a particular address in the Darknet they never connect to an SSH server because one does not exist. Therefore, they stop after one attempt. Conversely, a successful SSH connection may result in thousands of brute force attempts which explains the large volume of activity associated with SSH_Brute_Force.

The Darknet data in that mid-year report shows that the level of SSH brute force scanning is steadily increasing while the MSS data shows that the level of brute force attacks against active SSH servers is high.

JScript & UNIX

JScript_CollectGarbage detects the transfer of a JScript file containing a call to the function CollectGarbage(). CollectGarbage() is part of the .NET framework but, according to Microsoft, "is not intended to be used directly from your code." This function has been used by attackers and can be indicative of malicious intent. However, it can also be used for legitimate purposes.

Finally, the HTTP_Unix_Passwords signature detects attempts to access the /etc/passwd file on UNIX systems via a web (HTTP) server. While this activity is sometimes authorized, it can sometimes be suspicious. This is a very old attack, but is still successful today.

Trending in the dark—what does malicious traffic look like?

As we discussed in the previous section, one of the many data resources that IBM security analysts use to determine trending is the darknet, also known as a black-hole network. A darknet is a large range of IP addresses on the Internet that have never had any services running on them. Our darknet has an aperture of 25,600 addresses. Generally speaking, there is no legitimate reason why computers on the Internet would send packets to addresses in this range, but in fact they do. Often, traffic into this network is associated with malicious activity. This space is continuously monitored and all incoming traffic is captured in its entirety and stored for analysis and long-term archiving.

Spoofed Denial of Service attacks

Looking at the data over the past several years, a couple of patterns begin to emerge. The first trend is the gradual rise in backscatter activity (Figure 11). Backscatter is actually a side effect of a spoofed Denial of Service (DoS) attack. Attackers launching Denial of Service attacks on the Internet will often put incorrect source addresses in the packets they are flooding at their victim. This is known as spoofing. By spoofing randomly selected source addresses, the attacker makes it difficult for the victim's system to distinguish between the spoofed packets and legitimate packets from real users. The victim system will respond to a certain percentage of these spoofed packets. These responses are known as backscatter. If an attacker randomly selects an IP address in our darknet range, and the victim responds, we'll collect that response. By studying these responses we can learn things about Denial of Service activity on the Internet.

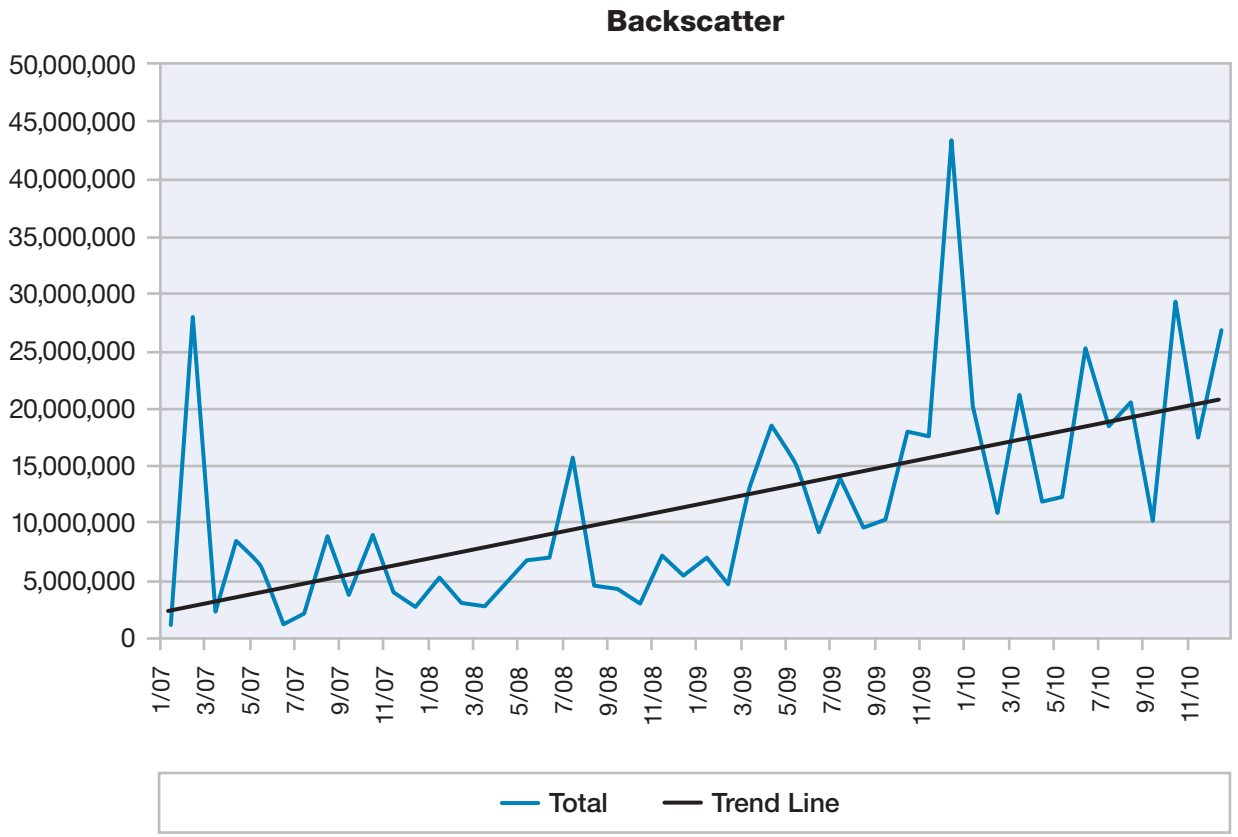


Figure 11: Backscatter

In the X-Force darknet, each SYN-ACK backscatter packet received is an indicator that an attacker sent a spoofed packet to a well-known service port on the machine under attack spoofed from one of X-Force darknet addresses. While there has been a gradual increase in backscatter activity since 2007, there was a large jump year-over-year between 2008 and 2009. Part of this increase is due to a significant spike in activity in 2009—the largest in the three and half year period. This trend of higher than previous year averages continues in 2010. At the close of Q2, the average count for the first half of 2010 is slightly higher than the total average for 2009, just over 16.5 million. At the close of the year 2010 we see that this number has now jumped to over 18 million. Figure 12 indicates the increase in volume from 2007 through 2010 of spoofed Denial of Service attacks on the Internet.

What can we deduce from this gradual rise in backscatter data and, in some instances, large jumps of backscatter activity? Since the majority of the backscatter data results from Denial of Service (DoS) attacks, we can speculate that there has been a steady increase in spoofed DoS attacks since 2007. However, backscatter is subject to a high degree of variability due to the nature of what is being collected and what is occurring. Some intense periods of backscatter are the result of internecine warfare within and between various attacker camps. During this warfare, one group attempts to block or take over the resources of another group. This "shelling match" between warring camps can result in a sudden increase in backscatter traffic and backscatter source addresses. It generally ceases as suddenly as it began. This type of activity most likely contributed to the dramatic spikes in February 2007 and December 2009 as shown in Figure 11 on page 24.

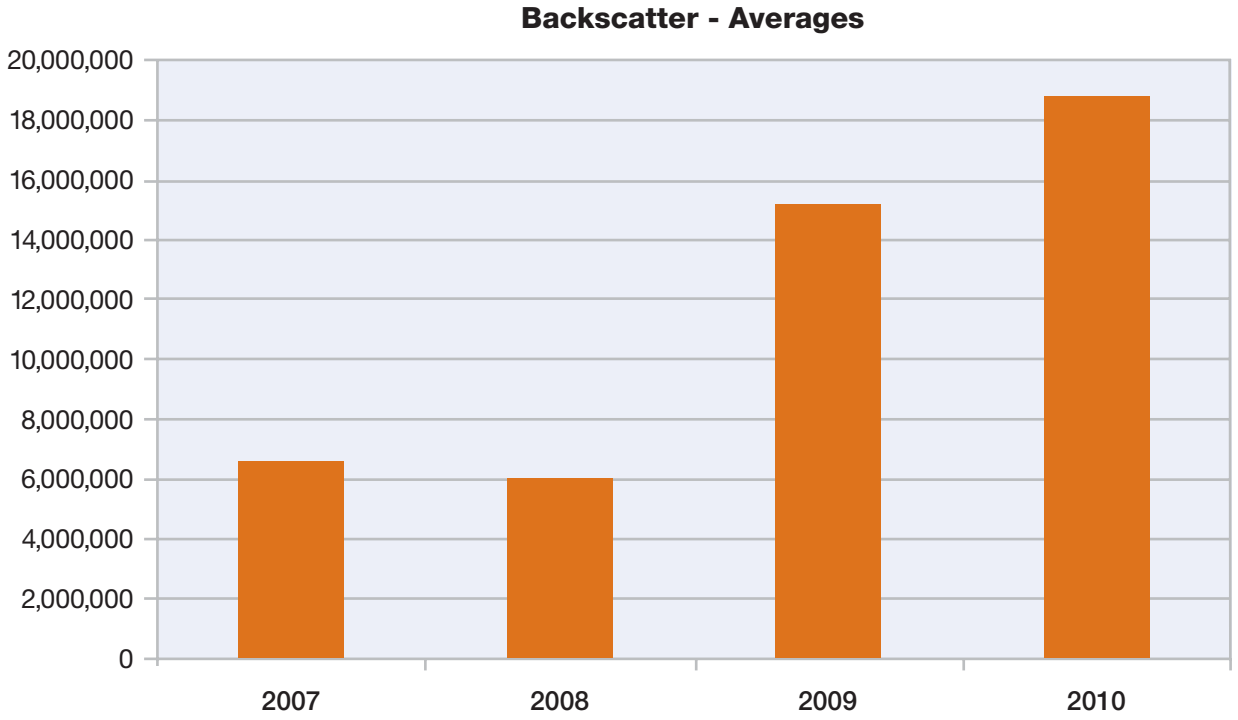


Figure 12: Backscatter – Averages

Targets of Denial of Service attacks

The nature of a spoofed Denial of Service attack makes it difficult to determine the attacker. The attacker fabricates origins for the connections to the victim's IP address. These fabricated connections can in turn come from a multitude of different machines. When looking at backscatter in the X-Force darknet, it is clear that the origins of the attack are spoofed, but the target of the attack is known. Examining the sources of the backscatter provides information on the targets of spoofed Denial of Service attacks. Figure 13 shows the top backscatter-generating countries for the second half of 2010 as calculated using the WorldIP database that maps addresses to countries.

There is a fairly common trend in the data. The United States is by far the largest generator, China is second, and Turkey is third. The United States and China have the first and second largest counts of IP addresses so their ranking as backscatter generators isn't surprising. If any IP address is as likely to be a target as any other then one would expect to see Japan, Germany, South Korea, or the UK in the top three. This assumption is clearly wrong. Further analysis and correlation with other data may help shed light on the matter.

For more discussion about brute force attacks and the information reported earlier in the year, please refer to the **2010 Mid-Year Trend & Risk report** located on our web page under report archives.

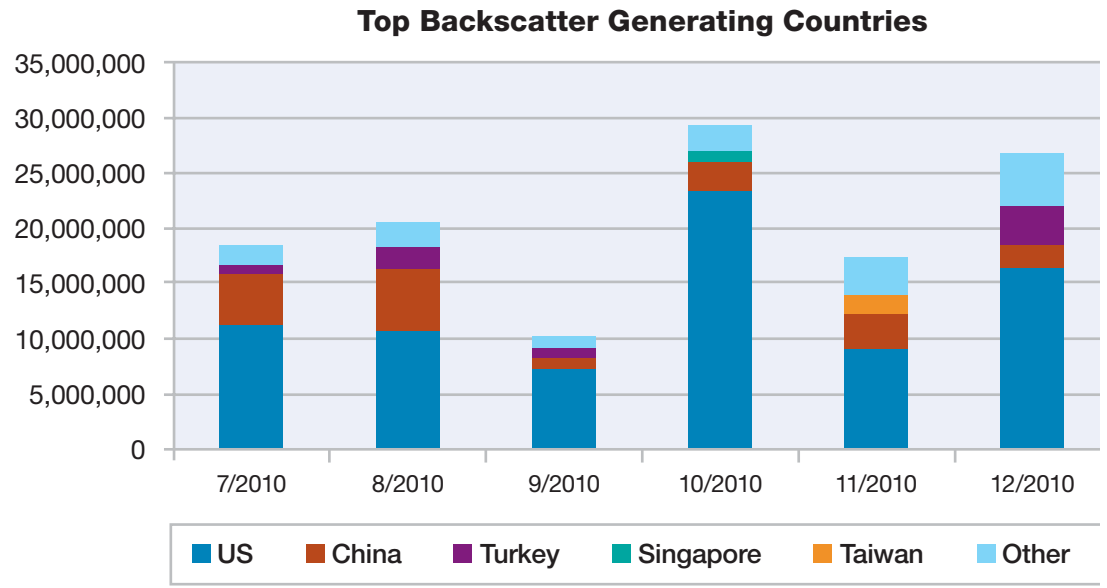


Figure 13: Top Backscatter Generating Countries

Worms of yesteryear: Where are they now?

The ongoing war against the threat of computer worms is cyclic. A new invader appears, after the battles to contain the infection and the initial outbreak appears won, it falls off the collective radar as years pass and the next invaders appear.

Worms propagate by a number of methods such as malicious email attachments, open or weakly protected shares and network accessible software vulnerabilities. A number of prominent worms have appeared over the last seven years but those that spread via exploitation of network accessible vulnerabilities tend to be the most virulent. They can spread across networks from machine to machine without a user interceding to view an email or open a file. The autonomous spreading of these worms can lead to high infection rates and frequently, disastrous side effects occur of machines crashing from unreliable exploitation and potentially crippling network utilization for virulent worms.

IBM's Managed Security Service tracks the malicious activity seen on its customer's networks and thus affords a window into the activity of these worms of yesteryear. The following list gives an overview of five of the most recent worms that spread entirely or partly by exploiting operating system vulnerabilities. All of these worms targeted software, usually operating systems, by Microsoft.

SQL Slammer first appeared in late January of 2003, generating such a deluge of traffic that it brought down numerous critical resources and noticeably slowed the Internet. Its single UDP packet payload targeted a vulnerability in Microsoft SQL Server that had been patched previously in July. The compromised host would then loop, spamming copies of itself to random IP addresses, DoSing (Denial of Service) itself and sending out a large amount of traffic.

Blaster appeared in August of 2003 and rapidly spread. This worm propagated by exploiting a buffer overflow in the Remote Procedure Call (RPC) interface and the Distributed Component Object Model (DCOM) interface that had been patched a month earlier. The worm payload would install an auto-starting executable that would continue trying to propagate and trigger a Denial of Service against Microsoft's update site at a specific time.

Sasser appeared at the end of August in 2004. It propagated by exploiting a vulnerability in the Local Security Authority Subsystem Service (LSASS), which is a process in Microsoft Windows operating systems that is responsible for enforcing the security policy on the system. LSASS patched a few weeks previously. Once infected, a machine would download and install an auto-starting executable which would scan and attempt to infect other machines. The worm itself wasn't malicious but a side effect of its scanning caused crashes and reboots in servers and desktops which had severe consequences for many companies.

The **Zotob** worm appeared in mid-August of 2005. It propagated by exploiting a buffer overflow in the Microsoft Plug and Play service that was patched earlier in the month. A side effect of this propagation was crashing and reboots of machines due to the exploit. Infected machines would download and install an executable to continue propagation and install a backdoor to phone back to an Internet Relay Chat (IRC) channel for further instructions.

Conficker was detected in early November of 2008. Propagation was via a vulnerability in the server service of all supported Microsft operating systems at the time. Later variants added additional vectors such as weak SMB passwords and infection of USB devices. Once compromised, the infected machine would attach itself to a botnet awaiting further commands. Most variants of this worm also performed controlled scanning and infection of further hosts.

Year	Worm	Vulnerability	IBM Signature	MS Bulletin
2003	SQL Slammer	CVE-2002-0649	SQL_SSRP_StackBo	MS02-039
2003	Blaster	CVE-2003-0352	MSRPC_RemoteActivate_Bo	MS03-026
2004	Sasser	CVE-2003-0533	MSRPC_LSASS_Bo	MS04-011
2005	Zotob	CVE-2005-1983	PlugAndPlay_BO	MS05-039
2008	Conficker	CVE-2008-4250	MSRPC_Srvsvc_Path_Bo	MS08-067

Table 2: Top Worms 2003 - 2008

Figure 14 breaks down the alert activity by worm. For clarity, the alert associated with the worm activity has been renamed for the worm. In most cases this network activity is based on detected exploitation by the worms but this is a tricky endeavor for a number of reasons. For one, the alert is not necessarily an indication of an attempted propagation by a worm, alerts can be due to a security audit or an exploitation attempt by something else entirely. Another issue is that worms have different propagation rates. Conficker regulates its propagation in an attempt to avoid overt detection while SQL Slammer can spam hundreds of exploitation attempts a second. Due to the number of ways that Conficker variants can spread, counts of peer to peer activity were used.

SQL Slammer[8] has by far the largest number of exploitation attempts. Even though seven years have passed in which time to remediate the worm, it remains extremely noisy. The large dip in activity between July and August is due to remediation in a single network. Ever afterwards, Slammer counts overshadow all others.

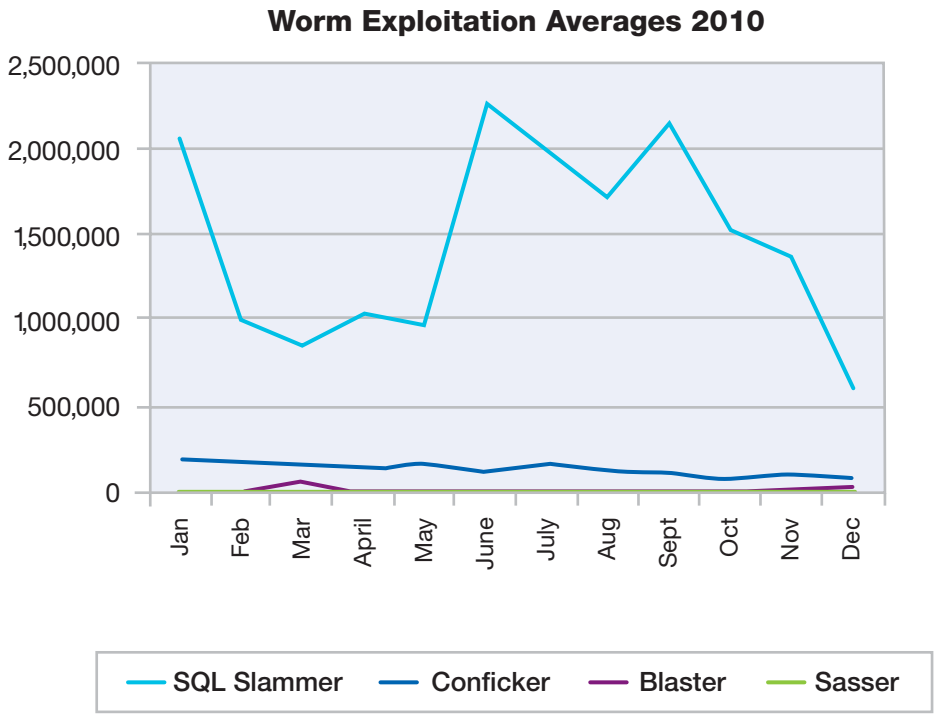


Figure 14: Worm Exploitation Averages 2010

8 SQL slammer traffic on the Internet significantly declined in March 2011 shortly before publication of this report. For more information on this topic, please see the Frequency-X blog. (http://blogs.iss.net/index.html)

Figure 15 shows the same monthly averages with SQL Slammer removed. Conficker traffic is the next highest. This is not surprising as it is the most recent of the studied worms and also known to be extremely widespread. There is a noticeable decline in activity over the year, likely attributed to infected nodes being cleaned or brought offline. Blaster and Sasser are still showing activity while Zotob's counts were so low that it was removed from the figure. This discrepancy may be due to the fact that Blaster and Sasser would affect both Windows 2000 and XP and they came at an earlier time while Zotob only affected Windows 2000 and came out in 2005.

It is interesting to note, that the worms exploiting vulnerabilities patched over seven years ago still show noticeable activity. The activity for all the worms is unlikely to grow significantly as any new machine brought online should not be vulnerable to the exploits they spread by. It seems inevitable that the activity from these worms will eventually die out as old infected machines are replaced but they do show a remarkable tenacity.

New worms will always be on the horizon. As the latest invader is brought under control and gradually driven out, it will likely never be fully ousted. There will almost certainly be a few survivors holding out in the dark corners of our networks.

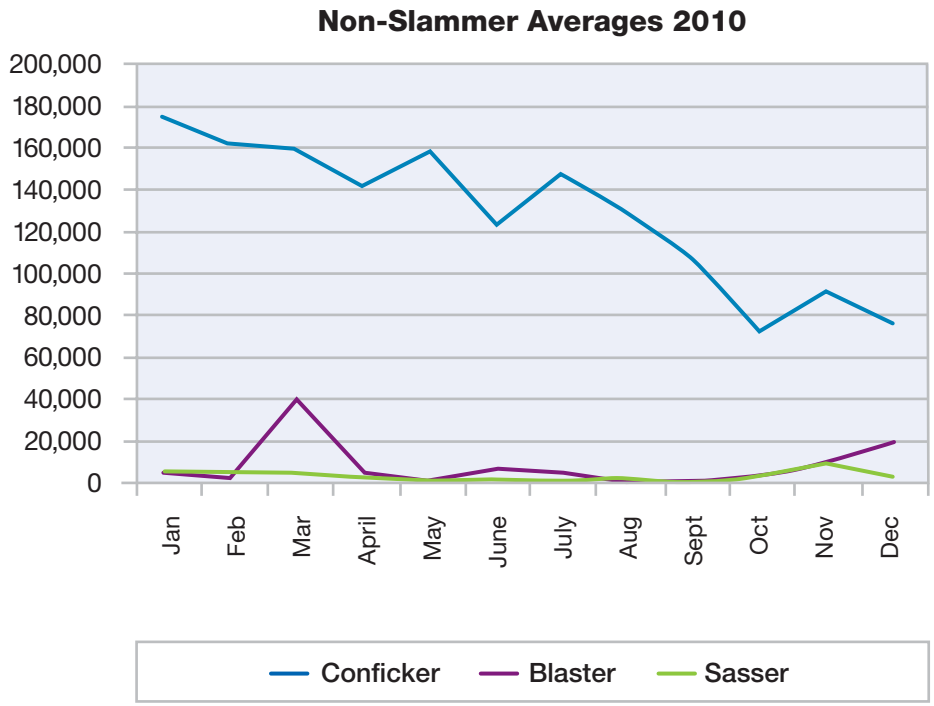


Figure 15: Non-Slammer Averages 2010

Web content trends

This section summarizes the amount and distribution of "bad" web content that is typically unwanted by businesses based on social principles and corporate policy. Unwanted or "bad" Internet content is associated with three types of websites: adult, social deviance, and criminal.

The web filter categories are defined in detail at: **http://www-935.ibm.com/services/us/index.wss/detail/iss/a1029077?cntxt=a1027244**

Table 3 below lists the IBM web filter categories that correspond with these types of sites.

This section provides analysis for:

- Percent and distribution of web content that is considered bad, unwanted, or undesirable
- Increase in the amount of anonymous proxies
- Malware URLs: Hosting countries and linkage

Website Type	Description & Web Filter Category
Adult	Pornography Erotic / Sex
Social Deviance	Political Extreme / Hate / Discrimination Sects
Criminal	Anonymous Proxies Computer Crime / Hacking Illegal Activities Illegal Drugs Malware Violence / Extreme Warez / Software Piracy

Table 3: Web filter categories associated with unwanted web content

Analysis methodology

X-Force captures information about the distribution of content on the Internet by counting the hosts categorized in the IBM Security Solutions web filter database. Counting hosts is a method for determining content distribution and generally provides a realistic assessment. Results may differ when using other methodologies such as counting web pages and sub-pages.

The IBM Content data center constantly reviews and analyzes new web content data. The IBM Content data center analyzes 150 million new web pages and images each month and has analyzed 14 billion web pages and images since 1999!

The IBM Web Filter Database has 68 filter categories and 67 million entries with 150,000 new or updated entries added each day.

Percentage of unwanted Internet content

Approximately seven percent of the Internet currently contains unwanted content such as pornographic or criminal websites.

Increase of anonymous proxies

As the Internet becomes a more integrated part of our lives—not only at home, but at work and at school—organizations responsible for maintaining acceptable environments increasingly find the need to control where people can browse in these public settings.

One such control is a content filtering system that prevents access to unacceptable or inappropriate websites. Some individuals attempt to use an anonymous proxy (also known as web proxies) to circumvent web filtering technologies.

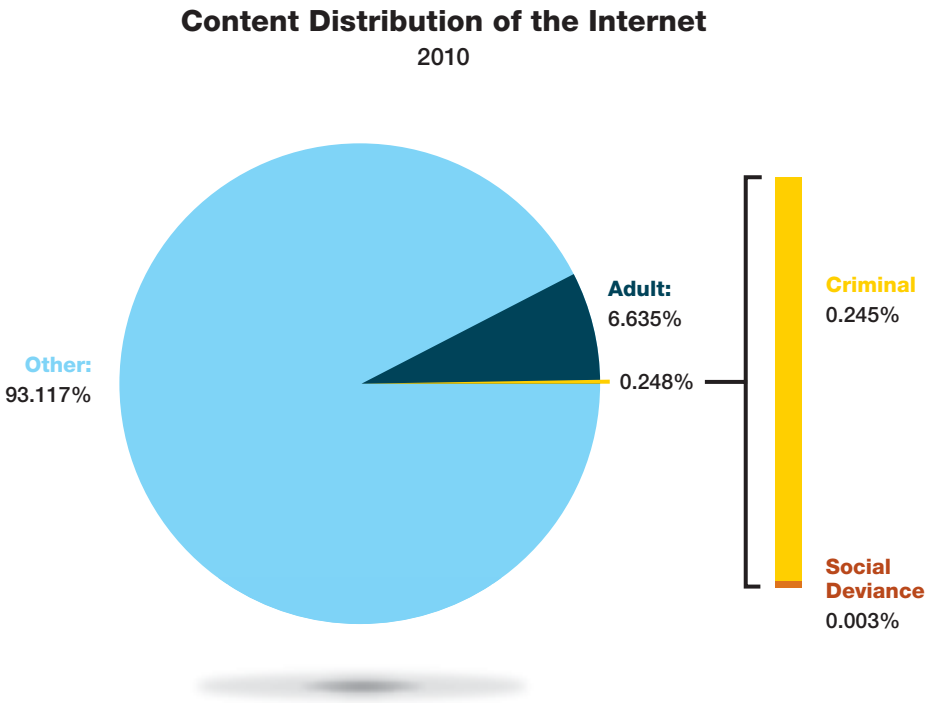


Figure 16: Content Distribution of the Internet – 2010

Web proxies allow users to enter a URL on a web form instead of directly visiting the target website. Using the proxy hides the target URL from a web filter. If the web filter is not set up to monitor or block anonymous proxies, then this activity (which would have normally been stopped) can bypass the filter and allow the user to reach the disallowed website.

The growth in volume of anonymous proxy websites reflects this trend.

In the past three years, anonymous proxies have steadily increased, more than quintupling in number. Anonymous proxies are a critical type of website to track, because of the ease at which proxies allow people to hide potentially malicious intent.

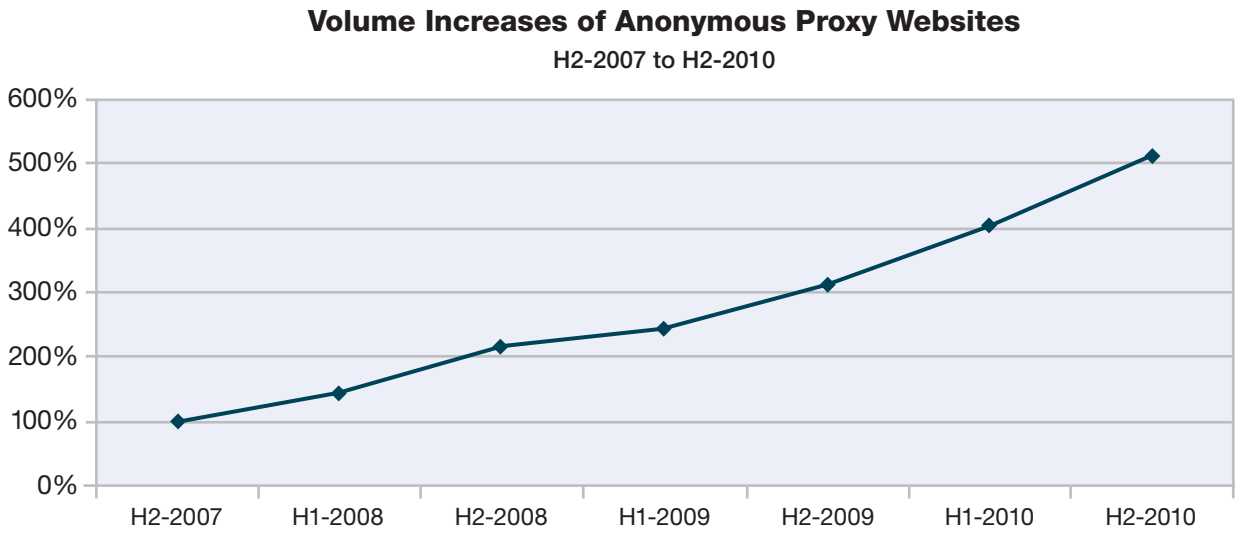


Figure 17: Volume Increases of Anonymous Proxy Websites – H2-2007 to H2-2010

Top Level Domains of Anonymous Proxies

Figure 18 illustrates the Top Level Domains (TLDs) of the newly-registered anonymous proxies.

In 2006, more than 60 percent of all newly-registered anonymous proxies were .com domains, but since the middle of 2007, .info has been at the top until the beginning of 2010 (while .com was runner-up most of the time).

But why is .info no longer in the prime position? It seemed to be a proven TLD for anonymous proxies for years. A reason could be that .info, similar to .com, is running out of names. Additionally, the question arises why anonymous proxies are now provided on .cc and .tk top level domains. These are the Domains of Cocos (Keeling) Islands (.cc), an Australian territory, and Tokelau (.tk), a territory of New Zealand. Nearly all .cc anonymous proxy websites are registered on the domain co.cc. It is free of charge to register a domain anything.co.cc (see **http://www.co.cc/?lang=en**). The same is true for .tk. (see **http://www.dot.tk/**). Thus, it is both cheap and attractive to install new anonymous proxies on .co.cc or .tk.

Additional trends:

- At the end of 2009, .cc (Cocos (Keeling) Islands) started to increase significantly and even reached the number one position in the second quarter of 2010. Nevertheless .cc went out of vogue by the end of 2010.

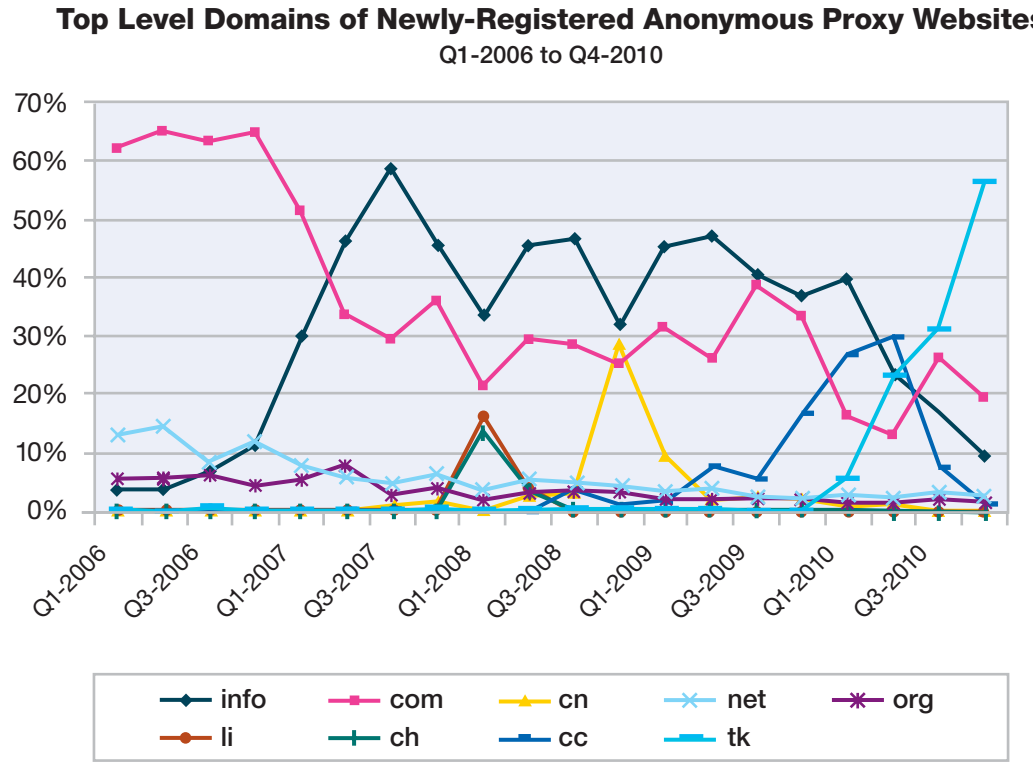


Figure 18: Top Level Domains of Newly-Registered Anonymous Proxy Websites – Q1-2006 to Q4-2010

- In the second quarter of 2010, another new star in proxy heaven, .tk (Tokelau), reached about 23 percent of new anonymous proxies. It dominated the rest of the year by acquiring nearly 30 percent in the third quarter and more than 56 percent in the fourth quarter of 2010.
- During that same time period, .info decreased dramatically and fell below 10 percent for the first time by the end of 2010.
- In the first quarter of 2010, even .com fell significantly below 20 percent for the first time, recovering to 26 percent and then 19 percent in the third and the fourth quarters of 2010.

It will be interesting to see whether .tk has a similar destiny as .co.cc—being the star of anonymous proxies for a year and a half before declining.

Country hosts of anonymous proxy websites

For anonymous proxy hosting countries, the United States has held the top position for years. More than 70 percent of all newly registered anonymous proxies were hosted in the U.S. for the years 2006-2009. In the third quarter of 2010 they fell below 70 percent for the first time in more than four years, but recovered to nearly 72 percent by the end of 2010.

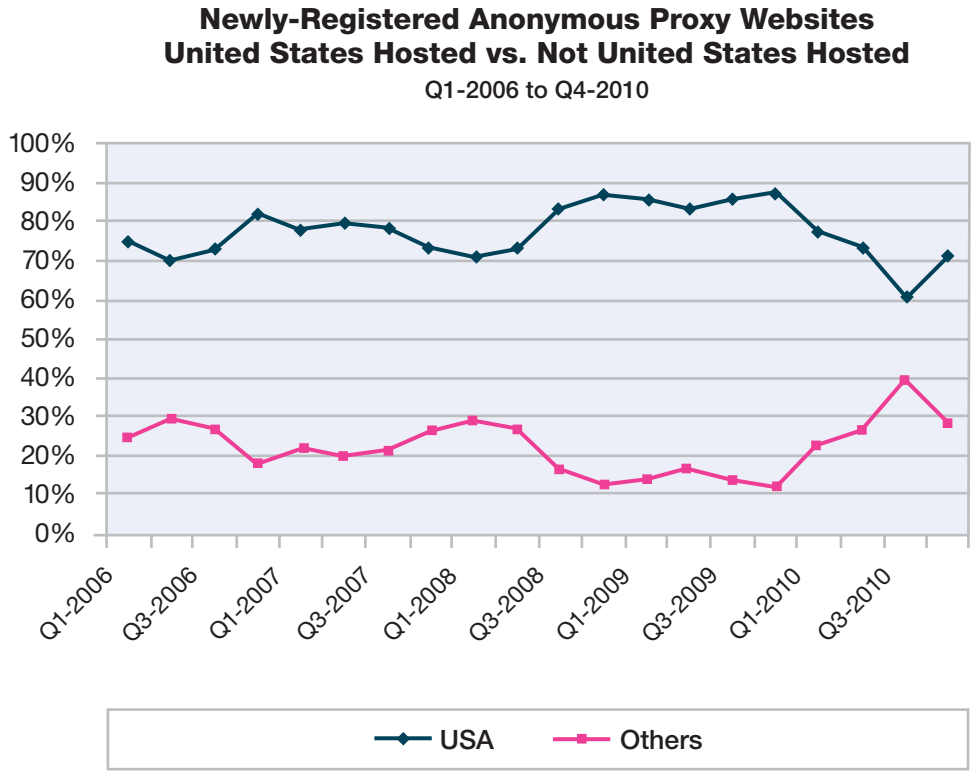


Figure 19: Newly-Registered Anonymous Proxy Websites United States Hosted vs. Not United States Hosted – Q1-2006 to Q4-2010

It is worth looking at the remaining 30 percent of all newly registered anonymous proxies in 2010. This remainder is dominated by UK (9 percent in the third quarter of 2010), Canada (6.4 percent in the third quarter of 2010), and Netherlands (5.8 percent in the third quarter of 2010). Thus, those three countries made up more than 20 percent in the third quarter of 2010. All other countries host less than 4.5 percent at the time of press in 2010.

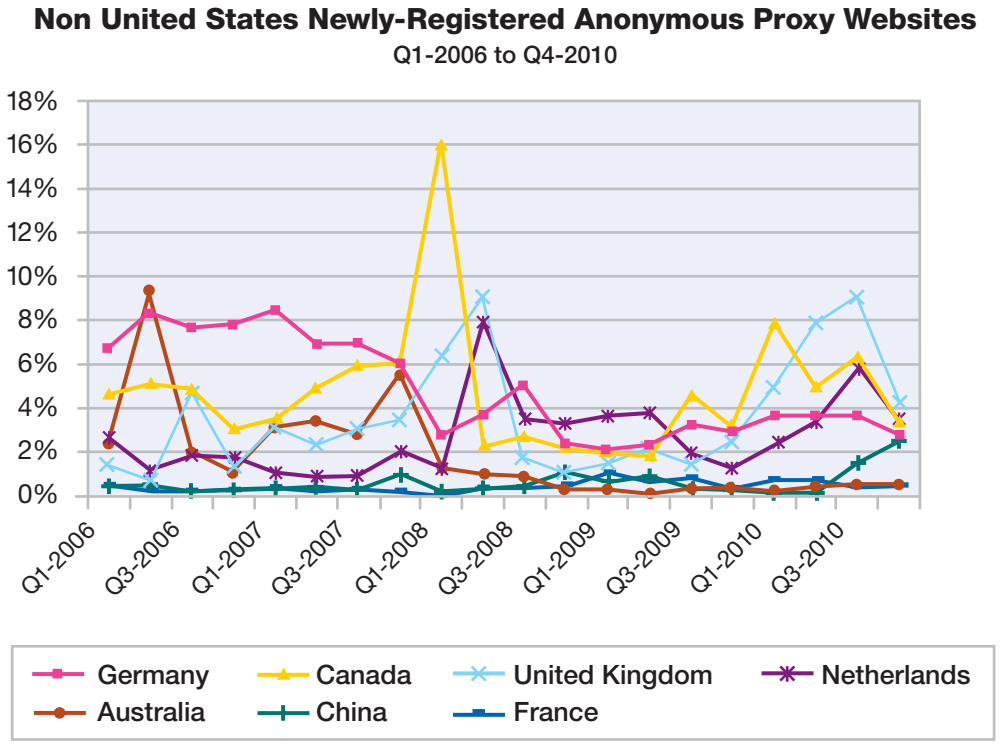


Figure 20: Non United States Newly-Registered Anonymous Proxy Websites – Q1-2006 to Q4-2010

Malicious websites

This section discusses the countries responsible for hosting the malicious links along with the types of websites that most often link back to these malicious websites. Exploits from Malicious websites discusses the web exploit toolkits involved in the majority of these malicious websites.

Geographical location of malicious web links

The United States continues to reign as the top hosting country for malicious links. More than one third of all malware links are hosted in the U.S. While China was on top two years ago, it is runner-up in 2010, hosting 8.5 percent—only 0.2 percent more than France. Romania is new within these top malicious URL hosting countries, claiming 7.9 percent (as shown in Figure 21).

The second-tier countries have also shifted, and, most significantly, many more countries seem to be jumping into the game.

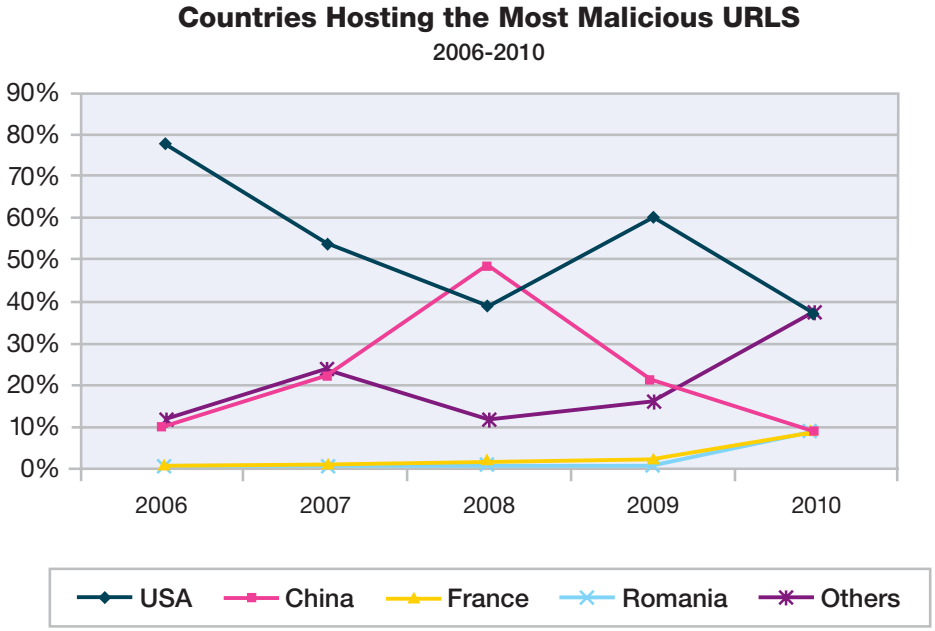


Figure 21: Countries Hosting the Most Malicious URLS – 2006-2010

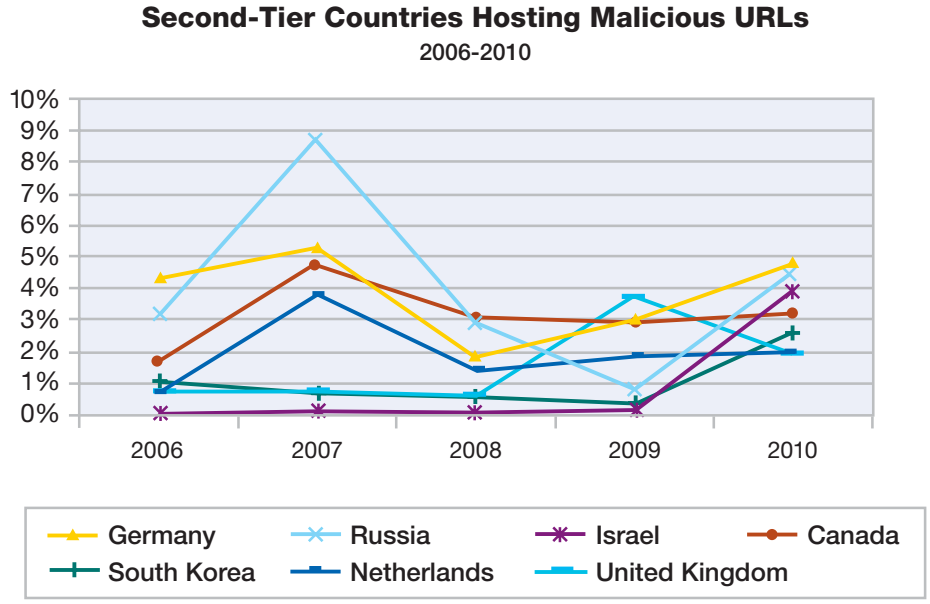


Figure 22: Second-Tier Countries Hosting Malicious URLs – 2006-2010

Good websites with bad links

As described in Web Application Vulnerabilities and Common Domains in URL Spam, attackers are focusing more and more on using the good name of trusted websites to lower the guard of end users and attempt to obfuscate their attacks with protection technologies. The use of malicious web content is no different. The following analysis provides a glimpse into the types of websites that most frequently contain links to known, malicious links.

Some of the top categories might not be surprising. For example, one might expect pornography and gambling to top the list. Indeed, together they own more than 30 percent of all malicious links. However, the second -tier candidates fall into the more "trusted" category.

Blogs, bulletin boards, search engines, personal websites, shopping sites, education, online magazines, and news sites fall into this second-tier "trusted" category. Many of these websites allow users to upload content or design their own website, such as personal content on a university's website or comments about a purchase on a shopping website. It is unlikely that these types of websites are intentionally hosting malicious links. The distribution is probably more representative of the types of websites that attackers like to frequent in hopes of finding a loop-hole (like a vulnerability or an area that allows user-supplied content) in which they can incorporate malicious links in hopes of compromising an unsuspecting victim.

The chart below lists the most common types of websites that host at least one link that points back to a known malicious website.

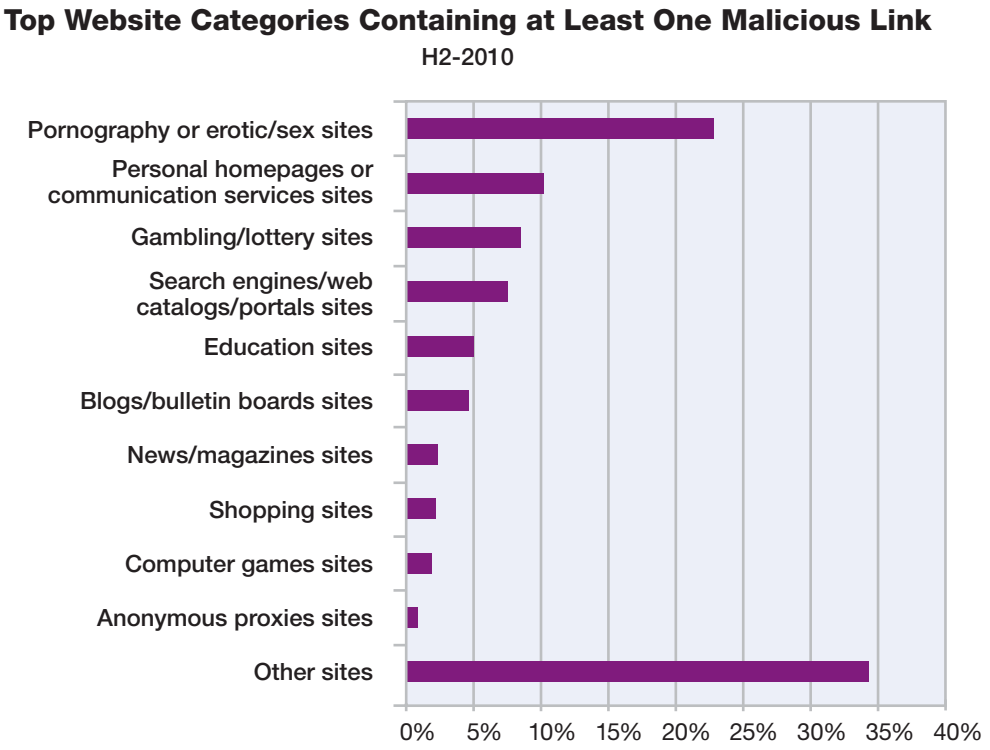


Figure 23: Top Website Categories Containing at Least One Malicious Link – H2-2010

When comparing this data with the data of the previous years, interesting trends appear. Particularly in the first half of 2010, professional "bad" websites like pornography or gambling websites have increased their links to malware, making it appear more likely that "professionals" are improving their efforts to systematically distribute their malware. However, in the second term of 2010 they declined again, but both end in a percentage above the levels of 2009.

Educational sites such as university websites have also seen increases in malware links since 2009. The same is true for Blogs and bulletin boards until mid-2010. Then they significantly decreased and fell below 5 percent for the first time in more than a year. Moreover, we noticed increases for computer games and anonymous proxy sites, but on a lesser level.

The only major category that did not decrease significantly in the second half of 2010 was gambling sites.

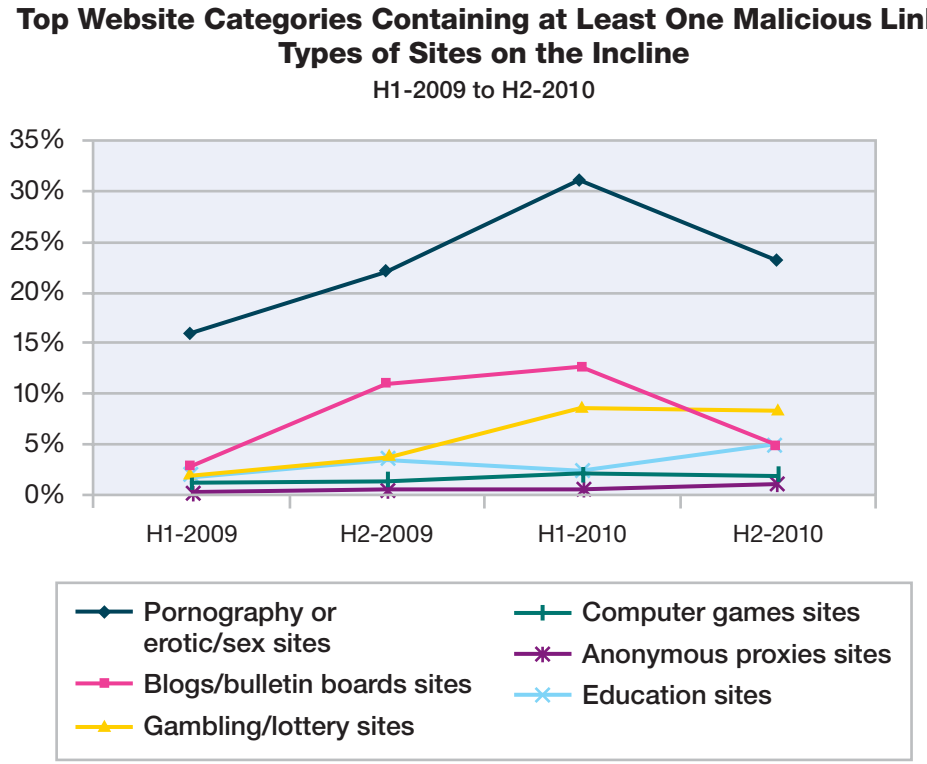


Figure 24: Top Website Categories Containing at Least One Malicious Link: Types of Sites on the Incline – H1-2009 to H2-2010

Personal homepages are no longer the most prevalent category that host at least one malicious link. Personal homepages have improved—they now host less malicious links—compared to the first half of 2009. One reason may be that personal homepages are more out of style in favor of web 2.0 applications such as profiles in social or business networks. Search engines, portals, shopping sites, and news sites have also improved or stayed on a low level. These traditional legitimate interactive sites have been used to exchange information and opinions for years. Thus, it is likely that providers of those services have increased their efforts in IT security.

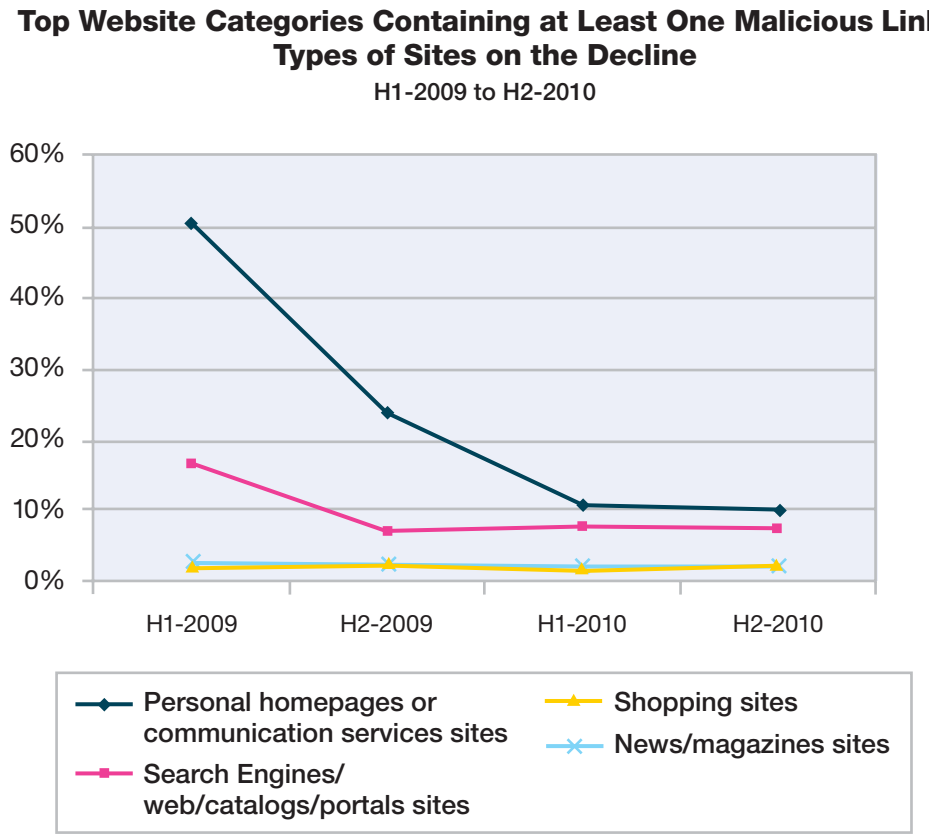


Figure 25: Top Website Categories Containing at Least One Malicious Link: Types of Sites on the Decline – H1-2009 to H2-2010

Until now we have not consider the number of malicious links placed on a website. The difference might be:

- When hosting only one or two malicious links on a site, the owner of the site might not understand or know that the link is bad – there is no ill intent.
- When placing ten or more links on a site, then this is done systematically and intentionally to get visitors clicking on bad links. The goal of the owner might be to enjoy a financial advantage from the compromises.

Out of the categories of websites that host 10 or more of these links, pornography accounts for nearly 30 percent and gambling accounts for nearly 29 percent.

Compared to the data six months ago, the values in most categories have stayed flat or slightly decreased but gambling increased by nearly one percent. Against the background of 0.6 percent of the adult population having problem gambling issues (see **http://en.wikipedia.org/wiki/Gambling_addiction#Prevalence**), gambling sites are a popular target for malware distributors. Note also that Personal Homepages and Communication Services increased by 1.7 percent and Educational sites increased by 0.6 percent.

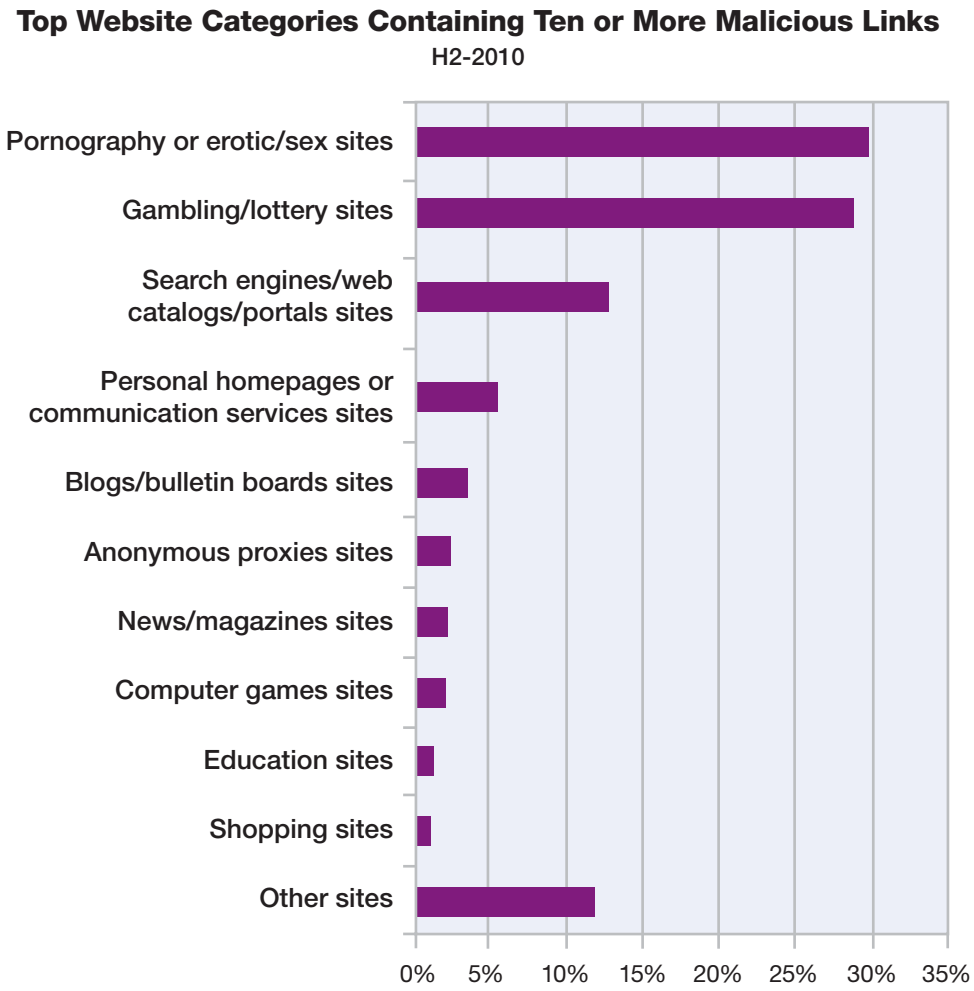


Figure 26: Top Website Categories Containing Ten or More Malicious Links – H2-2010

Spammers focus on content rather than volume

The IBM spam and URL filter database provides a world-encompassing view of spam and phishing attacks. With millions of email addresses being actively monitored, the content team has identified numerous advances in the spam and phishing technologies attackers use.

Currently, the spam filter database contains more than 40 million relevant spam signatures. Each piece of spam is broken into several logical parts (sentences, paragraphs, etc.). A unique 128-bit signature is computed for each part and for millions of spam URLs. Each day there are approximately one million new, updated, or deleted signatures for the spam filter database.

This section addresses the following topics:

- Major content trends in spam 2010
- Most popular domains and top level domains used in spam
- Spam country[9] of origin trends, including spam web pages (URLs)
- Most popular subject lines of spam

9 The statistics in this report for spam, phishing, and URLs use the IP-to-Country Database provided by WebHosting.Info (http://www.webhosting.info), available from http://ip-to-country.webhosting.info. The geographical distribution was determined by requesting the IP addresses of the hosts (in the case of the content distribution) or of the sending mail server (in the case of spam and phishing) to the IP-to-Country Database.

Major content trends in spam for 2010

After the last major threats of image-based and PDF spam in 2007, we did not see major changes in the content of the spams in 2008 and 2009, apart from another short-period threat of image spams in the first term of 2009. One characteristic for the low changes in technical spam content was the constant level of HTML-based spam (in most cases a bit more than 80 percent) and plain-text spam (mostly 10-15 percent).

In 2010 there were major changes in the technical content of spam. To see these trends at a glance, see Figure 27.

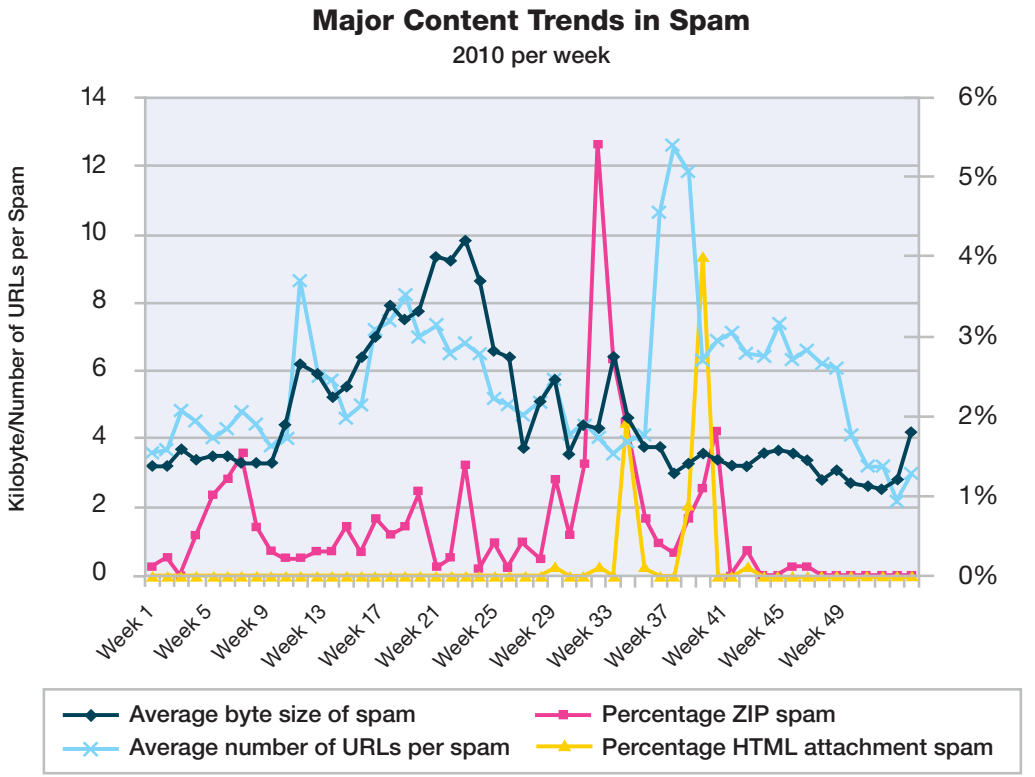


Figure 27: Major Content Trends in Spam – 2010 per week

Let's have a closer look at the trends and the characteristics:

- **March-August:** Random text spam combined with random URLs, significantly increased the average byte size of spam. In previous years the average byte size of spam was directly dependent on the percentage of image-based spam. But in 2010 the percentage of image spam was flat and below two percent (in most cases below one percent). When looking at these larger spams one can see large text fragments randomly chosen from the Internet, complemented by random URLs (syntactically correct URLs build from random characters or words, but many of them do not exist in the Internet). Random text is a very old technique of the spammers to make spam look more legitimate. However, recent anti-spam techniques do not have any problems with it. So why did spammers re-activate this old approach? Maybe they hoped that those masses of text would confuse Bayesian classifiers, particularly self-trained Bayesian classifiers, which are used in a non-business context; hence, these spam attacks might be targeted to these non-business users.
- **August—Spam with malicious ZIP attachments:** At the beginning of August, spammers began sending spam threats with ZIP attachments. We looked into these messages, and each ZIP file contained a single EXE file that was malicious. Spammers used different kinds of malware, e.g. variants of the Zeus Trojan or a copy of the Bredolab downloader (see sidebars). More details on these spam threats with ZIP attachments can be found at **http://blogs.iss.net/archive/ZIPMalwareSpam.html**. IBM Proventia customers can use the Email_Zip_Executable_Content signature to detect threats like these. The spammers used typical methods to attract the user's attention by using subjects such as:
  - Your Flight Ticket
  - Financial Summary
  - Statement Notification
  - Financials
  - FW: Car & Car loan
  - Employee Orientation report

Zeus Trojan

Zeus is a very common Trojan that's generated with a kit that anyone can purchase online. There are many different individuals and groups that have Zeus botnets set up. There are a lot of ways it gets spread, but the operators of this particular botnet are growing it by sending out emails with ZIP file attachments. The goal of Zeus botnets is usually to steal personal information, and the type of information stolen is commonly online banking data that criminals can use to access bank accounts to transfer money. For more information about the Zeus botnet see Trojan Bot networks in the section **"IBM Managed Security Services—A global threat landscape".**

- **September—Spam with HTML attachments:** There are some similarities between the ZIP attachment spam emails of the month before and the HTML attachment spam. In both cases the user's computer gets infected when clicking on the attachment. Furthermore, in the HTML attachment spam, the user's attention is attracted in the same way as in the ZIP attachment spam by something in the email text body such as:
  - Please see attached invoice for Stockton floor project
  - More details are in the attached invitation
  - See attached for breakdown—the $40 HOA will not be included in payment do deduct from total which = $1095/mo
  - Please print out the invoice copy attached and collect the package at our office
  - Attached is a copy of the deposit received for your records
  - Here are the signed documents
  - memo on image secrecy (attached)
  - Enclosed is my CV for your consideration
  - You will find the resume attached to this email
  - Attached you will find the fall daily tour schedule for your review

- **September-November—Random URL spam:** During this time period, spammers did not use random text but instead used random URLs extensively. This resulted in more than 12 (syntactically correct but otherwise useless and random) URLs per spam on an average during the beginning weeks of this time period. In the following weeks, we recognized more than six URLs per spam, which is still above the normal levels of 2-4 URLs per spam on an average.

- **December—Increased average byte size of spam again:** This time, this is a result of the drop of the spam volume by 70 percent (**see section "Spammers on holiday at the end of the year" page 46**), that affected particularly small spam.

Against the trends of previous years—wherein spammers made very few changes in the technical content of spam throughout the year, in 2010, spammers made a continuous effort to change the technical contents regularly. In the next section, we will discuss an associated factor—the volume of spam.

Bredolab downloader

This Trojan downloads a rogue antivirus program called SecurityTool that pretends to find viruses on your PC when none exist.

Spam volume

While we recognized significant increases of the spam volume year over year until 2009, in 2010 there were a few months with ups and downs in the volume of spam seen over the year. However, the overall trends stayed flat, and we saw less volume at the end of the year in comparison to the beginning of 2010.

Conclusions about spam volume and content

Why are spammers making an effort to change the technical content of spam more often than in previous years but are no longer focusing on increasing the overall volume of spam? Here we ponder a few presumptions about these possible trends. Some trends might be more plausible than others.

- Perhaps in recent years there was a linear connection between the number of spam messages and the profit reached by spam. Is this connection lost?
- Is the spam market saturated? Will we even see a decrease of spam volume in the upcoming years?
- Since there is only one single point to combat current spam - when receiving them - for the companies (or the users), did the bad guys change their focus to other - more distributed - areas that are more complicated to take countermeasures? This assumption is strengthened by:
  - the increase of Botnet Trojan Activity in 2010 - see section **"Trojan Bot networks"**
  - the surge of Obfuscation Activity in 2010 - see section **"Obfuscation"**
  - the growth of Backscatter Activity in 2010 - see section **"Spoofed Denial of Service attacks"**
  - the rise of the Vulnerability Disclosures in 2010 - see section **"2010 - A record setting year"**
- Is the increase of spam messages only achieved within internal social network messaging systems and other Web 2.0 applications?
- Are spammers cautious in efforts with increasing the levels too much because the more similar spam messages they produce, the easier they can be detected and blocked by perfected spam filters? That would mean, they assume that they have reached an optimum concerning the spam volume.
- Are the new operating systems more secure and prevent a further increase of the levels?
- Do even "spamming companies" suffer from the war for talent, hence, they have recruitment problems?

It is very unlikely that the spam business has become unprofitable. One scenario could be that spam volume stays flat but the kinds of spam change more frequently to circumvent spam filters with new types of spam that are more difficult to detect.

Maybe there will be more experiments with other attachment types? We tallied the most popular file types, and there is one file type becoming more and more popular – Open Office documents. When do spammers use those attachments?

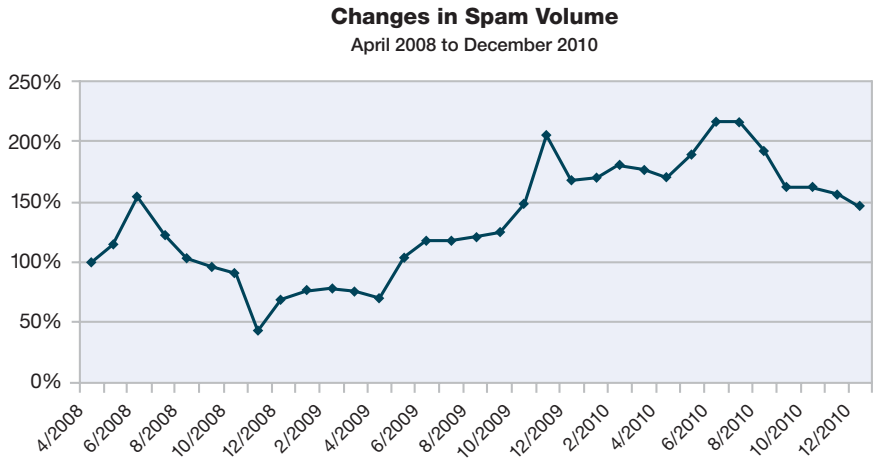


Figure 28: Changes in Spam Volume – April 2008 to December 2010

Spammers on holiday at the end of the year

One week before year's end, spammers surprised us by sending out 70 percent less spam than the weeks before; this period lasted about two and a half weeks. After the Christmas holiday season, spam levels returned to the same level as before Christmas.

When looking at the reductions of the spam volume per country there were some countries, such the U.S., Canada, and UK, with a decline of more than 90 percent. More about the declines per country and some more details can be found on **http://blogs.iss.net/archive/2011spambotdecline.html**.

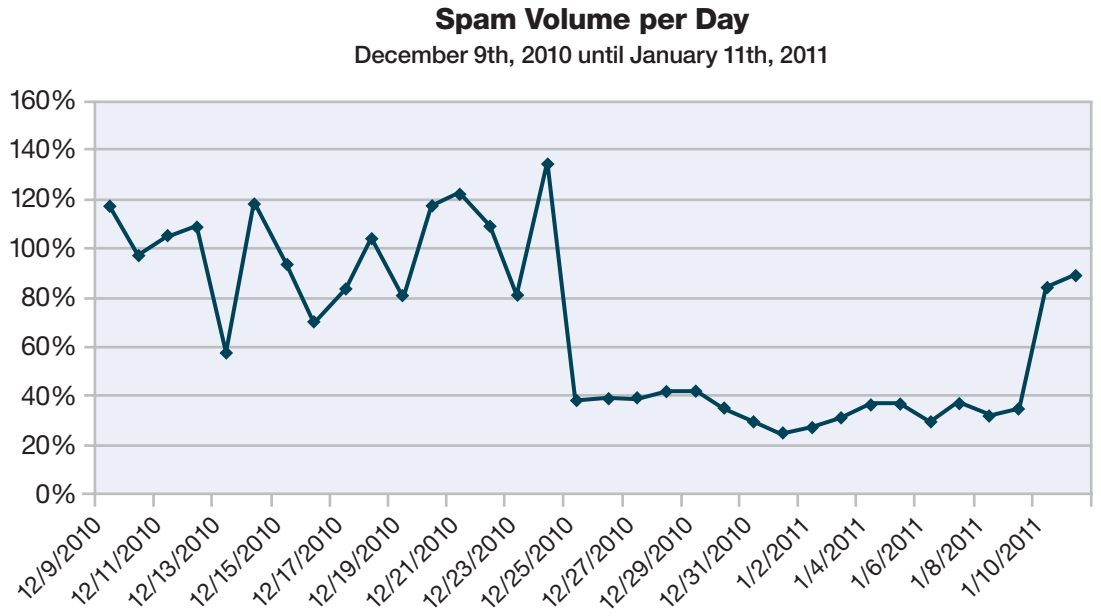


Figure 29: Spam Volume per Day – December 9th, 2010 until January 11th, 2011

Regional spam volume per day of the week

Another approach for looking at the spam volume is checking the spam volume per day of the week. If we received equal volumes each day, then we would receive 14.3 percent of the weekly spam volume per day. When looking at spam written in English, French, or Spanish, this appears more or less the case.

English spam is distributed very consistently over the week days. The days with the least amount of spam are Wednesday (14.0 percent) and Sunday (13.7 percent); the days of the week with the most spam are Tuesday (14.7 percent) and Friday (14.8 percent). The most French spam is received on Thursday (15.7 percent) and Friday (15.8 percent). The greatest spam day of the week for Spanish spam is Monday, when they process 18 percent of the weekly amount of their spam. However, the difference between the other week day amounts for French and Spanish is rather low.

The situation is different for Russian and Portuguese spam. On weekends, we receive much less spam written in these two languages. Almost 90 percent of spam in the Russian language is sent out on week days; on Saturday and Sunday, they only send out about five percent each day. Their strongest days are Tuesday, Wednesday, and Thursday, when they process about 20 percent of their weekly amount each day. The patterns are similar for Portuguese spam. Their strongest days are Tuesday to Thursday, and their weakest days are Saturday and Sunday.

Assuming that spammers prefer not to work weekends, it appears that spam in the English, French, and Spanish languages is sent out completely automatically, retaining its typical volume on weekends. Contrarily, Russian and Portuguese spam requires more manual work, resulting in a significant drop at the weekends.

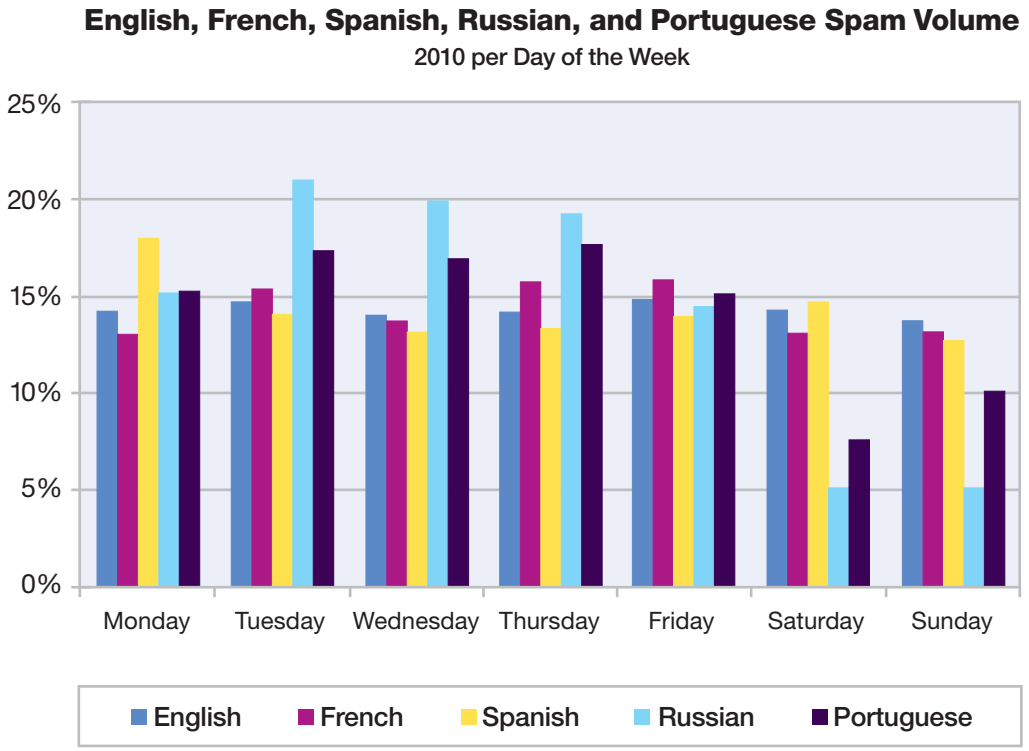


Figure 30: English, French, Spanish, Russian, and Portuguese Spam Volume – 2010 per Day of the Week

Common domains in URL spam

The vast majority of spam, more than 90 percent, is still classified as URL spam—spam messages that include URLs that a person clicks to view the spam contents. It is worthwhile to take a closer look at the most frequently used domain names in URL spam. The table on the following page shows the top 10 domains per month throughout 2010, with some key domains highlighted.

The majority of those domain names are well-known and trusted (highlighted in color in the table on page 49). Not only do these legitimate websites provide a recognizable (and trustworthy) web link to the end user, but spam messages using them may also successfully evade some anti-spam technology because they only use legitimate links in their spam emails. There are different types of well-known domains:

- **Internet service providers (blue):** Used by spammers in recent years to make look their spams appear trustworthy.
- **Image-hosting websites (green):** Also used by spammers for several years. Spammers like to vary between well known image-hosters like flickr.com and imageshack.us and many other small and medium-sized image-hosting websites.
- **Random word domains (orange):** From July to September 2010 spammers used random words to "build" URLs. This was done in such a massive way that the very common words "the", "of", "and", "in", "a" even made it to the top ten with the ".com" extension. Since then, we have seen random domains built from random characters and now it appears we see random domains built from random words.
- **Official websites of Pfizer and Rolex (yellow):** From September 2010 on, spammers used the official websites of Pfizer (pfizer.com, pfizerhelpfulanswers.com, viagra.com) and Rolex (rolex.com). Obviously, spammers include in their strategies that most spam filters do not use simple keyword search anymore and even assume that URLs from pfizer.com or rolex.com make their messages looking more legitimate.
- **URL shortening services (purple):** From September 2010 on, some of these services made it to the top 10.

The table of domains on the next page became more multicolored in the second half of 2010. That means that spammers used multiple methods to present their offers via URLs. This is another illustration of the move of spammers from volume to "content quality," as mentioned above.

Rank	January 2010	February 2010	March 2010	April 2010	May 2010	June 2010
1.	flickr.com	radikal.ru	livefilestore.com	livefilestore.com	imageshack.us	imageshack.us
2.	imageshack.us	imageshack.us	imageboo.com	imageshack.us	imageshost.ru	imageshost.ru
3.	radikal.ru	livefilestore.com	radikal.ru	imageshost.ru	myimg.de	pikucha.ru
4.	livefilestore.com	flickr.com	imageshack.us	imgur.com	xs.to	imgur.com
5.	webmd.com	live.com	googlegroups.com	myimg.de	imgur.com	mytasvir.com
6.	picsochka.ru	imageboo.com	live.com	xs.to	tinypic.com	mojoimage.com
7.	live.com	capalola.biz	akamaitech.net	icontact.com	livefilestore.com	myimg.de
8.	superbshore.com	feetorder.ru	gonestory.com	tinypic.com	icontact.com	twimg.com
9.	tumblr.com	laughexcite.ru	bestanswer.ru	live.com	googlegroups.com	icontact.com
10.	fairgreat.com	hismouth.ru	wrotelike.ru	binkyou.net	images-amazon.com	twitter.com

Rank	July 2010	August 2010	September 2010	October 2010	November 2010	December 2010
1.	imageshack.us	yahoo.com	the.com	businessinsider.com	rolex.com	pfizer.com
2.	icontact.com	the.com	of.com	migre.me	msn.com	viagra.com
3.	the.com	icontact.com	msn.com	4freeimagehost.com	bit.ly	msn.com
4.	myimg.de	feetspicy.com	pfizerhelpfulanswers.com	bit.ly	pfizer.com	rolex.com
5.	of.com	of.com	and.com	postimage.org	co.cc	bit.ly
6.	imgur.com	ratherwent.com	bit.ly	imgur.com	royalfoote.com	product45h.com
7.	by.ru	and.com	in.com	pfizer.com	royalbelie.com	newpfizermed5k.com
8.	and.com	facebook.com	yahoo.com	viagra.com	royalreleasable.com	xmages.net
9.	in.com	in.com	a.com	uploadgeek.com	luxurystorewatch.com	cordfork.com
10.	tastymighty.com	a.com	x-misc.com	vipplayerq.com	basincook.com	onlinepfizersoft2.com

Table 4: Most common domains in URL spam, 2010

It was the trend of recent years to use well-known and trusted domains in spam. In the second half of 2010 this trend stopped increasing for the first time in more than two years but stayed at a high level. The following chart shows the percentage of trusted domains versus spam domains within the monthly top 10 domains of the last three years. Not until the second half of 2010 was there no further increase of the usage of trusted domains in spam. At this point, the percentage slightly decreased to 77 percent.

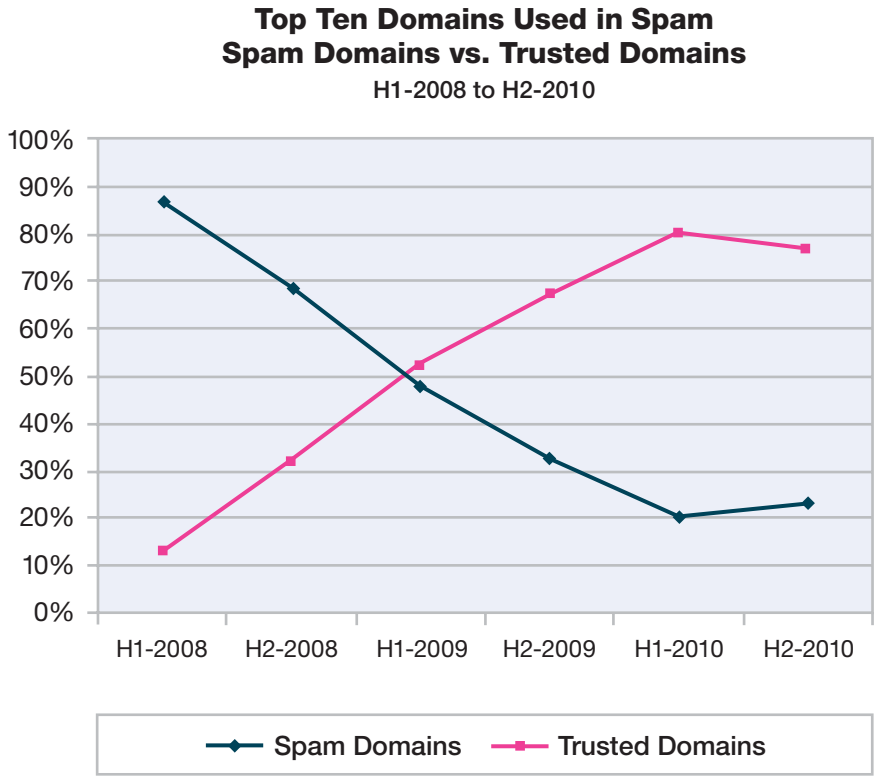


Figure 31: Top Ten Domains Used in Spam: Spam Domains vs. Trusted Domains – H1-2008 to H2-2010

Common top-level domains in URL spam

Table 5 shows the five most frequently used Top Level Domains used in spam by month. In this table we only consider URLs that really host spam content.

2010 was completely dominated by .ru spam URLs. In January .ru reached rank 4, and in nearly all months that followed .ru won the race (only in April it was runner-up). In December 2010, there was an interesting newcomer to the top 5; .ec, the top level domain of Ecuador, entered this table for the first time. This entrance was caused by the massive abuse of the URL shortening service redir.ec, another manifestation of the intensified usage of these services.

Perhaps the most surprising question is: What happened to China (.cn)? After ranking 2 in January, its rank decreased from month to month. Since May 2010, China no longer belongs to the most common top level domains used in spam. In the **IBM X-Force 2010 Mid-Year Trend and Risk Report** in section **"Spammers' domains move from .cn to .ru"** there is detailed information about this change and its reasons.

Internationalized country code top-level domains: First occurrences in spam

When looking at the midfield of the top level domains used in URL spam in November and December, we recognized the first occurrences of internationalized country code TLDs. This TLD reached rank 46 in November and rank 28 in December." The spam that used these URLs was rather unspectacular, just normal Russian language spam."

Internationalized country code top-level domains

Since the beginning of 2010 it is possible to register internationalized country code top-level domains. Therefore URLs can be displayed without using any ASCII letters. The first domains were registered in the Arabic and Cyrillic alphabet. More details on internationalized domains can be found on

http://en.wikipedia.org/wiki/Internationalized_country_code_top-level_domain

http://en.wikipedia.org/wiki/List_of_Internet_top-level_domains

http://en.wikipedia.org/wiki/Internationalized_domain_name

Top 5 TLDs used to host Spam Content

Rank	January 2010	February 2010	March 2010	April 2010	May 2010	June 2010
1.	com	ru (Russia)	ru (Russia)	com	ru (Russia)	ru (Russia)
2.	cn (China)	com	com	ru (Russia)	com	com
3.	net	net	net	net	de (Germany)	de (Germany)
4.	ru (Russia)	cn (China)	cn (China)	de (Germany)	net	net
5.	info	info	biz	cn (China)	org	org

Rank	July 2010	August 2010	September 2010	October 2010	November 2010	December 2010
1.	ru (Russia)	ru (Russia)	ru (Russia)	ru (Russia)	ru (Russia)	ru (Russia)
2.	com	com	com	com	com	com
3.	de (Germany)	net	net	net	net	ec (Ecuador)
4.	net	de (Germany)	info	in (India)	in (India)	info
5.	org	fr (France)	in (India)	de (Germany)	tk (Tokelau)	in (India)

Table 5: Most common top level domains with real spam content, 2010

10 'рф' are the letters rf in the Cyrillic language and mean 'Russian Federation'.

Spam—country of origin

The following map shows the origination point[11] for spam globally in 2010. As in the previous year, the U.S., India, Brazil, and Vietnam were the top four spam-sending countries, accounting for nearly one third of worldwide spam. However, the countries changed their positions, and the U.S. re-conquered the top position for the first time since 2007. UK, Germany, Ukraine, and Romania are newcomers to the top 10 while Poland, Turkey, China, and Colombia left the top ten spam senders in 2010 compared with 2009.

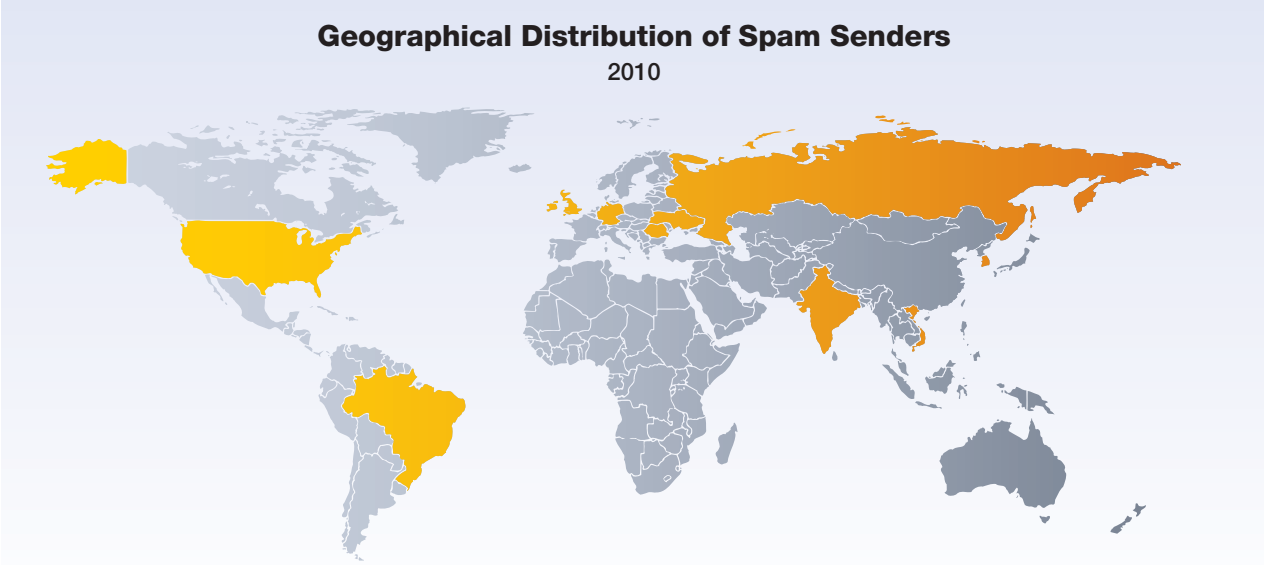


Figure 32: Geographical Distribution of Spam Senders – 2010

Country	% of Spam
USA	10.9%
India	8.2%
Brazil	8.1%
Vietnam	5.4%
Russia	5.2%

Country	% of Spam
United Kingdom	4.4%
Germany	3.7%
South Korea	3.3%
Ukraine	3.0%
Romania	2.9%

Table 6: Geographical Distribution of Spam Senders – 2010

11 The country of origin indicates the location of the server that sent the spam email. X-Force believes that most spam email is sent by bot networks. Since bots can be controlled from anywhere, the nationality of the actual attackers behind a spam email may not be the same as the country from which the spam originated.

When looking at shorter time frames and including the previous year, some more trends become visible, particularly the decrease of Brazil in comparison to 2009 and the continued incline of India from spring 2009 to autumn 2010.

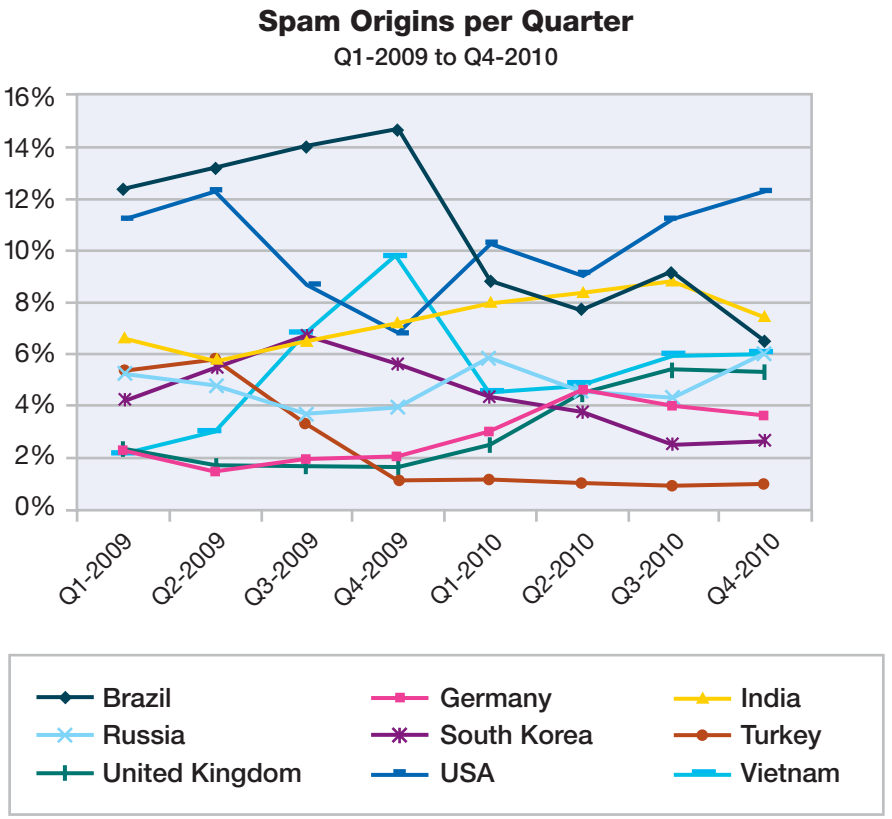


Figure 33: Spam Origins per Quarter – Q1-2009 to Q4-2010

Spam—country of origin trends

When looking at the last five years some long-term trends become visible:

- India is the only country having a continuous growth
- After two years of significant increases, Brazil and Vietnam declined for the first time
- After two years as runner-up the United States recaptured the top position in 2010
- Spain and France lost their dominating role beginning in 2007
- Russia lost its dominating role beginning in 2009
- South Korea fell below four percent for the first time

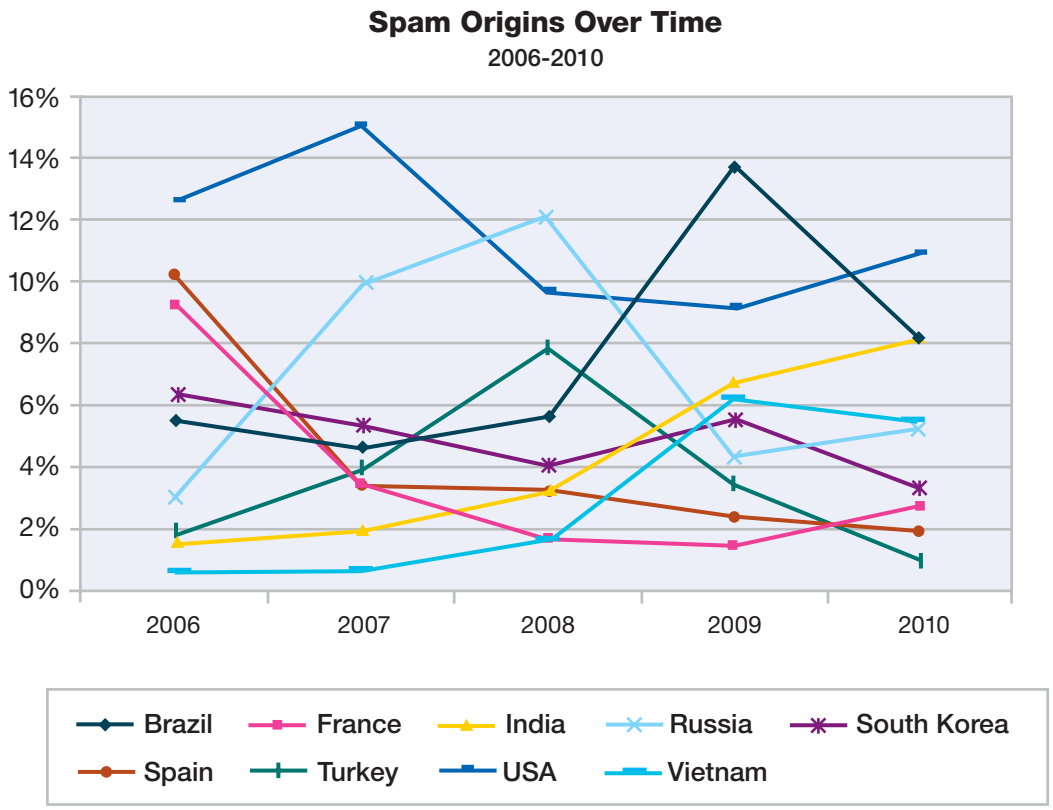


Figure 34: Spam Origins Over Time – 2006-2010

Spam URLs—country of origin trends

From 2007 until end of 2009, spam URLs hosted on servers in China dramatically increased. All other countries have stagnated or declined, particularly the United States. In 2010, the trend towards China has slowed, and China actually declined for the first time in the last two years. China still holds the number one position, hosting more than 30 percent of all spam URLs. Some other countries increased, particularly the U.S., now hosting nearly 27 percent of all spam URLs and South Korea, hosting more than 8 percent of all spam URLs. A newcomer to the top ten is Moldova, which hosts 5.4 percent of all spam URLs.

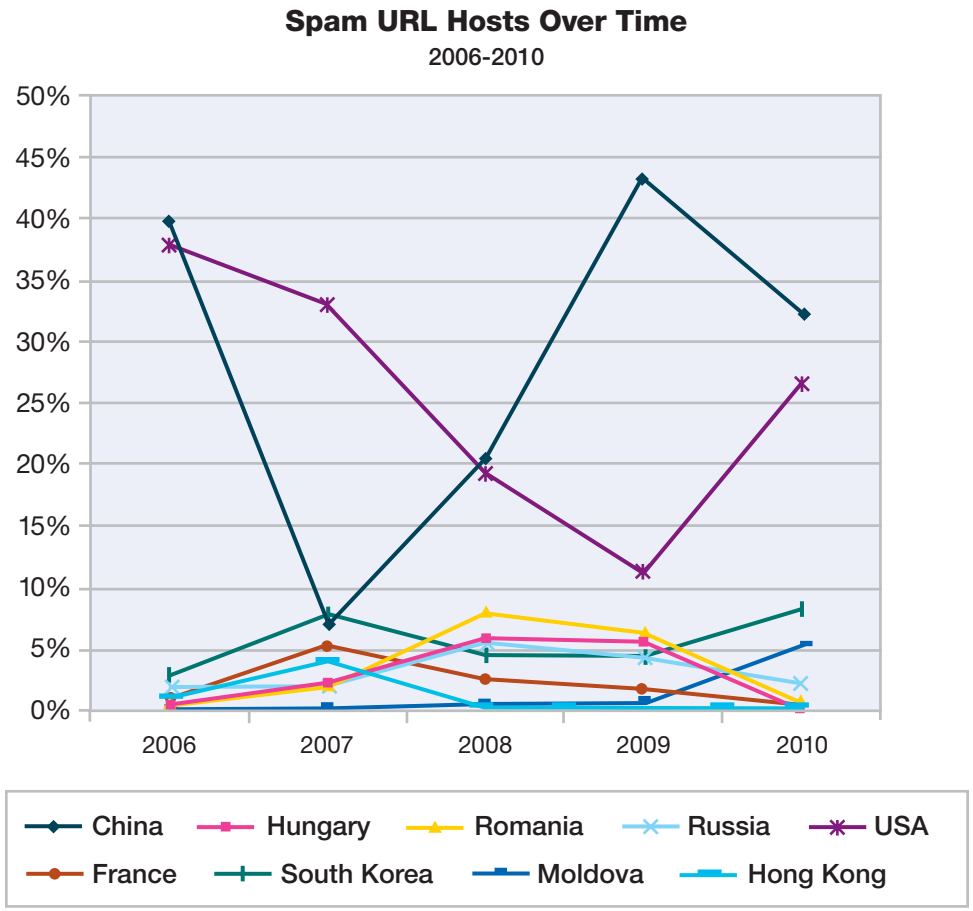


Figure 35: Spam URL Hosts Over Time – 2006-2010

The top ten subject lines in 2010 made up about 2.4 percent of all spam subject lines; this is less than 2009 (2.6 percent), 2008 (3 percent), and significantly down from the 20 percent figure recorded in 2007.

While the subjects on rank 1, 2, 3, and 8 are dating related (marked in orange in the following table), there are also subjects related to Web 2.0 and social networks (rank 5, 9, and 10, marked green). As expected, the "classical" topics about replica watches or medical products are still visible (rank 4, 6, and 7, marked in yellow). Particularly medical products of Pfizer enjoy great popularity when mentioned in spam subjects. Here spammers do it in their traditional way and play with upper and lower case, replace "o" by "0" (zero), use different percent numbers and so on. Obviously 70 and 80 percent seem to be their favorite percentage rates, as these two are the only ones which reached the top 10.

The following table shows the most popular spam subject lines in 2010:

Subject Line	%
Inna (status-online) invites you for chat.	0.45%
You have got new messages(dating)	0.40%
Marina 21y.o, I am on-line now, let's chat?	0.26%
Pfizer -80% now!	0.25%
You have a new personal message	0.22%
Replica Watches	0.20%
RE: SALE 70% 0FF on Pfizer	0.18%
I am on-line now, let's chat?	0.16%
News on myspace	0.15%
Please read	0.13%

Table 7: most popular spam subject lines 2010

Phishing

This section covers the following topics:

- Phishing as a percentage of spam
- Phishing country of origin trends, including phishing web pages (URLs)
- Most popular subject lines and targets of phishing
- Phishing targets (by industry and by geography)

Phishing volume

In 2010, Phishing emails slowed and the complete year volume did not reach the levels at the end of 2009. In 2010, after a drop in January and February we saw an increase in the phishing volume in March and April. In May there was another drop. This might be in relation to the apprehension of a Romanian phishing gang at the beginning of May (see **http://www.h-online.com/security/news/item/Police-apprehend-Romanian-phishing-gang-997151.html**). In June, the levels of March and April were reached again, but still far away from the volumes of summer of 2009. Phishing slowed down in the following months with a very slight increase in October and November.

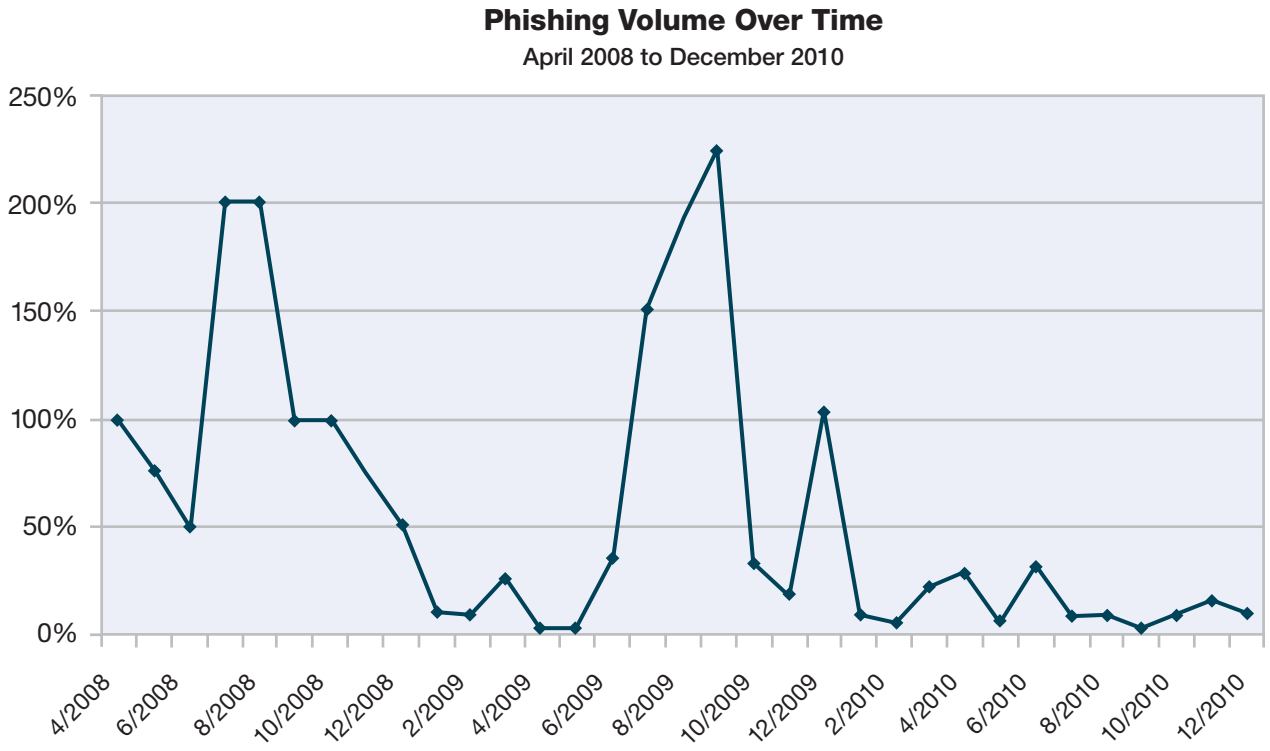


Figure 36: Phishing Volume Over Time – April 2008 to December 2010

Are phishers becoming skimmers?

When comparing the phishing email volume by quarter, we saw significant increases of phishing emails in summer and fall of 2008 and 2009.[12] In 2010, this seasonal phishing surge did not occur (see bars of Q3 in Figure 37).

Another lucrative phishing approach in the area of banks is ATM skimming. This could be an obvious resumption of the former email phishing "business" because:

- Most people are unfamiliar with ATM skimming
- ATM skimming occurred five times more in 2010 than in 2009 (see **http://www.cuna.org/newsnow/10/system121510-7.html**)—maybe even more unfamiliar than they are with spam and phishing emails.

ATM skimming occurred five times more in 2010 than in 2009 (see **http://www.cuna.org/newsnow/10/system121510-7.html** again). However, phishers do use other approaches, see the sidebar **"Zeus Trojan"** on page 43 for example.

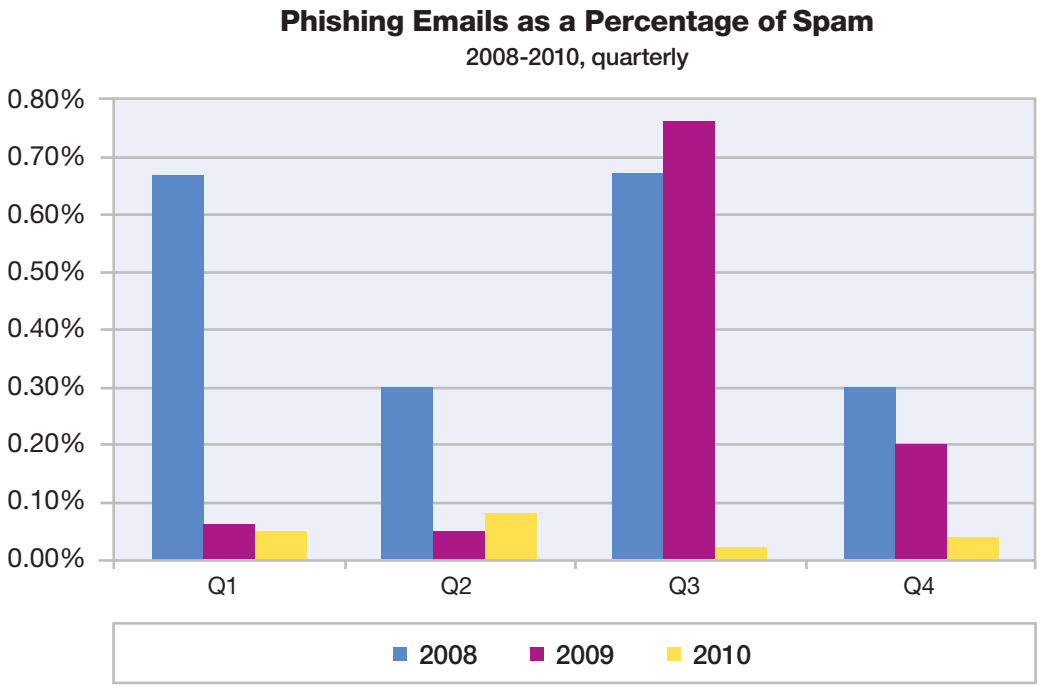


Figure 37: Phishing Emails as a Percentage of Spam – 2008-2010, quarterly

ATM skimming

ATM skimmers put a device over the card slot of an ATM that reads the magnetic strip when the unsuspecting users pass their card through it. More information about this topic can be found on **http://en.wikipedia.org/wiki/Credit_card_fraud#Skimming**.

12 The country of origin indicates the location of the server that sent the phishing email. X-Force believes that most phishing email is sent by bot networks. Since bots can be controlled from anywhere, the nationality of the actual attackers behind a phishing email may not be the same as the country from which the phishing email originated.

Phishing—country of origin

The top country of origin of phishing emails is now originating from India and the runner-up is Russia. The top phishing email country of origin of 2009, Brazil, reached rank three during 2010. Position four is owned by USA. Hence, the members of the top four are still the same as in 2009, only their positions have changed.

Newcomers in the top 10 are Ukraine, Taiwan, and Vietnam, while Argentina, Turkey, and Chile disappeared from this list.

The following map highlights the major countries of origin for phishing emails in 2010.

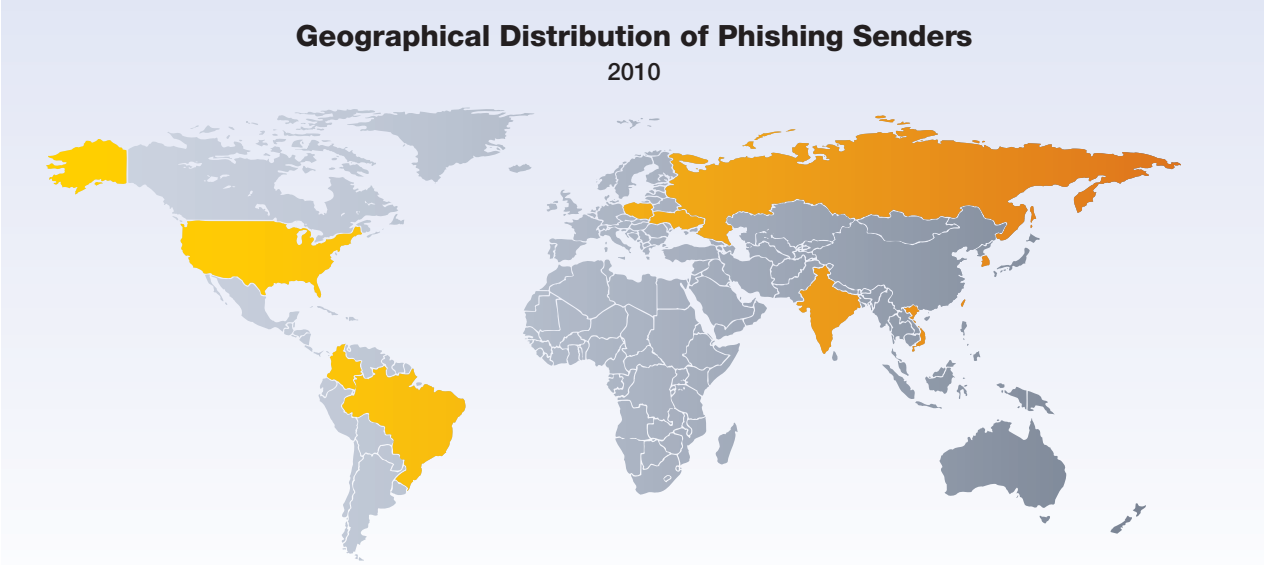


Figure 38: Geographical Distribution of Phishing Senders – 2010

Country	% of Phishing	Country	% of Phishing
India	15.5%	South Korea	4.7%
Russia	10.4%	Colombia	3.0%
Brazil	7.6%	Taiwan	2.2%
USA	7.5%	Vietnam	2.2%
Ukraine	6.3%	Poland	1.8%

Table 8: Geographical Distribution of Phishing Senders – 2010

Phishing—country of origin trends

Many of the leading phishing senders of 2006, 2007, and 2008, have declined significantly in 2009 and 2010. In particular, Spain and Italy have lost their position, but South Korea is still ranked six in 2010.

The new leading phishing senders are now originating from India, Russia, Brazil, with India holding the top position.

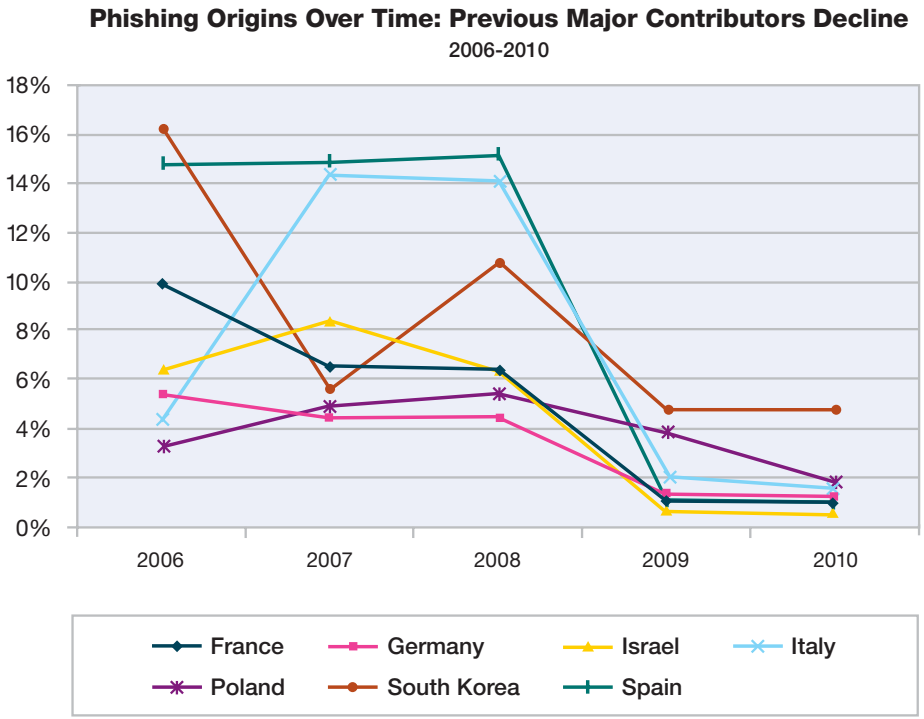


Figure 39: Phishing Origins Over Time: Previous Major Contributors Decline – 2006-2010

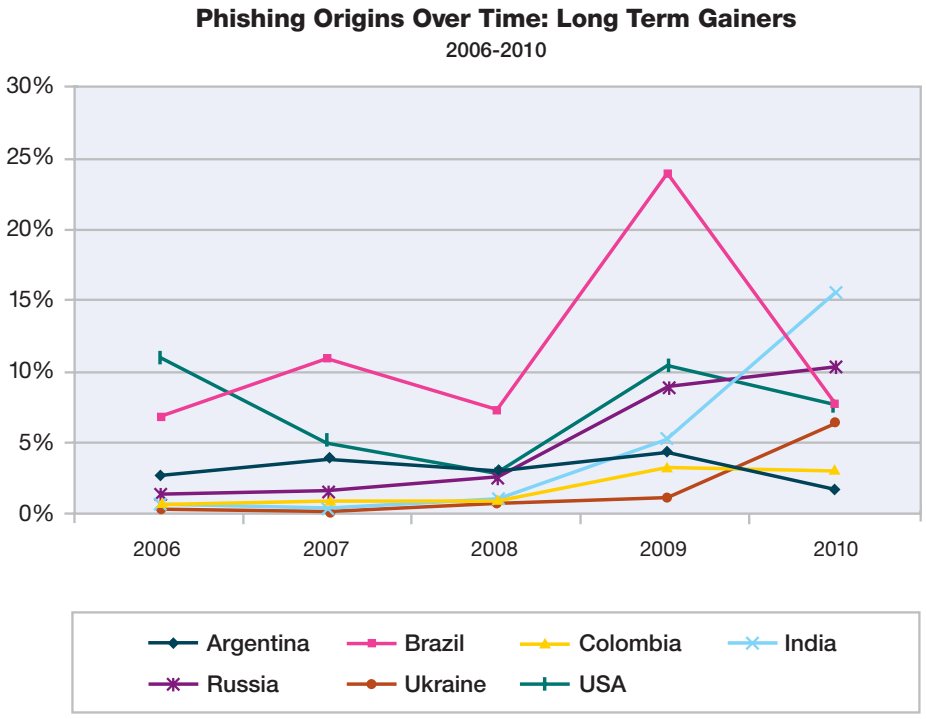


Figure 40: Phishing Origins Over Time: Long Term Gainers – 2006-2010

Phishing URLs—country of origin

The following map shows where the phishing URLs are hosted. The top ten countries have not changed in comparison to 2009, and even their place has changed only a little. Russia fell from rank eight to 10, while Spain and Poland each gained one rank.

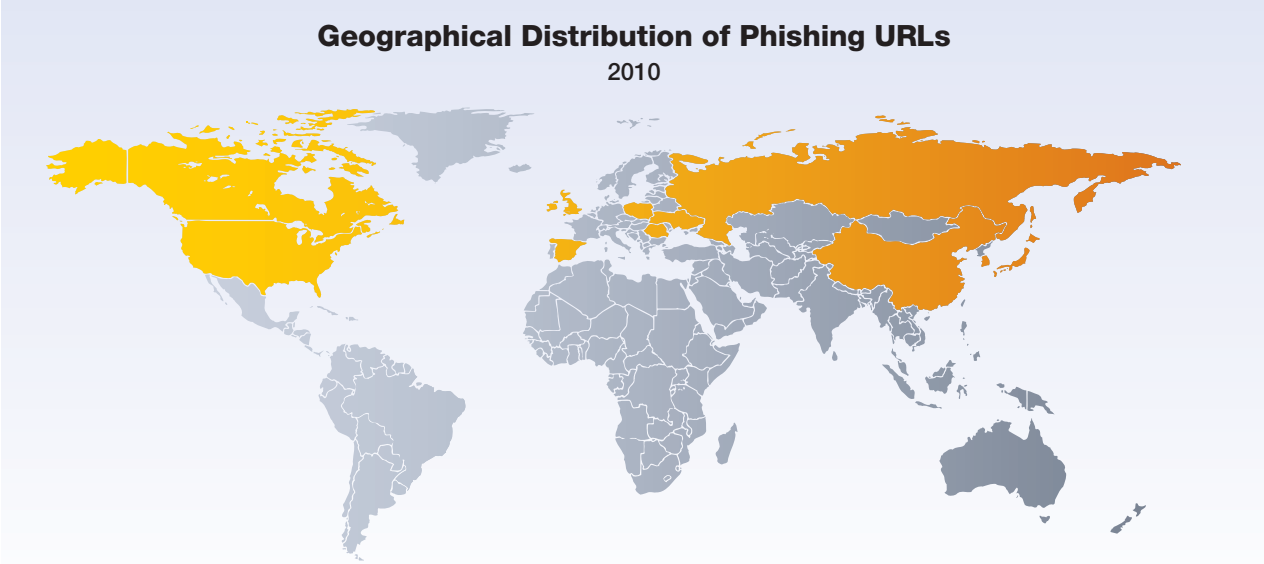


Figure 41: Geographical Distribution of Phishing URLs – 2010

Country	% of Phishing URLs
Romania	18.8%
USA	14.6%
China	11.3%
South Korea	9.8%
United Kingdom	7.2%

Country	% of Phishing URLs
Canada	4.7%
Japan	4.3%
Spain	3.2%
Poland	3.0%
Russia	2.9%

Table 9: Geographical Distribution of Phishing URLs – 2010

Phishing URLs—country of origin trends

Over the last five years, there have been many changes in the major phishing URL hosting countries. At one time, the U.S. hosted more than 50 percent of all phishing sites in 2006. In 2009 and 2010, less than one-sixth of all phishing URLs were located in the U.S. Romania hosted the most phishing sites in 2009. In 2010, the number of phishing sites in Romania increased and constitutes about 19 percent of all phishing URLs. Besides the United Kingdom, Romania is the only country with a significant increase compared to 2009.

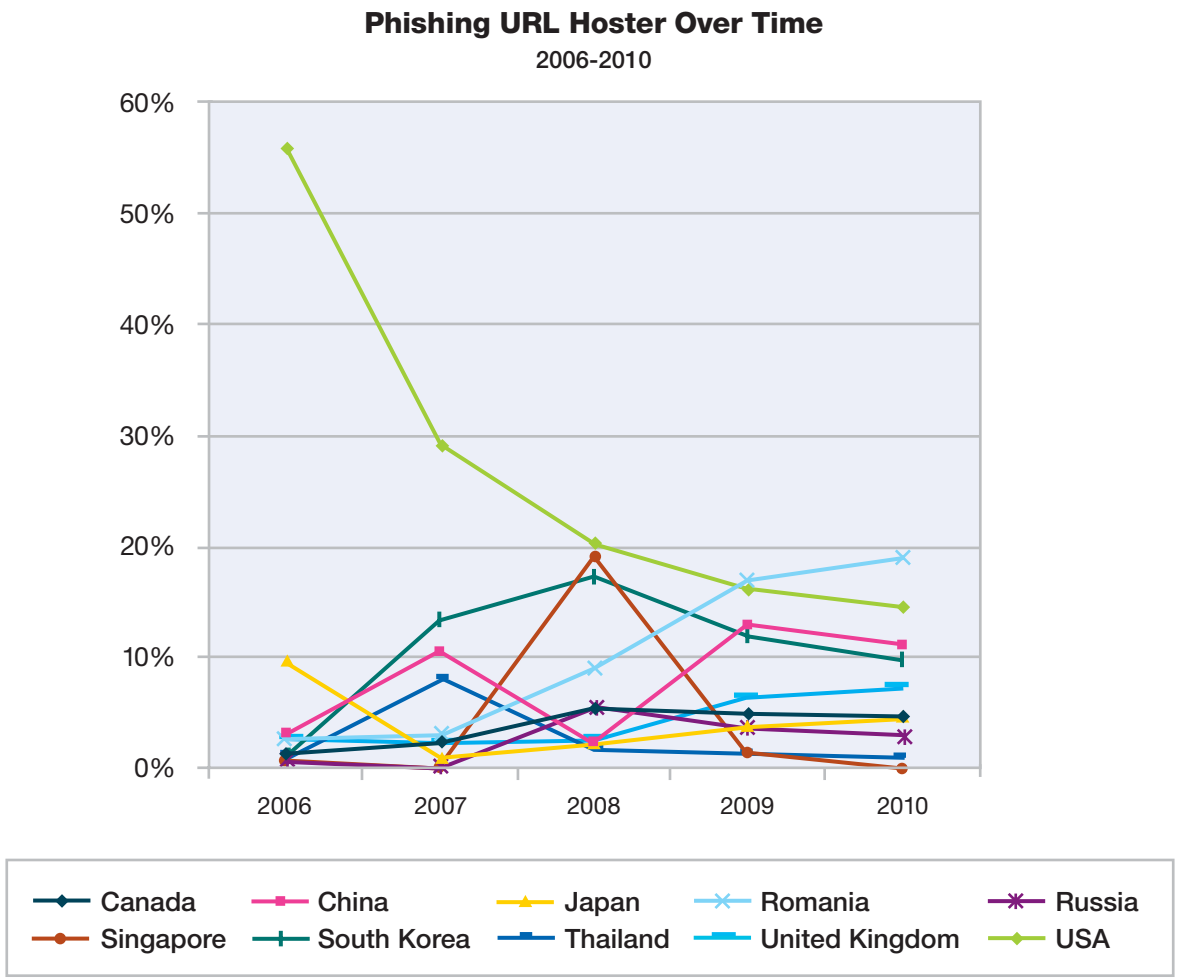


Figure 42: Phishing URL Hoster Over Time – 2006-2010

Phishing—most popular subject lines

Over time popular subject lines continue to drop in importance. By 2010, the top 10 most popular subject lines only represented about 26 percent of all phishing emails in comparison to earlier years where it represented as high as 40 percent. By far most popular subject line of the phishers is "Security Alert—Verification of Your Current Details". Nearly nine percent of all phishing emails use this subject. This text is very common and can be used for all phishing targets. Within the top 10 there are some further commonalities amongst the subject lines. All of them contain an urgent request for the user to do something—in most cases to log-in to their bank accounts by following the link in the email to a fraudulent website. On rank two, three, and four, we see subject lines targeted to special organizations or companies, and rank 10 is related to a U.S. tax website. Rank five is funny; a small typo makes the phishing email look like an advertisement for a bakery.

The following table shows the most popular phishing subject lines in 2010:

Subject Line	%
Security Alert—Verification of Your Current Details	8.62%
American Express Online Form	3.41%
Rejected ACH transaction, please review the transaction report	3.05%
Amazon.com: Please verify your new email address	2.92%
Welcome to Very Best Baking!	2.86%
For the security of your account we require a profile update.	1.50%
important notification	1.11%
Official information	1.10%
Your Account Has Been Limited	0.95%
Notice of Underreported Income	0.93%

Table 10: Most popular phishing subject lines, 2010

Phishing targets

Phishing—targets by industry

In 2009, financial institutions were unquestionably the dominant target of phishing emails. More than 60 percent were targeted to these institutions. In 2010, financial institutions remained the number one target, representing 50.1 percent of the targets. Additionally, credit cards represent 19 percent, auctions - 11 percent, governmental organizations - 7.5 percent, online payment institutions - 5.7 percent, and online shops - 4.9 percent.

The other 1.8 percent of phishing targets covers other industries such as communication services.

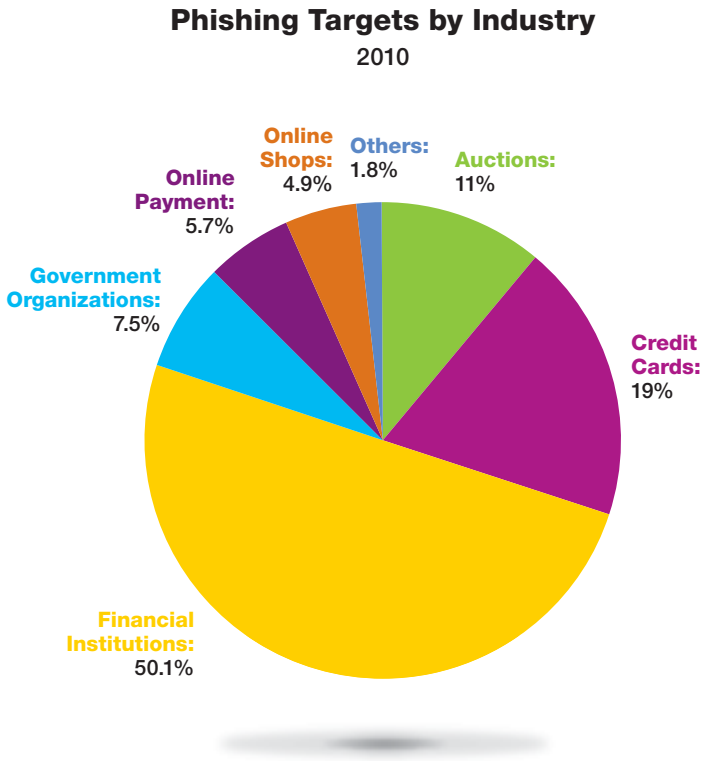


Figure 43: Phishing Targets by Industry – 2010

Until the middle of 2010, financial institutions were the predominant industry targeted by phishing emails. In the first half of 2009, online payment was a significant target of phishing emails. However, in the second half of 2009, we saw many more emails targeting government institutions (predominantly a U.S. tax-related website), credit cards, and auctions. At the same time, the percentage of phishing targeting online payment organizations declined. In the first quarter of 2010 financial institutions—still the dominant target of phishers—and credit cards declined again while auctions increased. But in the second quarter, all other industries declined, and phishers focused on financial institutions and credit cards. In the third quarter, financial institutions lost its top position for the first time, outpaced by online shops. Second runner-up in fall 2010 was online payment. But at the end of the year, the financial institutions re-conquered the top spot, and auctions became runner-up while all other industries declined.

Why did phishers stop targeting government institutions (in this case, a U.S. tax-related website) in spring 2010? One reason may be that after three quarters of targeting this tax-related website the profit was declining, and phishers were focusing on their traditional and proven business to target banks and credit cards. However, in the third quarter they seemed to try another business model by targeting online shops. This could be associated with the recovery of the global economy since more people are shopping online. Phishers returned to their traditional business to target banks in the fourth quarter of 2010.

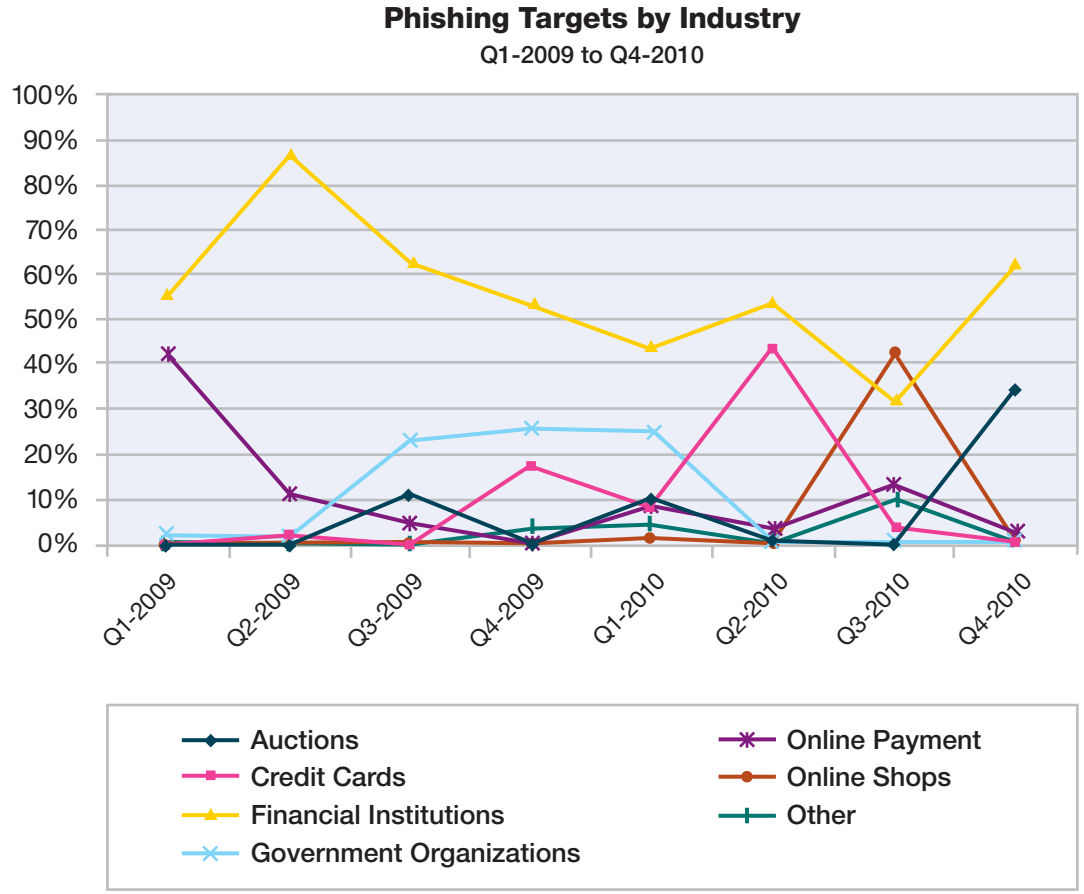


Figure 44: Phishing Targets by Industry – Q1-2009 to Q4-2010

Financial phishing targeted at banks located in the U.S.

As financial institutions remain a key focus for phishers, it is worth looking at the geographies where this activity is prominent. In 2010 more than three out of four financial phishing emails target banks located in North America. The remaining 22 percent are targeting Europe.

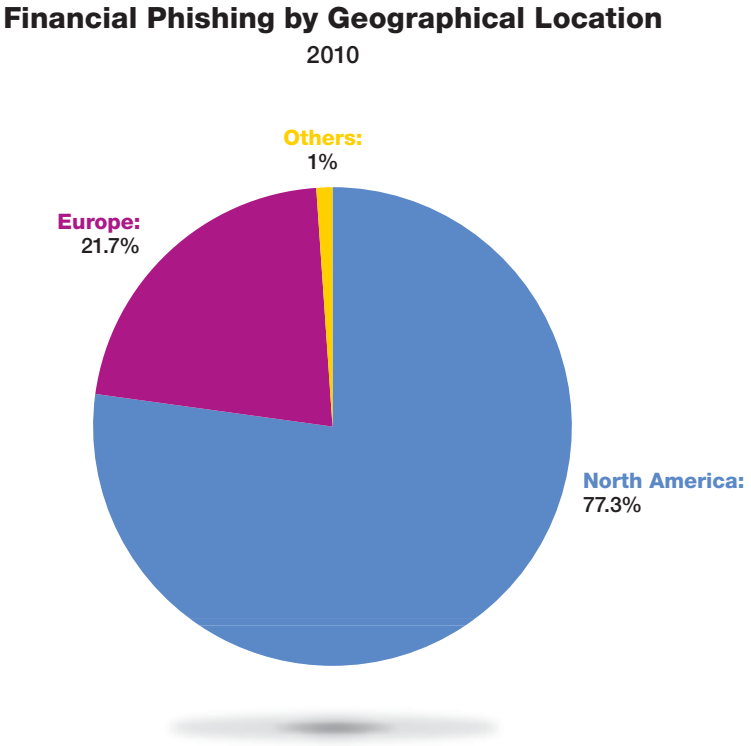


Figure 45: Financial Phishing by Geographical Location – 2010

However, after taking a closer look using shorter time frames, changes become apparent. The following chart shows the shift in geographical location that happened over the course of 2009 and 2010. While the last three quarters of 2009 were dominated by financial phishing that targeted U.S. banks (more than 95 percent), in the first quarter of 2010, nearly 45 percent of financial phishing targeted Europe. In the second quarter Europe began to decline to 24 percent, by the third quarter it was 9 percent, and by the end of the year is was nearly zero.

So why did financial phishers turn towards Europe in the first quarter of 2010 and then back towards the U.S.? A reason might be that, in the first quarter, the recovery from the financial crisis in Europe became noticeable while, in the second quarter, the budgetary crisis in Greece led to the crisis in Europe. In the second half of 2010 the budgetary crisis continued in Ireland. Furthermore, the countries of the Iberian Peninsula are under close examination concerning their national finances.

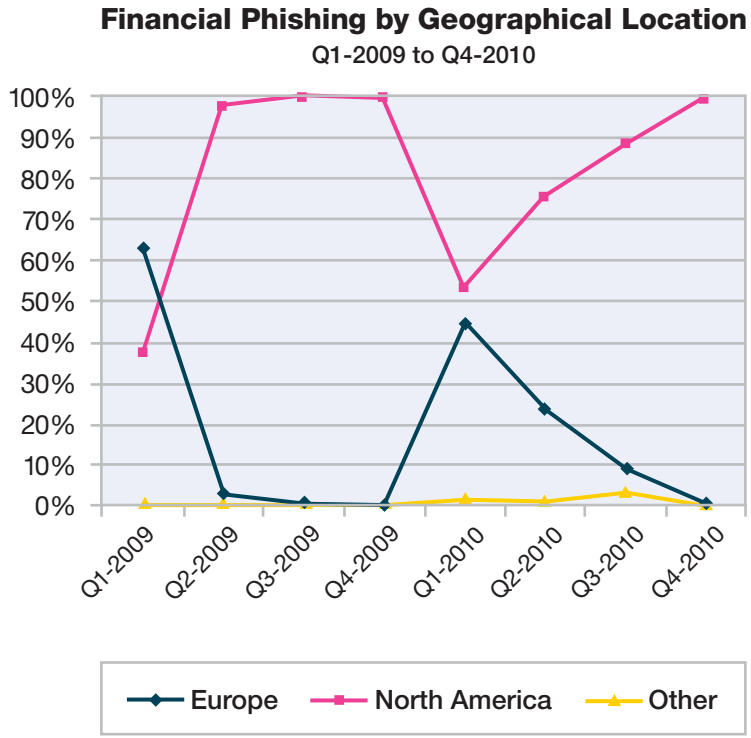


Figure 46: Financial Phishing by Geographical Location – Q1-2009 to Q4-2010

Section II—Operating Secure Infrastructure

In this section of the Trend Report we explore those topics surrounding the weaknesses in process, software, and infrastructure targeted by today's threats. We discuss security compliance best practices, operating cost reduction ideas, automation, lowered cost of ownership, and the consolidation of tasks, products, and roles. We also present data tracked across IBM during the process of managing or mitigating these problems.

Advanced persistent threat (APT) and targeted attacks

In early 2010, the term Advanced Persistent Threat (APT) became part of the everyday information security lexicon as a result of certain public disclosures and acknowledgement of a targeted series of attacks known as Operation Aurora. There has been much debate over this term and the underlying concepts within the information security community. As such, it is a topic that deserves attention and this section describes the background including historical meaning and broad interpretations, provides information based on actual response and research, and discusses how to reduce the risks associated with this type of threat.

Background and definitions

Prior to 2010, the term APT was generally used to describe a campaign or series of campaigns designed to systematically compromise systems and networks. This was based on observations by those responsible for defending certain networks and systems from attacks. Essentially, similarities across attacks were recognized, leading to the ability to classify attacks into a particular category. The term APT was given to this category and was associated with a specific adversary that was believed to have a mission for the exploitation of cyber-defense systems for the purposes of economic, political, or military gain.

During certain public disclosures in early 2010, the term APT was used when describing the attacks associated with Operation Aurora. At this point, the term began to take on a different meaning. In essence, APT became associated with any targeted, sophisticated, or complex attack regardless of the attacker, motive, origin, or method of operation.

The attention given to APT raised awareness and also sparked debate in 2010. This resulted in confusion and conflicting views. In fact, some views suggest that APT was a manufactured term for purposes of marketing security services while other views point out the specific nuances that define APT for them. While multiple viewpoints exist, it is important to note that this type of threat is a legitimate issue for certain organizations.

Response and research

The IBM Emergency Response Services (ERS) practice has been responding to computer security emergencies for over 10 years. Over the course of these incidents, there have been multiple constants: new vulnerabilities exploited, new attack vectors, new tools, and new techniques that are used by the adversaries we face. IBM X-Force often refers to this concept as the evolving threat.

In recent response efforts involving incidents of this type, we have noticed a sharp increase in the convergence of attack vectors and techniques. This is the single largest reason that attacks of this type are referred to as complex or sophisticated. In fact, many of the tactics used by adversaries with capabilities in this category are not individually unique or advanced. It is only when the procedures and tools are combined that the complexity begins to increase exponentially.

As an example of complexity, in many cases the attackers perform reconnaissance that goes beyond the simple ability to understand how to compromise the initial victim. In fact, the initial system is often not the ultimate target. Once the initial compromise occurs, the attackers may use various tactics to perform additional reconnaissance or may compromise the next host. These tactics can include things like privilege escalation, which might take place when the attacker has compromised a system at the user level and then subsequently runs a local exploit to gain administrative privileges providing an ability to use that system for lateral movement within the network to access another system.

The single most common threat vector used over the past few years as observed by ERS is spear phishing where an object contains a link to a web page that contains malware. The delivery of this type of message to victims can occur through email, instant messaging, and social network sites. The type of malware and method of initial compromise can differ as well. In many cases, different malware is used within the same attack wave to compromise different systems throughout the organization. While this is not always zero-day malware, there have been many instances where the malware used is not observed in the wild. This makes detection challenging, but there are generally accepted response procedures that can help identify compromised systems based on common indications and characteristics shared between compromised systems.

Often a high-value target is an end-user system such as one that belongs to person who has access to sensitive data. This might be an executive user, someone involved in strategic negotiations, or simply an engineer. Alternatively, a high-value target could be an actual server that contains sensitive data. While these are not novel concepts for information security professionals, understanding the progression of an attack and the motive of an attacker is essential.

With this type of threat, it becomes increasingly imperative to understand the type of data that an adversary is interested in rather than focusing too heavily on a specific attack vector, malware, or weakness. This is partly because there is evidence to suggest that this type of adversary has resources to study and understand the weaknesses of a targeted organization. Of course, it is still important to understand the specific weaknesses that exist because it is a good idea to close security gaps wherever possible depending on the overall cost and complexity of the solution required.

Another aspect with respect to the complexity of a targeted attack is that the attacker has an objective and a desire to achieve that objective. As such, a motivated attacker of this type is invested in the success of the mission and will expend resources to maintain unauthorized access. This includes observing remedial actions taken by a victim organization and using tools and tactics as activity is discovered and access is removed. Sometimes these tools and tactics are different and more sophisticated, but the key is that the attacker is dedicated to maintaining a persistent capability to extract data.

Finally, it is important to understand data exfiltration methods used to get sensitive data out of the target environment. While there are numerous ways to exfiltrate data, ERS has observed that attackers in this category often attempt to use some form of encryption and/or obfuscation to exfiltrate the data. This could be as simple as creating an encrypted compressed archive of files, or a bit more complicated such as the use of an encrypted tunnel. Regardless of the exfiltration method chosen, the common denominator seems to indicate that the attacker attempts to use legitimate protocols and masquerade as a legitimate user whenever possible. This is another reason to classify this type of threat as sophisticated.

Conclusions and recommendations

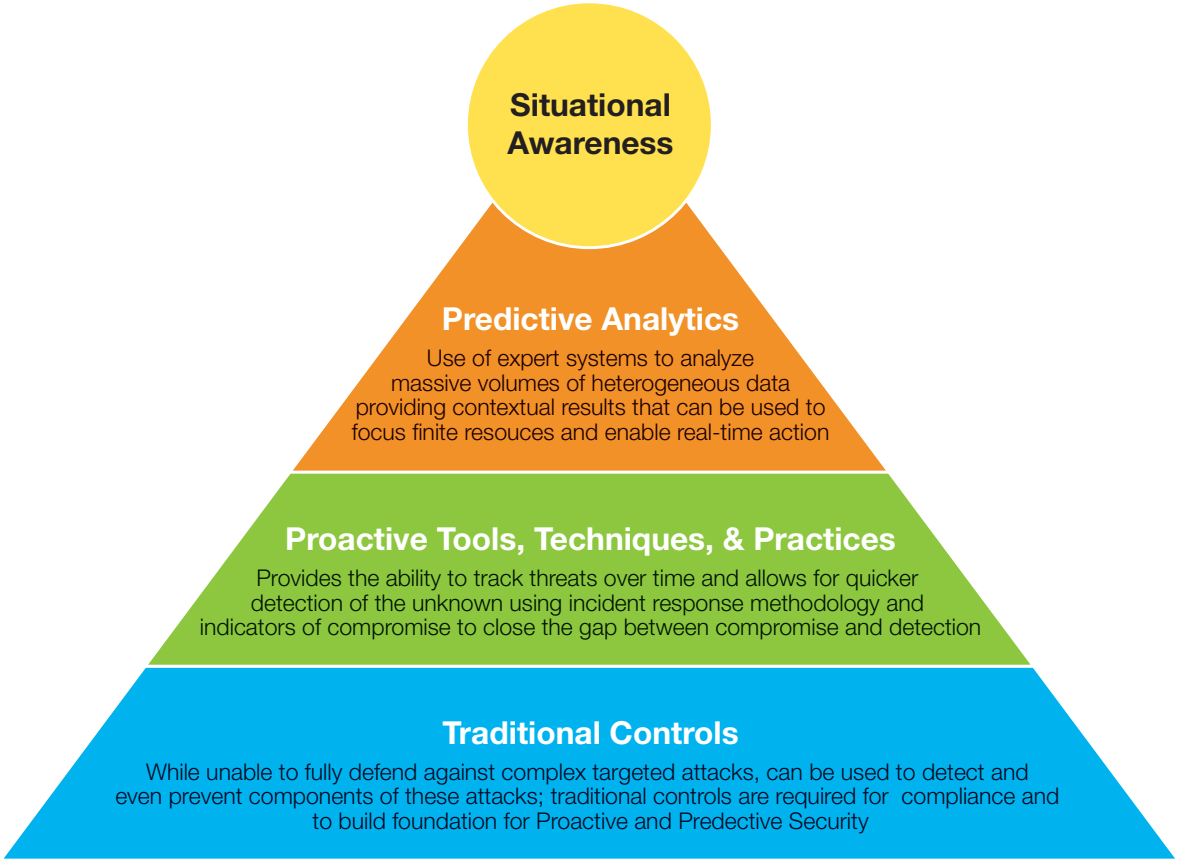
In summary, this is a dynamic and challenging problem; however, it is our strong belief that significant steps can be taken to understand and combat this type of threat through better situational awareness.

First and foremost, the decisions made with respect to the recommendations in this section should be made using a risk-based approach. That is, if your organization has been subjected to attacks of this nature, then these recommendations may be more likely to apply. If you are unsure or concerned about this type of threat, we suggest that you perform a specific threat assessment. This type of assessment should be performed by an organization familiar with this type of threat that can use knowledge and experience from previous engagements, as well as proper tools and sound scientific methods, to determine if there is a known issue.

These recommendations are not exhaustive but rather are listed here to show what has been successful based on previous engagements. They are considered best practices for mature organizations that want to have a comprehensive and superior security posture. In every case that ERS has been involved with, recommendations are made based on the use of current instrumentation that can be leveraged within a specific organization to augment the ability to get better situational awareness. If there is a particular concern, we recommend additional discussions to validate the listed recommendations and to discuss the current implementation and use of tools.

- **Use information risk management principles.** Consider the type of industry your organization is within, the type of information handled, and the perceived value from the adversarial perspective. Determine high value targets, the location of sensitive data, and the overall data flow with respect to sensitive data. Implement a tier-based control framework to ensure proper protection.
- **Enhance information security controls.** Traditional controls are absolutely necessary but compliance should not be the only driver for information security if an organization intends to have a robust and advanced security posture. An examination of current controls can help reveal gaps in capabilities. Analysis should be based on current instrumentation, with a focus on answering the questions that need answering to combat this type of threat. Enhancements can include the entire gamut of potential products but here are a few significant recommendations:
  - Use threat feeds and reputation services to help identify if an internal system is communicating with an external system that is known to be associated with malicious activity.
  - Use DNS and DHCP logging to understand internal hosts that may be compromised based on a known bad IP address.
  - Use network forensic tools to help detect anomalies such as malformed packets.
  - Investigate nefarious activity based on information obtained from various data points.
  - Use host-based enterprise forensic tools that aid in the detection of compromised systems based on shared characteristics and memory analysis for malicious characteristics.

Note that while network-based tools may seem like a more economical approach, the use of host-based tools is highly recommended in conjunction with network tools to be most effective.



- **Validate the incident response process.** Specifically, this means ensuring that the incident response process aligns step by step with enterprise tools and mapping the flow of the investigative process to the technology. Aspects of leveraging tools for detection, using advanced response techniques, and proper remediation timing should be taken into consideration. Use of an applicable process framework such as ITIL can be helpful.
- **Build a comprehensive data breach program.** This should include a Data Breach Program Manager that has the responsibility for coordinating all aspects of response efforts beyond the technical components. Specifically, this type of program should effectively establish a framework to ensure proper communications and information flow during a large-scale complex data breach. The areas to include would be the appropriate business leaders, data owners, information technology operations, legal counsel, public relations, security operations, etc. One of the goals should be to privately share information with external entities including law enforcement and industry groups.
- **Establish a dedicated response team.** While many organizations have an incident response team, they are often overwhelmed trying to determine what is significant, in part due to the sheer volume of malware that can be found in any given organization. Many successful organizations have built a dedicated advanced incident response team that can take advantage of the tools designed to assist with advanced threats. This team should attempt to focus efforts on proactive assessments if the tools are available to do so.
- **Consider a next generation predictive solution.** While security has begun to evolve from a reactive stance to a proactive stance, the methodologies used are still quite resource intensive in that they require human analysis. Sometimes, this analysis can lead to wasted productivity and significant cost. Use of an expert predictive system to model massive volumes of heterogeneous streaming data in real-time can be extremely valuable. This type of capability can learn from the past based on models, evaluate new information against previous information as it arrives, determine if the data is relevant, and even use new observations to reverse earlier assertions. This can tell us where to focus, and possibly recommend what to do or even take action for us when there is extremely high confidence (99.9%).

Stuxnet and SCADA

Midway through 2010 a new piece of malware was discovered that caused those outside the computer security field to take notice, and those inside the field to be exceptionally concerned. This new worm caused (and continues to cause) an immense amount of speculation regarding its origin, purpose, and targets. Named Stuxnet for some keywords found inside the program, this worm looks for a very specific environment before enabling its payload. Over the latter part of 2010, many media organizations speculated that this piece of malware was sponsored by a nation state or that this malware was created for a specific target.

These questions remain unanswered, but many facts have come to light about how Stuxnet transmits itself and what actions the payload takes once its intended environment is found. One of the factors that made those in the computer security field initially take notice is the complexity of this program and the quantity of zero day exploits used in this worm. Zero day exploits are those that have no work around or patch. This has not been seen in any other malware packages to date. Another unique aspect of Stuxnet is that it contained components that were digitally signed with stolen certificates. Further analysis by computer security researchers caused yet more concern as a root kit was found for the programmable logic controller

(PLC) which allows the manipulation of sensitive equipment. In this article, we take a look at what is true and what is speculation. We also point out what customers who do not have SCADA (Supervisory Control and Data Acquisition) equipment should look for and be concerned about with regards to Stuxnet.

Who is behind Stuxnet?

There are no solid facts regarding who is responsible for writing or funding Stuxnet. Some in the media speculate that Israel may be involved, but there is little evidence for that. One thing seems rather clear: this likely was not something created by a single individual. According to one published report (Madrigal, 2010), this could have been created by a team of as many as 30 individuals. This indicates a level of organization and funding that probably has not been seen before in the security field. What was Stuxnet designed to do? While we do not have any direct evidence to support the intent of the author(s), the programming code suggests that Stuxnet looks for a setup that is used in processing facilities that handle uranium used in nuclear devices.

After several months of analysis, discoveries suggest that Stuxnet alters the frequency at which processing centrifuges spin, which can cause permanent damage to those devices and their contents. Additional evidence suggested that the code also contained an element that falsely reported that the equipment was functioning normally (Broad, Markoff, & Sanger, 2011). This would have made it harder to detect its presence as well as harder to detect the damage it was doing to equipment. The question of where Stuxnet was targeting is one of the few areas for which we have some evidence. Many reports show that nearly 60% of the initial infection was centered on Iranian systems (Thakur, 2010).

The Stuxnet malware itself contained many different components and took advantage of four then-unpatched vulnerabilities in Windows systems. Two of the vulnerabilities were used to spread Stuxnet—the LNK vulnerability (CVE-2010-2568) and the Printer Spooler vulnerability (CVE-2010-2729). The other two vulnerabilities were used to elevate privileges on already infected machines—the Win32k.sys keyboard layout vulnerability (CVE-2010-2743) and the Task Scheduler vulnerability (CVE-2010-3888). (Since the detection of Stuxnet, each of these vulnerabilities has been patched by the vendor.) Stuxnet can take advantage of a default password in the Siemens WinCC software's database server as well as infect Siemens Step7 project files. There are rootkit components that have been signed with stolen certificates which make it difficult to fully clean an infected system. The certificates used have since been revoked, but it is still possible for Stuxnet to infect a system.

One question asked by many of our customers is "We do not have SCADA systems in our facilities, why should we care about Stuxnet?" While Stuxnet's payload might not apply to those that do not have SCADA equipment or the particular SCADA equipment that Stuxnet targets, the infection itself does impact affected computers. Stuxnet contains many components—including kernel-mode drivers—that can affect the reliability and performance of a PC. Stuxnet's installation of a peer-to-peer communication component opens infected machines to unauthorized remote access. A Stuxnet infection can also indicate the presence of unpatched vulnerabilities on networked computers.

One of Stuxnet's infection vectors is through portable USB drives and the use of the LNK vulnerability (CVE-2010-2568). From a policy perspective, one should review the use of USB drives. Many institutions including the United States military have opted to ban the use of these drives to limit threats that target that transmission method (Shachtman, 2010). Also, customers might not be directly impacted by Stuxnet, but it will not likely take long for other malware writers to copy aspects of Stuxnet for their own uses. Taking actions to help protect against aspects of Stuxnet will help protect against future threats that have yet to be discovered.

New information continues to come to light on details of Stuxnet and we expect more to come. This is the first big example of a cyber-weapon being discovered and publicly analyzed. Not surprisingly, the media have been covering the event heavily. These types of media reports on computer security are going to become more consistently seen going forward. Reports like this often are geared towards the general public, not computer security specialists.

Works cited

Broad, W., Markoff, J., & Sanger, D. (2011, January 15). New York Times. Retrieved January 21, 2011, from New York Times: **http://www.nytimes.com/2011/01/16/world/middleeast/16stuxnet.html**

Fanelli, J., Sisk, R., & Siemaszko. (2010, September 23). New York Daily Times. Retrieved January 20, 2011, from New York Daily Times: **http://www.nydailynews.com/news/politics/2010/09/23/2010-09-23_mahmoud_ahmadinejads_un_speech_discussing_911_prompts_us_delegation_to_walk_out.html**

Madrigal, A. (2010, November 4). The Atlantic. Retrieved January 20, 2011, from The Atlantic: **http://www.theatlantic.com/technology/archive/2010/11/the-stuxnet-worm-more-than-30-people-built-it/66156/**

Thakur, V. (2010, July 22). Symantec. Retrieved January 20, 2011, from Symantec: **http://www.symantec.com/connect/blogs/w32stuxnet-network-information**

Shachtman, N. (2010, December 9). Retrieved February 18, 2011, from Wired: **http://www.wired.com/dangerroom/2010/12/military-bans-disks-threatens-courts-martials-to-stop-new-leaks/**

Public vulnerability disclosures in 2010

The fundamental challenge posed by computer security is the asymmetric nature of the threat. Security professionals should identify and mitigate every single vulnerability in complex infrastructures, but an attacker need only find one to be successful. X-Force Research was founded in 1997 with the mission of understanding everything there is to know about security vulnerabilities. One of the first things that we did was create the X-Force Database—which tracks every single public security vulnerability disclosure, whether it comes from a software vendor or a third party.

At the end of 2010, there were 54,604 vulnerabilities in the X-Force Database, covering 24,607 distinct software products from 12,562 vendors. These go all the way back to a CERT advisory about FTPd published in 1988. All of this vulnerability data was entered by our database team, who search through security mailing lists, vendor security bulletins, bug tracking systems, and exploit sites in order to catalog every public vulnerability disclosure, along with the release of patches and exploits for those vulnerabilities. These disclosures live in our database, unless they are publicly refuted by the vendor or another reputable source.

The X-Force database is an invaluable operational tool for us in X-Force. If a vulnerability is being discussed somewhere on the Internet, we need to be aware of it, so that we can assess it and to help ensure our customers are protected from attacks that target it. The development of the security content for the vulnerability assessment and intrusion prevention products that we make is driven by vulnerability disclosures and the X-Force database.

We think our customers should be aware of these vulnerability disclosures too, so that they can respond by patching or through other means. Every vulnerability that we catalog can be viewed and searched on our website. The purpose is to provide a central resource where people can investigate security issues and find the latest information available from IBM. We also make vulnerability information available to customers directly via daily emails from our customizable X-Force Threat Analysis Service.

2010—A record setting year

From our perspective, 2010 had the largest number of vulnerability disclosures in history—8,562. This is a 27 percent increase over 2009, and this increase has had a significant operational impact for anyone managing large IT infrastructures. More vulnerability disclosures mean more time patching and remediating vulnerable systems.

The relative mix of vulnerability severities has not changed substantially for the past three years. X-Force ranks vulnerabilities in our database as Critical, High, Medium, or Low based on the industry standard Common Vulnerability Scoring System (CVSS) scores. Vulnerabilities with a CVSS base score of 10 are counted as critical; 7 to 9 are counted as high; 4 to 6 are counted as medium; anything else is counted as low. The vast majority of vulnerability disclosures are rated medium (60 percent) or high (33 percent) severity based on this CVSS methodology.

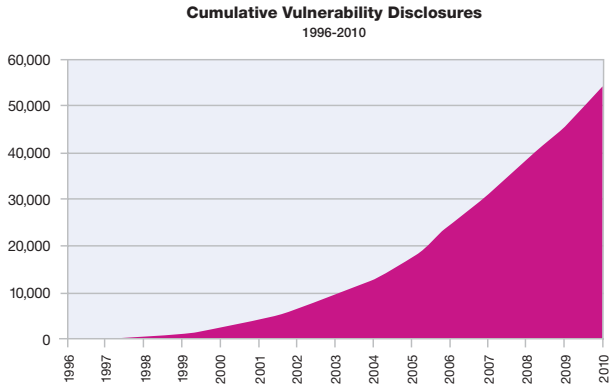


Figure 47: Cumulative Vulnerability Disclosures – 1996-2010

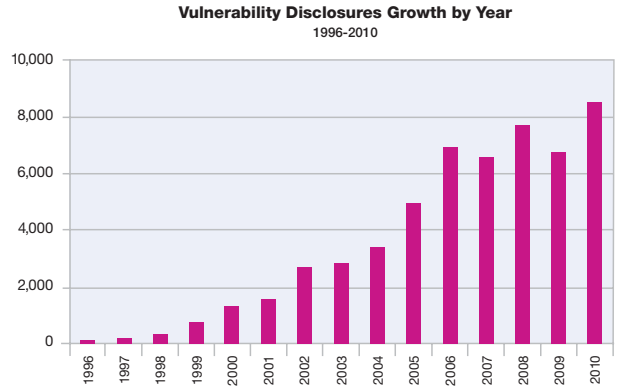


Figure 48: Vulnerability Disclosures Growth by Year – 1996-2010

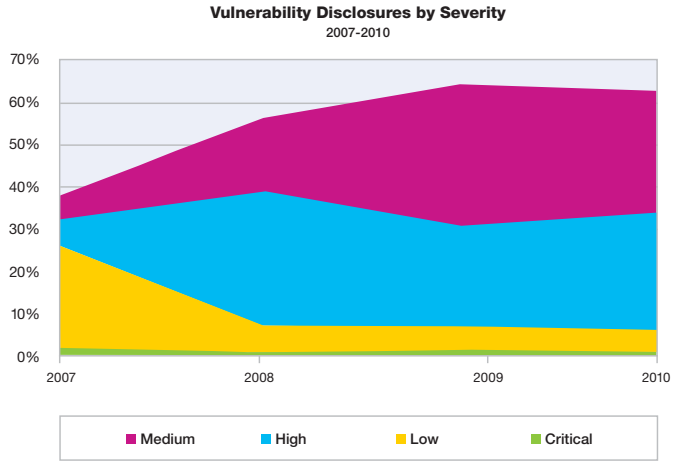


Figure 49: FVulnerability Disclosures by Severity – 2007-2010

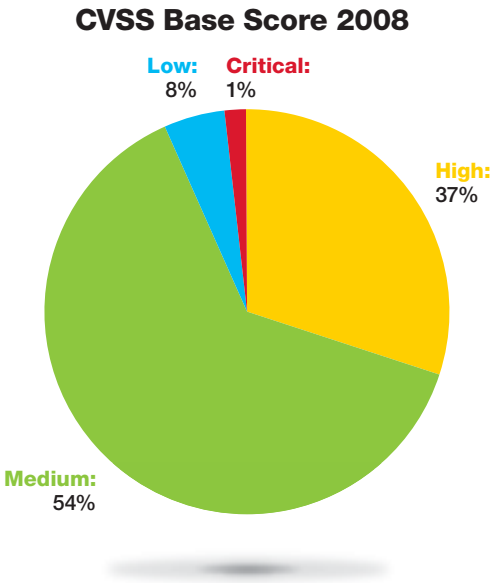


Figure 50: CVSS Base Score 2008

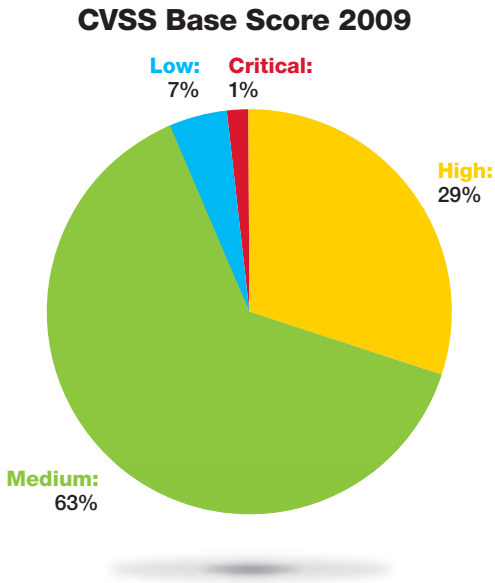


Figure 51: CVSS Base Score 2009

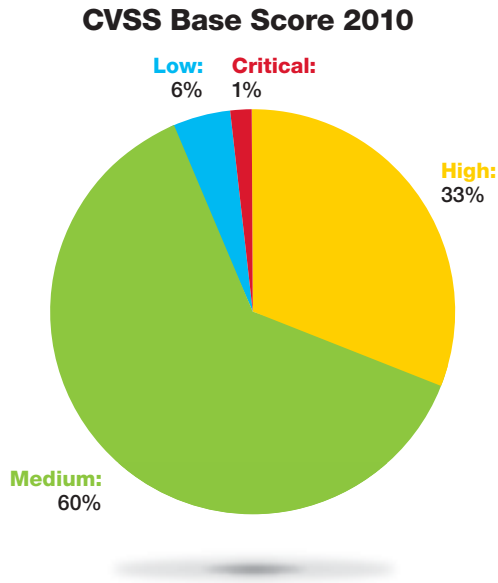


Figure 52: CVSS Base Score 2010

Were there really more vulnerability disclosures in 2010?

It's worth noting that X-Force is not the only organization that tracks and counts vulnerability disclosures. Every organization that does this has a slightly different perspective on the total number of vulnerabilities disclosed in a given year and how large the fluctuations are, year over year. We believe that there are two factors that influence these differences in perspective. The first and most important factor is the number of sources of vulnerability disclosure that an organization is tracking. X-Force strives to be as comprehensive as possible, but many organizations choose not to count everything. For example, Common Vulnerabilities and Exposures (CVE) is the industry standard naming scheme for vulnerabilities. However, only 4,128 CVEs have been made public for 2010 at this time (when actually over 10,000 CVE's were issued for the year). Often, CVEs are not issued for vulnerabilities impacting software made by small, independent software developers.

The second factor that impacts the number of vulnerability disclosures is the number of individual vulnerabilities counted in a particular vulnerability report. In some cases, a single vulnerability appearing in a single vulnerability disclosure report might appear in multiple places or be exploitable through multiple vectors within an application. This is particularly common with web applications which may be bundled with a large number of scripts that all share the same vulnerability due to shared code or a single shared library. Such a vulnerability might be counted multiple times or a single time, depending on the standards set by a particular vulnerability tracking organization.

However, when X-Force digs beneath the surface of the total number of vulnerability disclosures that we witnessed this year, we see increases in important areas that support our hypothesis that 2010 was a particularly busy year for those of us who work with security vulnerabilities. When we look at the ten enterprise software vendors with the most total vulnerability disclosures (excluding open source web content management platforms), the average increase was 66%, with eight of the ten vendors seeing more vulnerability disclosure in 2010 versus 2009.

There is a complex set of dynamics that impacts the volume of vulnerability disclosures coming from a particular vendor, including the total number of products a vendor supports and the complexity of those products, the maturity of their internal efforts to find and fix security issues, the amount of external vulnerability research targeting that vendor, mergers and acquisitions, etc. However, we think that such a significant increase across the board signifies efforts that are going on throughout the software industry to improve software quality and identify and patch vulnerabilities. The ensuing increase in vulnerability disclosures is keeping a lot of us busy tracking and patching these issues on our networks. Hopefully, all of this work is moving us toward a future in which much of the software that we are using is much safer than it is today.

Public exploit disclosure

Public exploit disclosure was also up 21 percent in 2010 on a real basis versus 2009, although not on a percentage basis. Approximately 14.9 percent of the vulnerabilities disclosed in 2010 had public exploits, which is down slightly from the 15.7 percent last year, but because so many more vulnerabilities were disclosed this year, the total number of exploits increased.

The vast majority of public exploits are released the same day or in conjunction with public disclosure of the vulnerability. Many are released within one week of disclosure. However, we still see a small number of exploits surfacing tens or hundreds of days after initial public disclosure. In many of these cases, attackers may have had private access to these exploits shortly after (or even prior to) public disclosure of the vulnerability. The exploit code only emerges publicly after its usefulness to the attackers has diminished. This happens slowly over time as more and more vulnerable hosts are patched or upgraded. Thus, the long tail of exploit releases is a window into some of the real world attack activity that networks are facing in the time period between patch releases and patch installation. Keeping this window as short as possible is an important element of running a secure network.

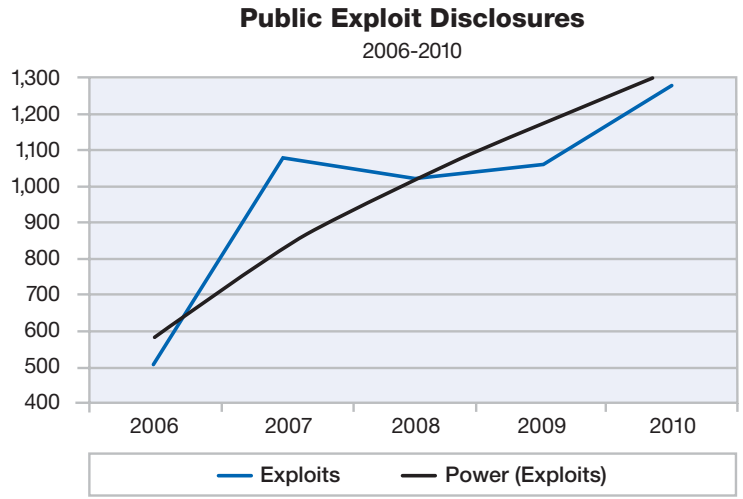


Figure 53: Public Exploit Disclosures – 2006-2010

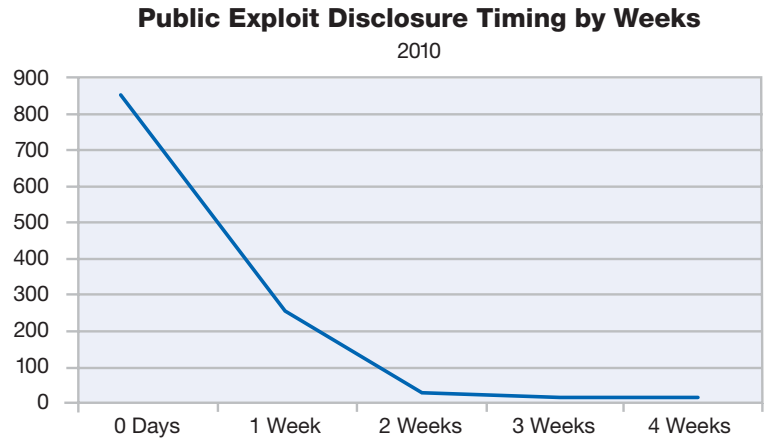


Figure 54: Public Exploit Disclosure Timing by Weeks – 2010

	2006	2007	2008	2009	2010
True Exploits	504	1078	1025	1059	1280
Percentage of Total	7.3%	16.5%	13.4%	15.7%	14.9%

Table 11: Public exploit disclosures 2006 – 2010

Exploit Timing	0 Days	1 Week	2 Weeks	3 Weeks	4 Weeks
0 Days	854	270	18	9	9

Table 12: Public Exploit Disclosure Timing by Weeks – 2010

Vendor supplied patches

Approximately 44.1 percent of the vulnerabilities that were disclosed during 2010 currently have no vendor supplied patch information in our database. How quickly do patches become available for publicly disclosed issues? This is an important question for network operators. Table 13 provides some insight into this question. The first column shows how many weeks after public disclosure patches became available for vulnerabilities in our database. Fortunately, most patches become available for most vulnerabilities at the same time that they are publicly disclosed, however that isn't always the case. Some vulnerabilities are publicly disclosed for many weeks before patches are released. We are only showing the first eight weeks of data in this chart but these numbers trail off over time with the odd vulnerability being fixed hundreds of days after initial public disclosure.

In order to maximize the relevance of this data, we looked specifically at a list of vendors that X-Force considers the most important because they make the most popular enterprise software. The third column limits our inquiry to these important vendors. Even in this case, there are often many weeks between vulnerability disclosure and patch release. The worst example in our dataset was 313 days.

Why do these gaps exist and why can they be so long? The situation faced by a vendor varies on a case by case basis. Obviously, vendors try to avoid public disclosure of vulnerabilities that have not been fixed in order to protect their customers, however disclosure is not always under the vendor's control. Unfortunately in many cases where security vulnerabilities are disclosed without vendor coordination, some exploitation details are also publicly released. Some vulnerabilities are trivial to fix, but even in the best case, time is required to verify the vulnerability report, fix the bug, verify the fix, and test update packages before they are released to customers. In more complicated cases, a single vulnerability might need to be fixed in a wide array of different supported versions and packages of a particular product, all of which need to be updated and tested. Changes to a piece of software may also require other changes to additional software components that it relies upon. In the most complicated cases we've seen, a fix to a single security vulnerability requires coordination with an ecosystem of different vendors who make software that incorporates or relies upon the component that is being changed. This sort of multi-vendor coordination can be extremely complicated and, frankly, slow.

What this means for network administrators is that there are going to be publicly disclosed exposures in our networks no matter how much pressure is put on vendors to improve their patch responsiveness. It is important to recognize this reality and plan effectively—although patch management is an important part of running a secure network, it is not sufficient to protect a network from known threats, not to mention the risk of zero-day attacks of which vendors are not aware.

Patch Timeline	All	Top Vendors
Same Day	3400	1814
Week 1	192	34
Week 2	55	11
Week 3	57	12
Week 4	33	7
Week 5	27	7
Week 6	22	4
Week 7	17	3
Week 8	16	8

Table 13: Patch release timing 2010

Toward more reliable public vulnerability reporting

Keeping track of public vulnerability disclosures and remedy information across thousands of vendors is a challenging task. No one should have a better understanding of the true status of security issues impacting a particular vendor's products than the vendor itself. However, that perspective is not always perfectly reflected in public information resources about security vulnerabilities. Every software vendor takes a different approach in how they respond to public vulnerability reports. Some vendors do not respond to every public report. Some only respond privately in forums accessible to paying customers. Some responses are not clearly tied to the public vulnerability reports that they are intended to address. As a consequence, while it is relatively easy to pick up on public security vulnerability reports and catalog those, tracking down remediation information can be challenging.

We think that better standardization of vulnerability reporting would help improve the consistency of this sort of information. Currently an effort is underway to develop an XML standard for publishing security advisories and remedy information called the Common Vulnerability Reporting Format (CVRF). This standard is being developed through a multi-vendor effort under the auspices of the Industry Consortium for Advancement of Security on the Internet (ICASI)—a forum through which the IT industry addresses multi-product security challenges. The first draft of this standard has yet to be published as of this writing, but it should include a mechanism that allows vendors to clearly indicate the status of a vulnerability remediation effort, including cases where they dispute a public vulnerability report. We think that as CVRF matures and is adopted, it should help to eliminate questions and concerns about inconsistent information and differences of perspective regarding the remediation status of a vulnerability.

It may take a long time before we get to a point where most major software vendors are publishing remediation information in a standard format for every public disclosure, but the benefits should be enormous. Every IT operation is tasked with running a complex array of different types of products in a production environment. It can be difficult to keep on top of the myriad of different vulnerabilities that may affect those products while also making sure that remedies are installed promptly. The more reliable and comprehensive public vulnerability information resources are, the easier these tasks will be.

With standardized vulnerability reporting coupled with advanced endpoint management technology one could also imagine a high level of automation, wherein network managers could monitor exposures across the entire enterprise. When a vulnerability is disclosed, endpoint management systems could automatically deploy temporary workaround measures or temporarily disable the vulnerable component. Later, when a patch becomes available, it could be automatically deployed and the workaround reverted. This approach would result in more consistent security posture with less concern about missing an important detail that might be leveraged by an attacker.

Shift from local to remotely exploitable vulnerabilities

The most obvious question that one may ask about the various vulnerabilities disclosed in 2010 is what kind of exposure do they represent? By and large, they are remote code execution vulnerabilities—this is as opposed to local privilege escalation issues. Twenty years ago, individual computer systems, particularly with Internet access, could be relatively expensive. Many had to be shared among multiple users. In this environment, privilege escalation vulnerabilities were valuable to an attacker who might obtain access to an individual user account on a multiuser system and seek to gain full control over the system.

As computers became less expensive we gradually entered an era where one computer system, generally speaking, served one function—one has a separate mail server, web server, database server, and so on. These machines usually do not have a lot of individual user accounts—they are generally accessed by the individual who administrates the system. Therefore, in this environment, privilege escalation vulnerabilities typically do not have as much value. Over time we have seen a corresponding shift in vulnerability disclosure from local to remote issues.

We are presently starting to enter a third era, where individual computer systems have become so powerful that it is usually inefficient to use them for just one function. Enter virtualization, where the one system, one function principal is maintained by running a number of different virtual systems on a single hardware platform. Here, local privilege escalation issues generally are still only marginally valuable. However, a new vulnerability class has arisen—hypervisor escape vulnerabilities that allow an attacker with control over one system to control the other systems running on the same physical machine.

Although relatively rare, our study on virtualization vulnerabilities published in the **2010 Mid-Year X-Force Trend Report** showed that these vulnerabilities are the most common type disclosed in virtualization software. It will be interesting to see if their numbers increase as we continue to shift into the virtualization era. Fortunately, we have learned a lot about designing secure software in the past 20 years and we are bringing those lessons into this new environment. You can read more about our findings in the area of virtualization later in this report on page 90.

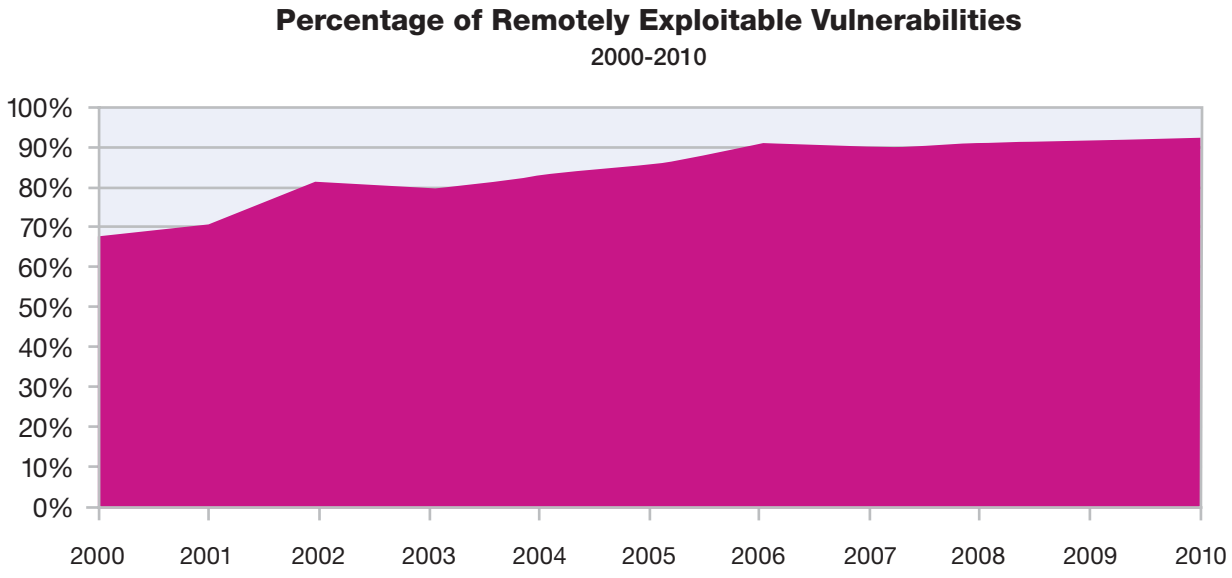


Figure 55: Percentage of Remotely Exploitable Vulnerabilities – 2000-2010

Web application vulnerabilities

What kind of software do these vulnerabilities impact and how do they relate to the attack methodologies employed by the bad guys? We think that the real threat today revolves around the web. The web is the primary platform on which network applications are developed. A great deal of functionality has been pushed into the protocols both on the client and the server side. The complexity of these systems has produced a wealth of vulnerability disclosures and attack activity.

Let's start on the server side with web application vulnerabilities. Forty-nine percent of the vulnerabilities disclosed in 2010 were web application vulnerabilities. The majority of these were cross-site scripting and SQL injection issues. However, as we have been saying for years, these vulnerabilities represent just the tip of the iceberg. In the X-Force database, we track public vulnerability disclosures. When it comes to web applications, this means vulnerabilities in web apps that are maintained for use by third parties, such as commercial web application frameworks or open source projects. The majority of web applications are custom—they are developed by in-house or outsourced development teams to meet a very specific need. These custom web apps are not usually subject to public vulnerability disclosure because there is no reason to notify the public about a vulnerability in a private web app.

Therefore, the total number of web application vulnerabilities is likely much larger than the quantity of public reports that we track in our database. Web application vulnerabilities may vastly exceed the quantity of other kinds of security issues on the Internet.

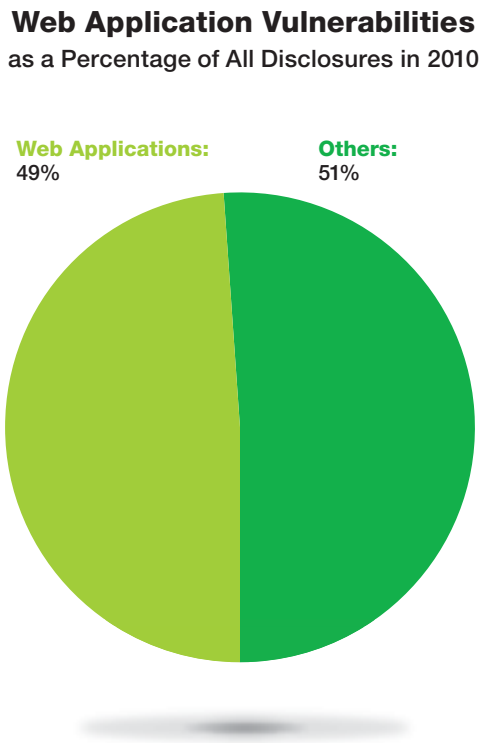


Figure 56: Web Application Vulnerabilities as a Percentage of All Disclosures in 2010

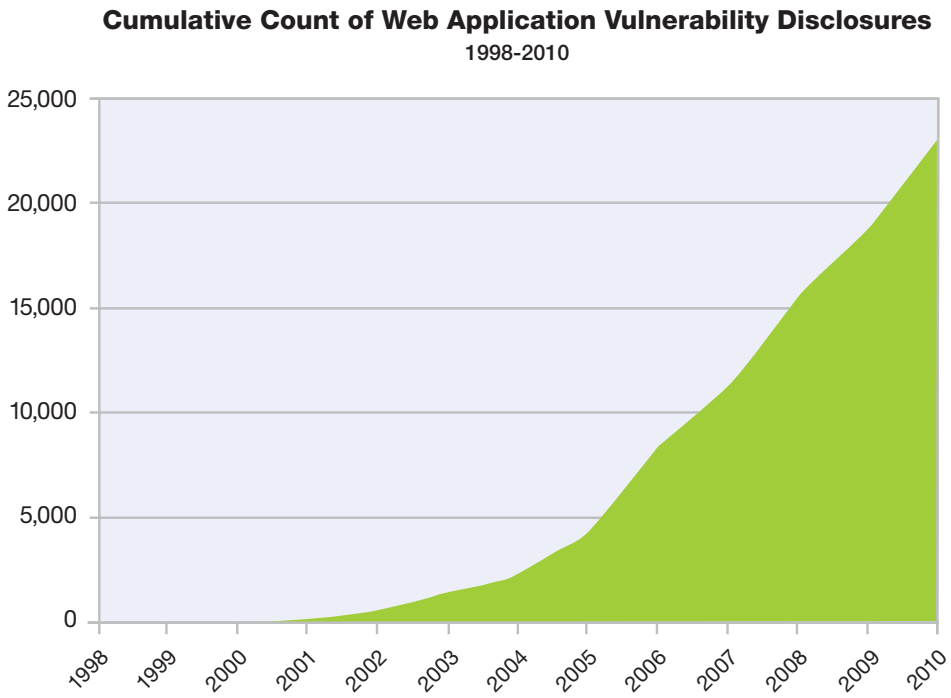


Figure 57: Cumulative Count of Web Application Vulnerability Disclosures – 1998-2010

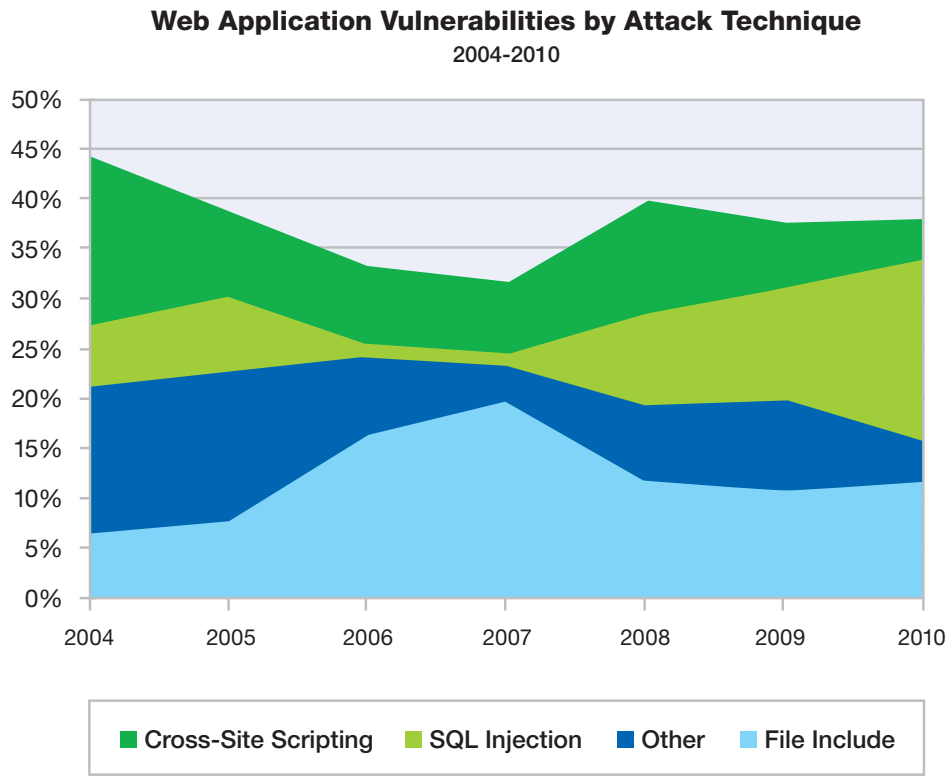


Figure 58: Web Application Vulnerabilities by Attack Technique – 2004-2010

Web application platforms vs. plug-ins

One important web application category is open source content management systems. These tools often find their way onto both internal and external corporate websites because they simplify the task of setting up a complex site, and there is a wide array of different plugins available for these platforms that can add all kinds of useful functionality. However, it is important for organizations that are running these platforms to be aware that the plugins typically have different developers who may not be as prompt at providing patches for security issues as the maintainers of the core platform itself.

Looking specifically at Drupal, Joomla!, Typo3, and Wordpress, there were six times the number of vulnerabilities disclosed in the plugins for these platforms than in the core platforms themselves during 2010. Only 41 percent of the vulnerabilities disclosed for these plugins had patches available. Patch promptness may vary widely from one plugin developer to the next, so it is important that users of these CMS systems examine vulnerability disclosure and patch promptness associated with the specific plugins that they intend to use.

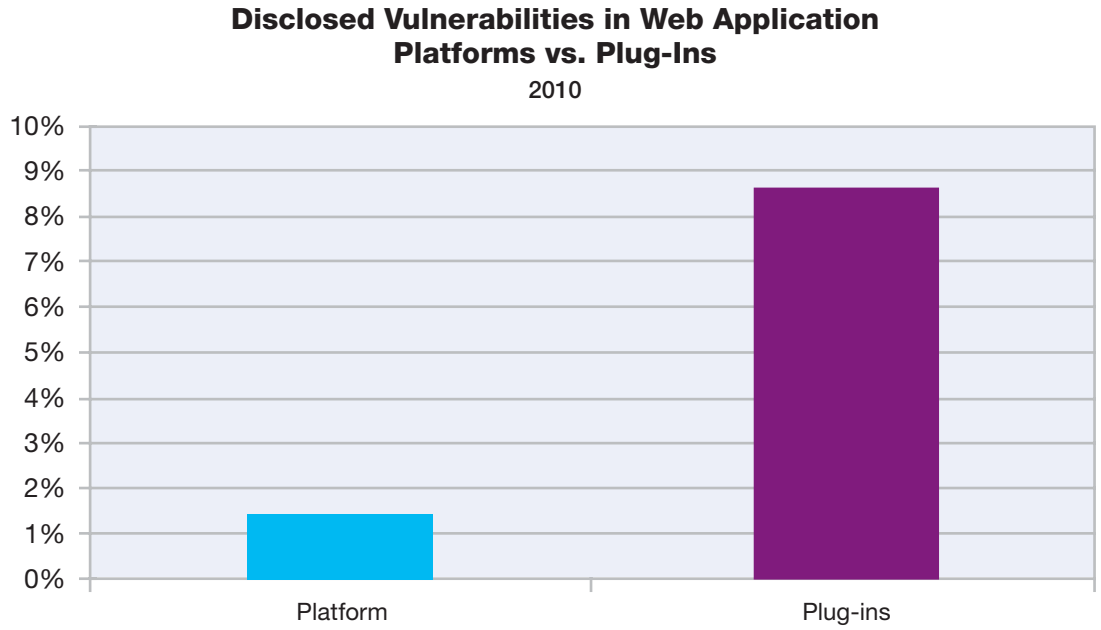


Figure 59: Disclosed Vulnerabilities in Web Application Platforms vs. Plug-Ins – 2010

Client-side vulnerabilities and exploits

The bad guys know that web application vulnerabilities are plentiful and our managed security services reports large volumes of attack activity targeting them. SQL Injection is a particularly significant risk as these attacks are sometimes launched in order to gain a foot hold within corporate networks from the Internet. SQL Injection vulnerabilities in external web applications can sometimes be exploited to gain code execution privileges on a database server in the Demilitarized Zone (DMZ). Once an attacker has hopped onto this lily pad, access to the internal network may be just another exploit away, often facilitated by data replication between DMZ databases and databases on the inside.

SQL Injection vulnerabilities can also used to manipulate the content of websites. Attackers take advantage of this capability to insert code into legitimate websites, redirecting visitors to malicious sites. These malicious sites usually host exploits that target the victim's web browser and the browser environment, often using automated exploit toolkits that obfuscate their attack payloads.

The browser and the browser environment have been primary targets for attack activity for several years. In previous trend reports, X-Force observed a decrease over time in the total number of high and critical vulnerabilities in this client environment, particularly due to a substantial decrease in the volume of vulnerable ActiveX controls that were being discovered. We interpreted this trend positively, as it seemed that some of the low hanging fruit client side vulnerabilities had been plucked and we seemed to be progressing toward a future in which substantially fewer client vulnerabilities remained for attackers to target. Unfortunately, this trend seems to have reversed in 2010, meaning that the promise of a future with few client side vulnerability disclosures is further out than we originally hoped.

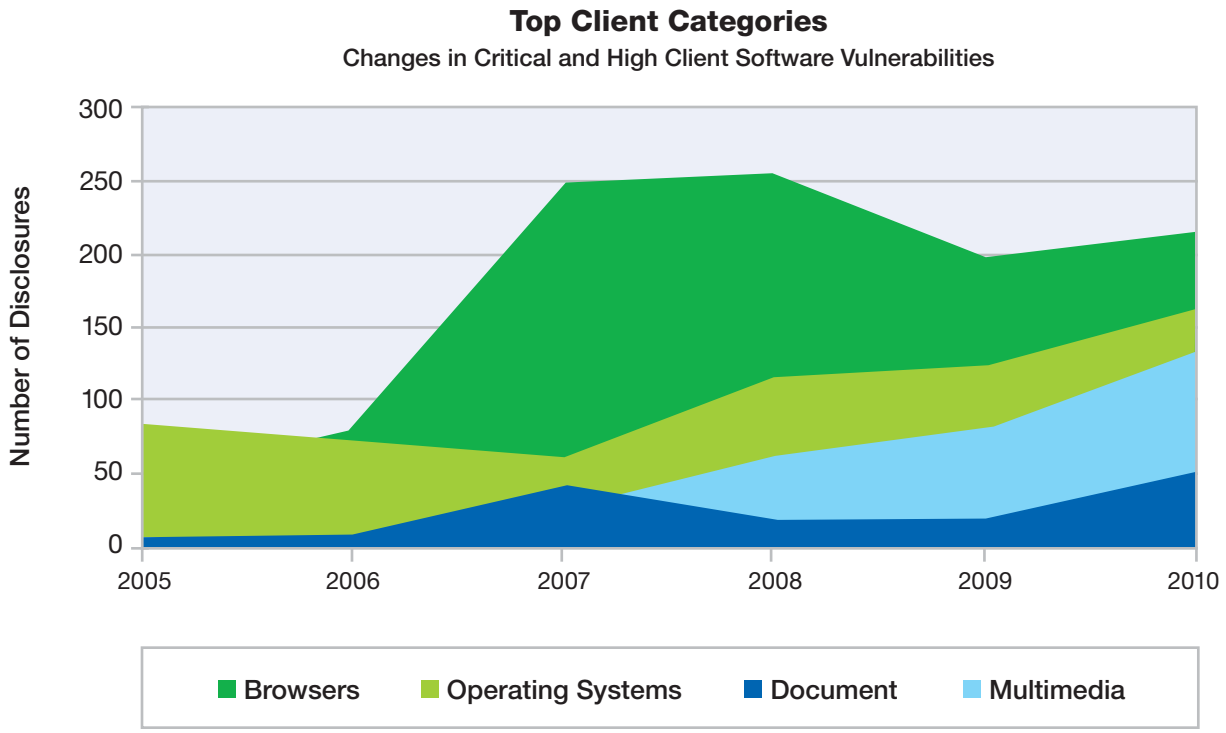


Figure 60: Top Client Categories – Changes in Critical and High Client Software Vulnerabilities

The total number of high and critical browser vulnerability disclosures has leveled off since 2009, but 2010 saw an increase in the volume of disclosures in document readers and editors as well as multimedia players (particularly Flash) and Java. Many of these vulnerabilities have been subjected to attack activity in the wild. X-Force believes that these formats are targeted in part because the browser market has become more competitive. A vulnerability in a particular browser may only successfully exploit a percentage of the potential victims who visit a malicious website. Popular document and multimedia viewers have more universal market penetration and malicious code can reach them regardless of what browser is being used by the victim. Furthermore, document readers can also be targeted over email. Malicious email attachments were exploited in 2010 through mass spam attacks, as well as in cases of sophisticated, targeted spear phishing, sometimes with zero-day vulnerabilities.

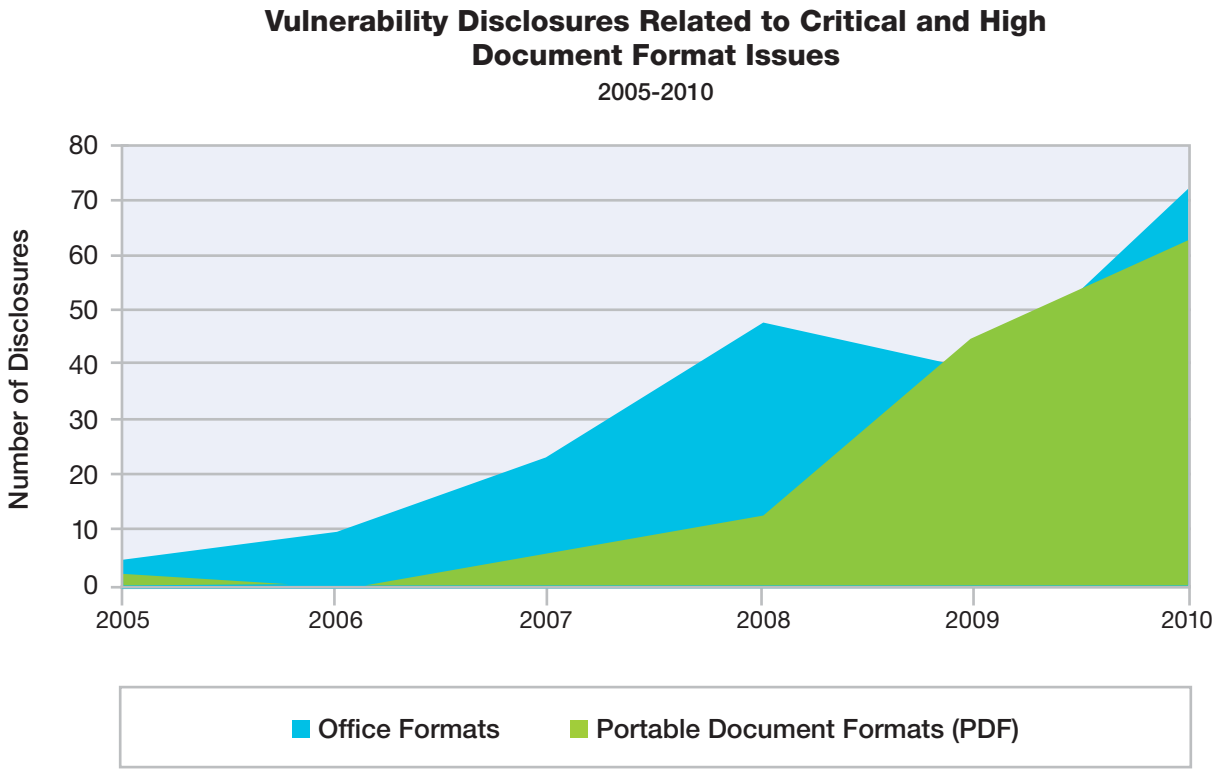


Figure 61: Vulnerability Disclosures Related to Critical and High Document Format Issues – 2005-2010

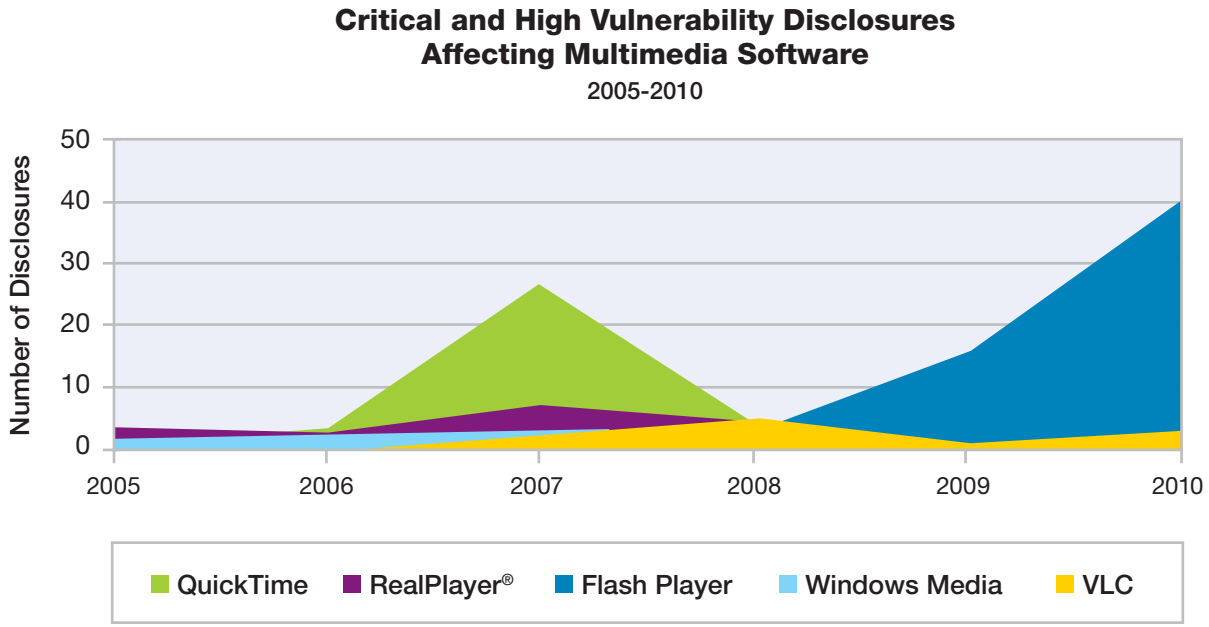


Figure 62: Critical and High Vulnerability Disclosures Affecting Multimedia Software – 2005-2010

Exploit effort versus potential reward matrix

When particularly critical vulnerabilities are disclosed, X-Force typically issues an alert. In some cases this is coupled with out of band coverage in our products. **X-Force issues alerts** in cases where we think that the risk of widespread exploitation for a vulnerability is particularly high. In the second half of 2010, X-Force issued 34 alerts and advisories, 19 of which were eventually subject to public exploit releases. Predicting the future is not an exact science. In the 2008 year end trend report, we introduced a model that helps explain how we decide which vulnerabilities are likely to see widespread exploitation. This model is called the "Exploit effort versus potential reward matrix."

The exploit effort versus potential reward matrix functions by attempting to chart the opportunity that each vulnerability represents to attackers from a financial perspective. On the X (horizontal) axis we chart the estimated effort associated with exploiting a vulnerability. Vulnerabilities that fit readily into the existing model that attackers have for breaking into computer systems and harvesting data from them score high on this dimension. Vulnerabilities that are hard to exploit or which require development of new business models around them, score low. On the Y (vertical) axis, we chart the overall opportunity that a vulnerability represents to attackers who do exploit it—how much value can be extracted out of exploiting this vulnerability.

A chart of these two axes breaks out into four quadrants. The first quadrant (in the upper right) represents vulnerabilities that are relatively inexpensive to exploit and represent a large opportunity to attackers. These are exactly the sort of vulnerabilities that are likely to see widespread exploitation in the wild. The second quadrant (in the upper left) represents vulnerabilities that are high value but harder to exploit—cases which may be targeted by sophisticated attackers. The third quadrant (in the lower left) represents low value, high effort vulnerabilities that are unlikely to be targeted widely. The fourth quadrant (in the lower right) represents lower value, lower effort vulnerabilities which are sometimes targeted if it is sufficiently easy for attackers to do so.

Figure 63: Exploit Effort vs. Potential Reward

In the previous chart (page 88),we show in which of the four quadrants we categorize the 34 vulnerabilities that X-Force released alerts and advisories for in the second half of 2010. For obvious reasons most of these fall into the first quadrant. All but one of the 25 vulnerabilities in the first quadrant are vulnerabilities in the browser, the browser environment, or in email clients. As we have discussed before, this software area is a popular target for exploitation. In some of these cases, exploits emerged before patches were available from the vendors. The only vulnerability in this category that is not a browser or email client side issue is the LNK file vulnerability (**CVE-2010-2568**) that **the Stuxnet worm used to exploit computers via malicious USB keys.**

Most of the seven vulnerabilities classified as high value, but hard to exploit, can only be targeted by attackers with some special knowledge or access, or were hard to exploit for technical reasons. One interesting exception was a cryptographic attack against ASP.NET (**CVE-2010-3332**) that could be used to predict cookie values and gain access to web applications without permission. We think this is a very serious vulnerability and tools have been publicly disseminated that can be used to exploit it. So why do we place it in the second quadrant? The reason is that it represents an unusual kind of vulnerability. Attackers are used to the regular disclosure of client side vulnerabilities that they can plug into their exploit tool kits. They have highly developed processes for taking those kinds of vulnerabilities and turning them into cash. But this vulnerability represents a new kind of attack vector. Cryptographic attacks against cookie values do not surface frequently. Basically, the challenge for the bad guys is that they would have to think outside of the box in order to adopt and use this vulnerability. Although we believe that this vulnerability is being exploited maliciously, the volume of activity that we associate with more mainstream issues is likely not going to be there. It takes time for a new attack methodology to take hold and become popular. Hopefully most people will have patched this particular vulnerability long before then.

The two vulnerabilities (**CVE-2010-3229** and **CVE-2010-2742**) classified as low value, low cost are both remote Denial of Service issues. Neither was subject to public exploit disclosure.

Key Recommendations

Looking at all of this vulnerability disclosure data holistically, three key recommendations for network administrators stand out:

1. **Web Application vulnerabilities represent a significant risk to the modern enterprise**
   Large quantities of web application vulnerabilities are being disclosed and attackers are actively targeting these vulnerabilities wherever they can be found. IBM believes in a total lifecycle approach to managing web application security, from design, to development, to testing, to operational deployment. An array of tools and processes should be employed, from vulnerability assessment at development time to protection in production with web application firewalling or intrusion prevention. Pay close attention, in particular, to third party web applications that may be running in your environment, such as open source content management systems and their associated plugins. It is also important that vulnerabilities in these systems are patched.
2. **Client side vulnerabilities are a favorite target for attackers**
   It is important that software on client machines stays patched up to date—particularly browser and browser related software such as multimedia players, and document viewers and editors. The bulk of malicious activity on the Internet targets this kind of software.
3. **Patching is not enough**
   Some vulnerabilities are disclosed and are exploited before a patch is available. Although we want the window between disclosure and patch to be as short as possible, it will always exist. Sometimes long time frames are unavoidable. This means that other mitigation strategies are needed, including network IPS, as well as the ability to automatically deploy workarounds and mitigations during the window of time that a fix is unavailable.

Virtualization—risks and recommendations

Virtualization systems have been growing in importance. Their increasing deployment across network infrastructures makes it important to understand the security concerns surrounding them, as there is danger in deploying any new technology before its security issues are well understood. In the IBM X-Force 2010 Mid-Year Trend and Risk Report we investigated different security vulnerabilities that had been disclosed in Virtualization technology. This section continues the discussion by looking at the different security issues these vulnerabilities relate to and providing recommendations for managing them.

We begin by describing the various components of virtualization systems and the security issues surrounding them. Next, we describe some new types of attacks that are unique to virtualization systems. We then provide a description of public exploits that have been published for virtualization systems, illustrating that the risk against these systems is real. Finally, we summarize virtualization system security concerns and provide recommendations for operating virtualization securely.

Virtualization system components

To understand how to secure virtualization systems, it is necessary to understand their components, and vulnerabilities and configuration issues associated with each of them. Figure 64 shows the components of a typical virtualization system. Each of these components has been subject to computer security vulnerability disclosures.

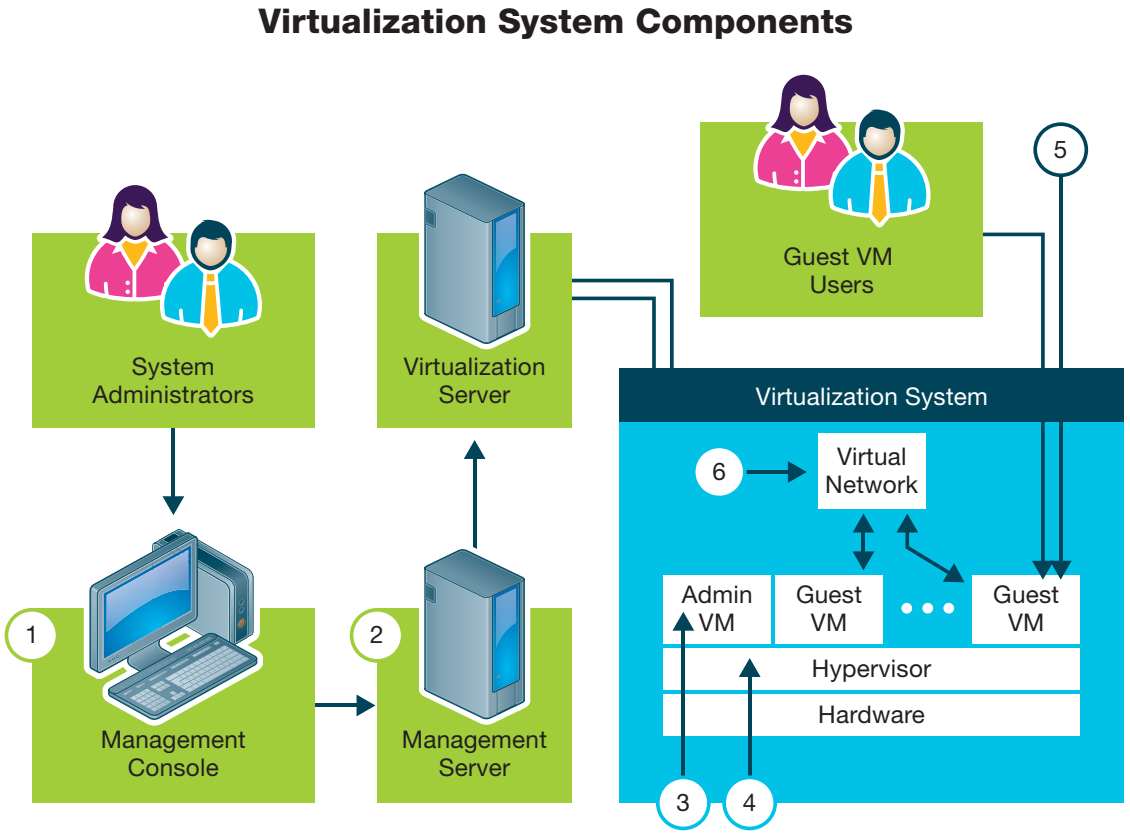


Figure 64: Virtualization System Components

1. Management console

The management console is the application used by system administrators to configure the virtualization system. It may be either a web browser using a web application, or a custom console.

Management console vulnerabilities

Some vulnerabilities disclosed in management consoles can divulge password information or allow attackers to gain access to the management server without logging in. Others allow an attacker to execute code within the context of the web browser or to redirect configuration requests to other management servers.

2. Management server

The management server is the component that stores configuration information. It is configured via the management console and interacts with the virtualization system to provide configuration information.

Management server vulnerabilities

Vulnerabilities have been disclosed in management servers that allow local users logged into the management server to gain elevated privileges or to execute arbitrary code on the management server.

3. Administrative VM

The administrative VM is a special virtual machine that exposes network services to the management server for configuring the virtualization system. It receives configuration information from the management server and implements the configuration by communicating with other elements of the virtualization system.

Administrative VM vulnerabilities

A number of different types of vulnerabilities have been disclosed in administrative VMs. Some allow a Denial of Service either by halting the system or crashing the administrative VM. Others allow attackers to obtain passwords stored in the administrative VM. Still others allow an attacker to exploit the network services exposed by the administrative VM to cause buffer overflows that allow arbitrary code to be executed, to gain elevated privileges, or to bypass authentication altogether.

4. Hypervisor

The hypervisor is the operating system of the virtualization system. It runs directly on the hardware and provides the substrate on top of which the virtual machines run.

Hypervisor vulnerabilities

Disclosed hypervisor vulnerabilities either allow an attacker to cause a Denial of Service by crashing the hypervisor or to violate the isolation of guest VMs by allowing one guest VM to access another without communicating across the virtual network. This latter type of vulnerability is known as hypervisor escape vulnerability.

5. Guest VMs

Guest virtual machines provide the operating environment within which virtual servers run. Like physical servers, they are configured by installing operating systems and applications on them. The hypervisor isolates virtual machines from one another so they can communicate only through the virtual network.

Guest VM vulnerabilities

One type of vulnerability disclosed in guest machines allows an attacker who is logged into the machine to gain elevated privileges. Others allow an attacker to crash the virtual machine or truncate arbitrary files on the guest VM. A final class of vulnerability allows an attacker to remotely exploit buffer overflow vulnerabilities to execute arbitrary code on the guest VM.

6. Virtual network

The virtual network is the network implemented within the virtualization server through which guest VMs communicate with one another without going across a physical network The topology of a virtual network is defined through virtual switches that are established through the configuration of the virtualization system and through virtual firewalls that are installed as special-purpose VMs.

Virtual network vulnerabilities

Vulnerabilities have been disclosed in workstation virtualization products that impact virtual network infrastructure components such as DHCP servers that run within the virtual network.

Vulnerability distribution

It is instructive to examine the distribution of disclosed vulnerabilities in the various virtualization system components, as this provides a picture of the risks they involve. In our **Mid-year 2010 X-Force Trend Report** we analyzed vulnerabilities that were disclosed between 1999 and 2009 in virtualization products from Citrix, IBM, Linux VServer, LxCenter, Microsoft, Oracle, Parallels, RedHat, and VMware. In Figure 65 we categorize the vulnerabilities from that report that impacted server class virtualization products. These production class products are intended to be used in operational IT environments and usually have "Type 1" hypervisors, as opposed to workstation class virtualization products that usually have "Type 2" hypervisors. This data represents vulnerabilities disclosed in the vendor's code, as opposed to third-party components. It is difficult to classify vulnerabilities in third-party components because it is usually not clear where these components are used within virtualization systems based on vulnerability advisories. This data encompasses a total of 80 vulnerabilities.

Of particular note here are the first two classes of vulnerabilities. The most common class of vulnerabilities in server class virtualization products, hypervisor escape vulnerabilities, generally represents the most serious risk to virtualization systems as these vulnerabilities violate the principal of isolation of virtual machines. The next largest class of vulnerabilities, administrative VM vulnerabilities, also present serious risk, as these can provide control over the configuration of the entire virtualization system.

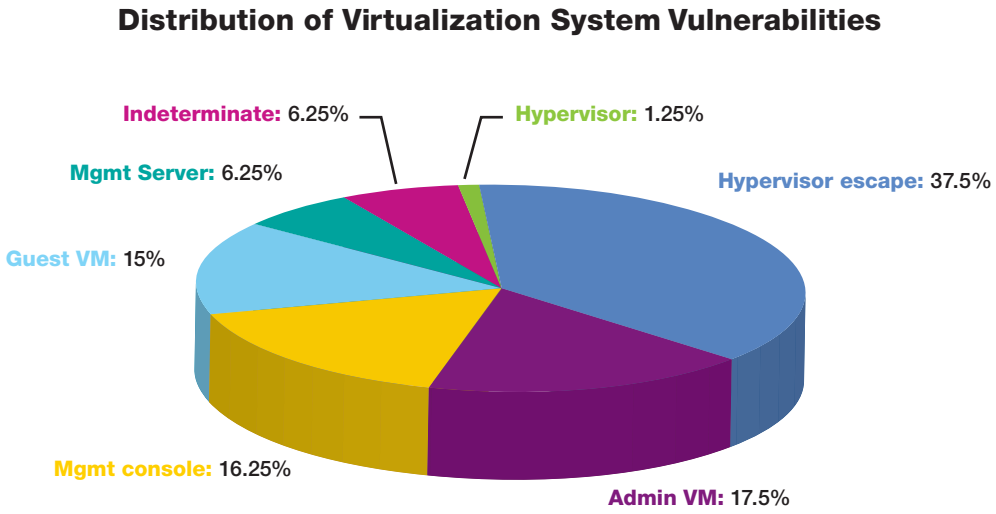


Figure 65: Distribution of Virtualization System Vulnerabilities

Attacks unique to virtualization systems

A number of attacks are unique to virtualization systems, and so represent new types of risk to network infrastructure. One such attack is VM jumping or guest hopping, which allows one virtual server to access another without going across the virtual network by exploiting hypervisor escape vulnerabilities. Other types of attacks affect virtual machine images. They can be modified during deployment and duplication, can be deleted to effect a Denial of Service attack, and can be modified on disk to inject code or files into the virtual file structure.

A third type of attack affects VM migration, a feature that allows a running guest VM to be transferred from one virtualization server to another with very little downtime (on the order of a few seconds). There are several virtualization products that implement this feature, whose purpose is to provide high availability and load balancing.

John Oberheide at the University of Michigan has demonstrated that if the communication channel between virtualization servers is not encrypted, it is possible to execute a man-in-the-middle attack that enables an attacker to change arbitrary state (in files or processes) in a VM when it is migrated. The operation of this attack is illustrated in Figure 66.

To affect this attack, the attacker must first insert a process under his control in the communication path between the two virtualization servers. This could be done by compromising a router, or by changing the configuration of an intervening switch to insert a machine under the attacker's control. If best practices are followed, VM migration occurs over a dedicated network that is hard to compromise, but unfortunately real world systems are not always deployed in an ideal way. Once established the attacker's process can observe VMs being migrated from one virtualization system to another, and modify the state of a migrating VM via the process under his control.

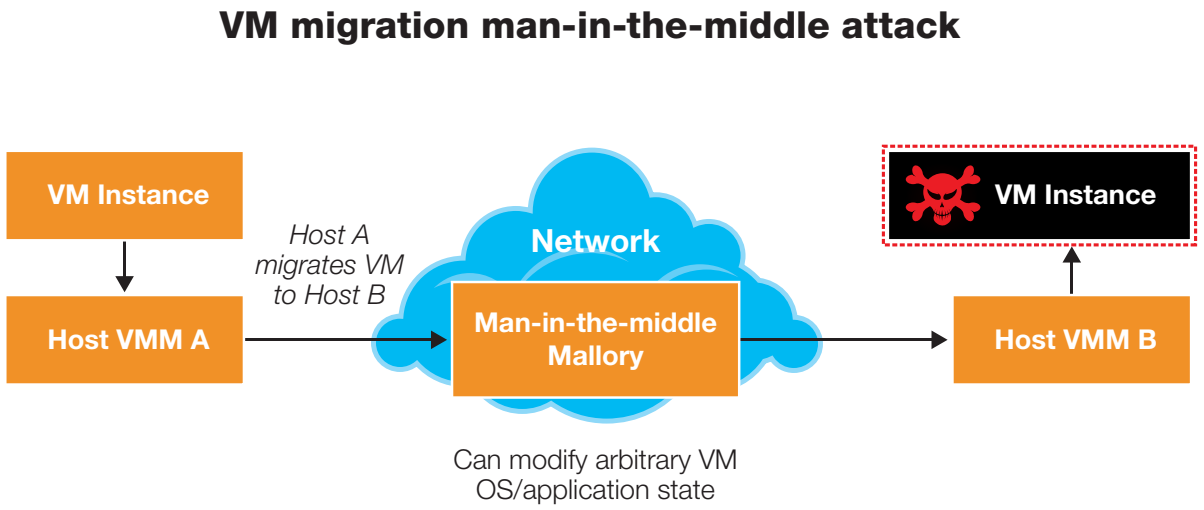


Figure 66: VM migration man-in-the-middle attack

Public exploits

Not all vulnerabilities have been shown to be vulnerable to attack, so it is significant when exploits have been published that demonstrate attacks against specific vulnerabilities. Thirty-six such exploits are known for virtualization system vulnerabilities. Most of these attacks are against third-party software that is used by vendors in implementing their systems, rather than in vendor-developed code. A few examples are given below.

- **CVE-2009-2267.** This vulnerability allows a guest OS user to gain elevated privileges on a guest OS by exploiting a bug in the handling of page faults. It affects ESX server 4 and other VMware products. A binary executable for exploiting this vulnerability has been posted at lists.grok.org.uk.
- **CVE-2009-3760.** This vulnerability allows a remote attacker to write PHP code (the scripting code used to implement web server functionality) into a web server configuration script and then to take advantage of this change to execute commands with the privilege of the server. This vulnerability affects the XenCenter web server. It is exploited by sending specially crafted URLs to the server, which has been published in a Neophasis post.
- **CVE-2007-5135.** This is an OpenSSL buffer overflow that enables an attacker to crash a service on VMWare ESX server 3.5, presumably in the administrative VM. This is a good example of a vulnerability in a third party component. Although this has not been demonstrated, it is possible that the vulnerability could allow an attacker to execute arbitrary code on the system. The attack involves sending multiple ciphers to the OpenSSL service to exploit a bug in the cipher processing code. A Neophasis post explains how this vulnerability might be exploited.

Summary of security concerns

The forgoing discussion raises a number of security concerns related to virtualization systems.

1. Virtualization systems added 373 new vulnerabilities to the network infrastructure in the period between 1999 and 2009.
2. A number of public exploits exist that demonstrate the risk from virtualization system vulnerabilities is real.
3. Contrary to the perception of some, virtualization systems don't add any inherent security, because the same connectivity is needed as between servers on physical networks.
4. The addition of new components to the network infrastructure provides new targets of attack.
5. Some entirely new types of attacks are introduced due to the nature of virtualization systems.
6. Migration of VMs for load balancing can make them more difficult to secure, because they move from one execution environment to another.
7. The ease of addition of new VMs to the infrastructure can increase the likelihood that insecure systems will go online.

Operating Secure Virtual Infrastructure

Keeping in mind all of the aforementioned security concerns, of course it is important to acknowledge that the value of virtualization can far outweigh the new risks that it introduces in most cases. However, we should approach the deployment of virtualization with an understanding that smart practices in terms of the configuration and management of these systems can help reduce the security risks. To that end we include a discussion of our configuration recommendations for each component of the virtual infrastructure.

Management server

Management servers should be treated like application servers; they should be segregated from operational networks by appropriately configured firewalls and routers. To help protect management system databases, you should restrict their access to the management server, a database administrator, and backup software. You should limit access to remote management tools and use accounts with limited privileges. Finally, all communications with the management server should be authenticated and encrypted, and use logging to track the operations performed on the management server.

Administrative VM

Avoid installing third-party software on administrative VMs, as this can violate the vendors' hardening of their systems and introduces unnecessary risk. You should scan your systems to discover all exposed network services and disable or reduce access to those you don't need. To aid in log analysis, synchronize clocks on virtualization servers and management servers and manage log size to avoid filling partitions. It's also a good idea to implement file integrity checking and password policies, disable root logins, and only allow server administrators to manage administrative VMs.

Hypervisor

The measures available to help protect the hypervisor are limited—install hypervisor updates and patches as soon as they are available.

Guest VMs

Virtual servers running in guest VMs should be hardened just like physical servers. You should update and patch their operating systems. Use single-role servers and disable unnecessary services. You should use a local firewall to insure limited host control and use limited scope administrative accounts with strong passwords. You should also protect files on your virtual servers—use access control lists, use encryption if possible, and audit file operations such as access, creation, and deletion. Finally, there are a couple of measures that are unique to virtual servers. You can disable virtual devices that are unused and use hardened server images as the basis for new VMs. For example, VMware supports the definition of templates that can be used for the creation of new VM images.

Virtual network

There are a number of measures you should take to protect your virtual networks. If possible, you should install VMs with different security profiles on different physical virtualization servers. This is advised because of the existence of hypervisor escape vulnerabilities that enable one virtual server to affect other virtual servers running on the same virtualization server without communicating over the virtual network. Failing this measure, you should at least use virtual firewalls between groups of machines with different security postures. You should also isolate VM traffic by defining VLAN port groups in virtual switches and associating each VM virtual adapter with the appropriate port group. If supported, you should configure port groups to prevent virtual adapters from entering promiscuous mode and to prevent virtual NICs from changing their MAC addresses.

Endpoint security and systems management

In 2010 there was no slowdown in the frequency or velocity of conditions that can lead to compromised systems. In the first half of 2010, reported vulnerabilities were at an all-time high at 4,396, of which 94 percent were remotely exploitable. The full year total of reported vulnerabilities for 2010 reached 8,562.

Although vendors typically have been diligent in providing patches, at least 44 percent of all vulnerabilities in 2010 had no corresponding patch. Compounding the problem is that alternative methods of mitigating an exposure, such as disabling certain services or modifying the system registry, can often be a time-consuming and error-prone task across today's highly complex and distributed computing environments.

Malware has become more sophisticated as well, using blended techniques, stealth, evasion, and polymorphism to impact the ability to detect and prevent compromise, in many cases including targeted techniques to counter traditional endpoint security solutions.

In **June of 2010 Stuxnet** appeared and was called the most sophisticated malware ever discovered. Not only did it employ close to a dozen individual executables, it exercised almost as many different propagation methods. What made Stuxnet so insidious was its targeting of physical Supervisory Control and Data Acquisition (SCADA) systems.

Stuxnet is an especially troubling incident because it is a proof of concept for what a well-organized group can accomplish in a fairly short amount of time to compromise command and control systems and modify programmable logic controllers used in many industrial processes, including nuclear plants.

Even though we are experiencing an increase in highly sophisticated and stealthy malware, in the majority of cases, including Stuxnet, the mechanisms used to initially compromise a device still tend to exploit misconfigured, poorly administered, and unpatched systems.

A well-managed device is a more secure device

The same methods and controls we have known about and have been available for decades are the same methods most organizations struggle with effectively implementing. Basic device management hygiene is elusive for most, but it is still one of the most effective methods for maintaining resiliency in your computing environment.

Basic device management hygiene should include

1. Real-time asset inventory and configuration information for all devices, regardless of location.
2. Installed, running, and up-to-date anti-virus, and other endpoint security technologies.
3. Patching early and often.
4. Defining and enforcing security configuration policies including:
   a. OS, application, data, and user settings
   b. Removable media access
   c. Firewall configuration
   d. File and print sharing
   e. Asset and configuration inventories
5. Educating and empowering users on corporate use policies and changes in the threat environment.
6. The ability to monitor the computing environment and quickly identify any deviations from normal operating state, system compromise, or failure.

Although we are seeing increasingly sophisticated attacks, the reality is that most attackers take advantage of our inability to practice basic device management hygiene. The attackers may be getting smarter, but it doesn't take a genius to take the path of least resistance.

Case study one: large technology company

Environment: Tens of thousands of end-user computing devices located across three major geographies in North America, Asia Pacific, and Europe, but managed centrally from the Eastern United States.

Problem: Malware outbreaks had been increasing significantly, especially in geographies with less IT control. Clean-up, technical support, and administrative costs were increasing and the situation was becoming untenable.

Approach: Forensic analysis suggested that the majority of these malware outbreaks were initially compromising the systems through fairly standard methods of exploiting misconfigured or poorly administered systems, including unpatched systems.

The company had a defined security configuration policy, but was struggling to ensure compliance across their global deployment. They decided to deploy a security configuration technology to implement the operational controls needed to enforce configuration compliance.

They chose a control group, which included devices from all geographies, and implemented the security configuration technology and tracked the malware outbreaks over the course of 3 quarters. The control group showed an 80 percent reduction in malware outbreaks compared to other groups, even though many of these systems were located in regions that historically been quite susceptible to attack.

Summary: Eliminating the most common attack vectors—many common configuration and administrative errors—in end-user computing devices was an effective approach in limiting successful compromise.

Case study two: public sector

Environment: 5,000 end-user computing devices located throughout North America.

Problem: In April of 2008 several dozen computers were exhibiting strange behavior, including running port scans against the network and periodically rebooting. It was determined that they had been infected with a new polymorphic virus, which was rapidly spreading to other computers. There were no AV signatures available.

Approach: As there were no signatures available and it wasn't clear all the propagation methods the virus would use, the organization made a decision to quarantine the devices. The problem was that they were not sure which machines had been infected.

During the incident response process, it became clear that infected systems all shared a common characteristic. Using this information and their existing systems management tools they were able to identify infected machines with real-time configuration data across their 5,000 endpoints in less than five minutes. They forced these machines to auto-quarantine themselves from the network, which allowed them additional time to determine if the devices needed to be completely reimaged or if there was a method to clean the infection without further data loss.

Summary: Situational awareness into the state of all computing assets was able to dramatically limit the impact of a compromise when one does occur.

There is no silver bullet and the goal of security professionals isn't to guarantee 100% security against attack. Rather, the focus should be on eliminating attack vectors and limiting the impact when a compromise does occur. That goal requires coordination and a common language between the security and operational teams to manage and secure the computing environment.

The State of Affairs in DNSSEC

Introduction

DNSSEC[11] is the set of security extensions to the Domain Naming System (DNS) to perform verification and validation of received responses to DNS queries. When someone references a site such as xyzzy.test.com, they expect to get answers back regarding the Internet address where the servers are located on the Internet. In the past, we've simply relied on them and trusted them to be true. DNS is a highly distributed cloud (some would say fog) of servers relaying requests and responses back and forth and is a fundamental core protocol on which the Internet itself is highly dependent. Outside of certain limited, predefined transactions, the servers themselves have had no real trust mechanism between them, depending on a hierarchical tree of recursive queries and redirections to discover other servers with answers.

That trust in the relationships between servers and the integrity of the responses has proven to be ill founded at times. Over the last several years, name servers have come under attack through spoofing, where false information is deliberately fed into the stream of DNS responses. Once in the stream, these responses have been trusted as if they came from true authoritative sources. There has been no way to verify, end to end, the validity of the data returned by the DNS. Improvements in server software have made spoofing attacks more difficult, but have never completely eliminated the threat. This is what DNSSEC was designed to thwart.

DNSSEC has been under design by the Internet Engineering Task Force, IETF, for the last 15 years. The IETF itself only turned 25 in January of 2011. These extensions have been a long time in coming, and are finally beginning to arrive.

2010 The year in review

2010 opened with a whole new promise in DNSSEC.[13] Agreements had finally been reached for the signing of the root zone "." and initial testing was begun. The .gov global top level domain (gTLD) had been signed with a mandate that all the domains within .gov would also be signed by the beginning of 2010. Many, if not most, were. A number of the country code top level domains (ccTLDs) had also been signed.[14] The Public Interest Registry (PIR) began 2010 by testing signing the .org gTLD which they finalized and signed in mid-2010.

During the course of the year, the months of testing signatures of the root zone came to a successful conclusion and the root zone was formally signed in June with all 13 root name servers supporting the signed zone.

Software deployment and components

Most modern DNS implementations already support DNSSEC out of the box. Some older deployments of name servers certainly remain but, since the root has been signed and is serving up signatures, all servers actively on the net and handling DNS requests have proven at least compatible with DNSSEC and the kinks have been worked out at that level.

Bind version 9 has supported DNSSEC for many years and introduced the concept of a "Domain Look-aside Validation", DLV, service.[15] This service was intended to be a third party trust anchor to serve in the interim until the root was signed. Now that the root is signed, the DLVs still serve a need by providing a mechanism for domain owners to register their keys and have their zones validated until all the registries and registrars are fully up to speed and supporting DNSSEC.

Some popular caching forwarder servers for DNS, such as DNSmasq, still do not support DNSSEC and depend on the downstream servers for validation at this time. These servers can still handle DNSSEC requests and can also pass them upstream to requesters for validation. These cachers may not do validation themselves, but they do not interfere with the proper functioning of DNSSEC.

13 DNSSEC: DNS Security Extensions – http://www.dnssec.net/
14 DNSSEC Deployment – https://www.dnssec-deployment.org/
15 A Handy Table Showing the Status of TLD DNSSEC Deployment – https://www.dnssec-deployment.org/index.php/deployment-case-studies/a-handy-table-showing-the-status-of-tld-dnssec-deployment/
16 DNSSEC Look-aside Validation Registry – https://dlv.isc.org/about/using

Several packages—both commercial software and Open Source freeware—are on the market now for the domain holders to conveniently support DNSSEC. OpenDNSSEC[17] is one such Open Source package. This package allows near drop-in support of DNSSEC using a "bump on the wire" technique. Zone signings are handled automatically on a dedicated machine situated between an isolated primary authoritative name server and the slaves that service the requests from the outside Internet. This allows the basic DNS management to continue on with little change, but can require rearchitecting of some deployments that may not have conformed to best common practices in the past.

DNSSEC challenges and stumbling blocks

DNSSEC has now overcome some of the perceived major challenges—agreements over signing the root zone, getting the registries to sign their supported zones, and getting software available and deployed. However, in the upcoming years there will still be challenges that threaten to hold back the utilization and realization of the full benefit of DNSSEC. Most of the challenges now being faced are less apparent than the ones that have already been handled.

One overt problem on the provider side is in regards to the registrars. These organizations accept domain registrations for the registries, along with providing other value-add services within their business model. Domain registrations are inexpensive on a domain by domain basis, and very competitive. The registrars are depending on large volumes of domains with little manual work and highly automated processes. Throwing the issue of DNSSEC key registration into the mix threatens to complicate the registration process and drive up their cost of doing business. Even once they have the process automated, the chances are this may still drive up their support costs with little if any increase in revenues. It should come as little surprise that very few registrars have announced support for DNSSEC.[18] Even some which are said to be furthest along in their support plan are saying that they are still studying it and have no immediate plans for deployment.[19] In this environment, the only choices for the domain holder who wishes to sign his zone and support DNSSEC may be to change to one of the very few registrars supporting DNSSEC or to continue to participate in a DLV service such as that at the Internet Software Consortium (ISC).

Key management itself is going to add some burden on IT staff and operational procedures should be put into place.[20] Zones should be re-signed and verified periodically. The idea of updating a zone every few weeks, whether it has changed or not, just to freshen up the signatures may not sit well with some IT departments. Packages such as OpenDNSSEC can alleviate this to some extent by separating out the zone signing from the actual zone management. The zones records are signed by zone signing keys (zsks) and that can be largely automated. But the zone signing keys are signed by key signing keys (ksks). It is these keys, the ksks, which are registered with the registrars or a DLV and should be rotated or updated on a yearly basis. This is difficult to automate on the domain holder's side and likely to be a source of support problems on the registrar's side, even if they manage to automate the process.

In spite of the momentum in favor of DNSSEC, there is still some dispute and disagreement. Some services, such as OpenDNS (a large and popular DNS service provider) have indicated they have no intention of participating in DNSSEC, preferring instead to deploy DNSCurve, a competing protocol.[21]

17 OpenDNSSEC – http://www.opendnssec.org/
18 Public Interest Registry (.org) listing of Registrars and DNSSEC – http://www.pir.org/get/registrars
19 Domain registrars lagging behind over DNSSEC security – http://news.techworld.com/security/3218219/domain-registrars-lagging-behind-over-dnssec-security
20 Five Strategies for Flawless DNSSEC Key Management and Rollover – http://www.securityweek.com/five-strategies-flawless-dnssec-key-management-and-rollover
21 OpenDNS adopts DNSCurve – http://blog.opendns.com/2010/02/23/opendns-dnscurve/

They argue that their DNS resolvers already support DNSCurve and use it whenever possible. But, they then go on with the caveat that, "Of course, authoritative servers need to be upgraded to support DNSCurve as well..." which is something that has not happened. But the debate continues to hold back deployment.

There are also vexing problems on the consumer or client side of the DNSSEC issue. Some ISPs have indicated that they will not be providing DNSSEC validation because they have seen no demand for it.[22] This is hardly surprising, since the attacks have been few and DNSSEC is designed to be largely transparent to the end consumer. There doesn't seem to be a pain point on the consumer side to help push adoption. This should be a simple thing to provide. It's just a matter of enabling an option in the caching name servers, which most installations should now support.

For zones that are not signed, this situation results in no increase in server load and has no impact at all. If a domain is signed, those signatures can be checked and the failing records dropped. The end consumer may not even see that something has happened to help protect him. He might not even know if it is not checked and he does get trapped by some attack. This transparency problem has the associated problem of creating a lack of demand for a feature that really should be just there. Lacking some mandate from the registry authorities to the ISPs, this may be difficult to overcome.

This applies equally, if not more so, to change-adverse IT departments in corporations that are unwilling to make changes even if it is "doing the right thing." The risk, no matter how minuscule, of causing something to break due to a change which isn't going to provide them with some overt, observable, benefit can create reluctance in a corporate environment.

What's ahead now

Now that the root zone has been signed and more and more of the TLD zones are being signed, we can expect to see more progress in the coming years and some new and fresh ideas for taking advantage of DNSSEC. Already, proposals have been put forth to add e-commerce certificate hashes in DNS to firmly tie a certificate to the domain holder through the use of DNSSEC to sign those records.[23] While this is certainly no substitute for a Certifying Authority, it helps build trust in the authenticity and reliability that sites are what they say they are. There have also been proposals and discussions for IPsec keys and information to be overloaded into the DNS and authenticated by DNSSEC to facilitate opportunistic encryption and VPNs. These are desirable features that make DNSSEC more valuable but must wait for DNSSEC-aware applications.

It's important to keep in mind that DNSSEC was designed to deal with one particular threat, that of DNS spoofing and falsified DNS data. It is not the be all and end all of DNS security. But it has a valuable role to play. And, as we can learn from some of the proposed uses for DNSSEC, there may yet be other uses to which DNS with DNSSEC may be applied.

Conclusion

2010 was a watershed year for DNSSEC with many important milestones passed. Some would argue that DNSSEC has finally reached a critical mass and momentum is building behind it. But, in spite of all the good press, DNSSEC still has a long way to go in achieving true end-to-end validation of the Domain Naming System.

22 DNSSEC Deployment Among ISPs – The Why, How, and What – http://www.circleid.com/posts/20100629_dnssec_deployment_among_isps_the_why_how_and_what/
23 Dan Kaminsky's "The DNSSEC Diaries" – http://dankaminsky.com/2010/12/13/dnssec-ch1/

Section III—Developing Secure Software

In Developing Secure Software, we present data surrounding proven processes and techniques for developing secure software. We discuss how enterprises can find existing vulnerabilities and help prevent new ones from being introduced. If you use networked or web applications to collect or exchange sensitive data, your job as a security professional is harder now than ever before. We take a look at both the static and dynamic security testing done by the Rational® AppScan® group in all stages of application development and share insights on what was discovered.

Further analysis on web application trends

IBM Rational Security and Compliance provides further analysis on web Application Security trends in this year's report in two different ways. Continuing from its 2009 research, IBM® Rational® AppScan® onDemand Premium Service derives trends on web application vulnerabilities from 2010 assessment data. Additionally for this year's report, new automated technologies in the IBM® Rational® AppScan® portfolio are able to provide visibility to an organizational blind spot regarding web Application vulnerabilities.

Conclusions from real-world web application assessments

Methodology

IBM has collated real-world vulnerability data from hundreds of security tests conducted in 2010 from the IBM® Rational® AppScan® OnDemand Premium Service. This service combines application security assessment results obtained from IBM® Rational® AppScan® with manual security testing and verification. In all cases, identified false positives were removed from the results and the remaining vulnerabilities were categorized into the following key security categories:

- Cross-site request forgery
- Cross-site scripting
- Error message information leak
- Improper access control
- Improper application deployment
- Improper use of SSL
- Inadequate or poor input control
- Information disclosure
- Insufficient web server configuration
- Non-standard encryption
- SQL injection

For each of these categories, two core metrics were calculated:

1. The percent chance of finding at least one vulnerability in that category.
2. The average number of vulnerabilities that are likely to be found in that category.

Having collated similar data since 2007, it was also possible to trend this data over the past four years. In 2010 additional metrics were also captured for each test data point to gain deeper analysis of the data. This included the following areas.

Business Segment where test data was attributed to belong to one of the following:

- Financials
- Industrials
- Information technology
- Logistics
- Retail
- Other

Application Security Test Cycle depicting the type of test the application was involved in:

- One-time assessment—Applications tested for the first time
- Quarterly assessment—Applications tested in a regular, ongoing basis
- Retest—Follow-up test to confirm the findings (typically from the one-time assessment)

Application Technology depicting the main technology used to develop the application:

- ASP.NET application
- Java-based applications (including JSP)
- PHP-based applications

2007—2010 Application vulnerability trends

Several conclusions can be derived from our application assessment data, many of which indicate trends in the susceptibility of websites to these vulnerabilities. Since we started recording application security statistics in 2007 we have seen a steady decline in the instances of cross-site scripting (XSS) while, at the same time, cross-site request forgery (CSRF) has increased. In 2010, for the first time, we now find that CSRF is more likely to be found in our testing than XSS.

This change is attributed to better detection techniques for CSRF and also a greater awareness of the risk. We find that some organizations tolerate having some outstanding issues with CSRF if the risk of exploitation is minimized. This is generally not the case with XSS and these issues are often quickly resolved.

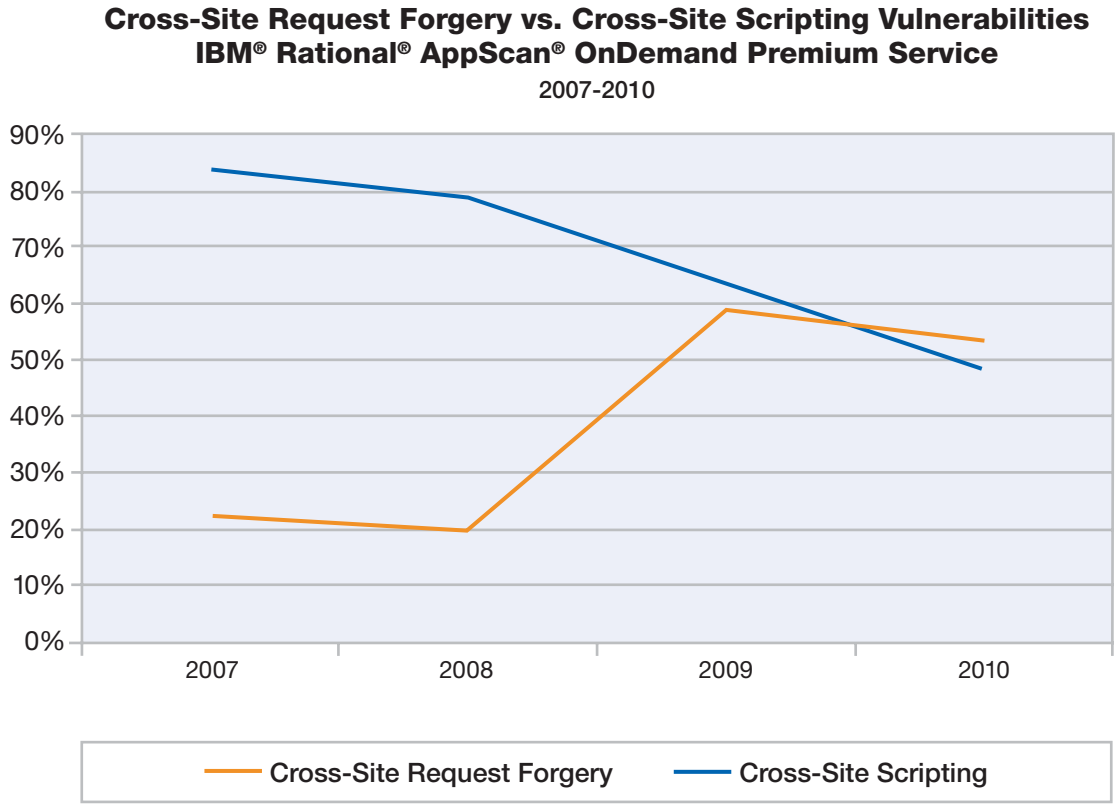


Figure 67: Cross-Site Request Forgery vs. Cross-Site Scripting Vulnerabilities IBM® Rational® AppScan® OnDemand Premium Service – 2007-2010

The true risk from CSRF is dependent on the specific application transaction that is vulnerable. It can be a simple search page or a more volatile money transfer transaction. As a consequence, we find that each instance of CSRF should be fully investigated. In the cases where it is a search page, the business may choose to accept this or put it on a slower track for mitigation.

XSS and SQL injection are both attributed directly to a lack of input control in code. Although we are seeing that instances relating to input control are on the decline, it is not as steady as XSS. We still find it present in our testing in excess of 60 percent of the time. SQL injection instances increased slightly in 2010, but are still down considerably from the numbers we had in 2007. Our data suggests that better database controls and methods appear to be the main reason for the decline, rather than any specific improvement in the lack of input control.

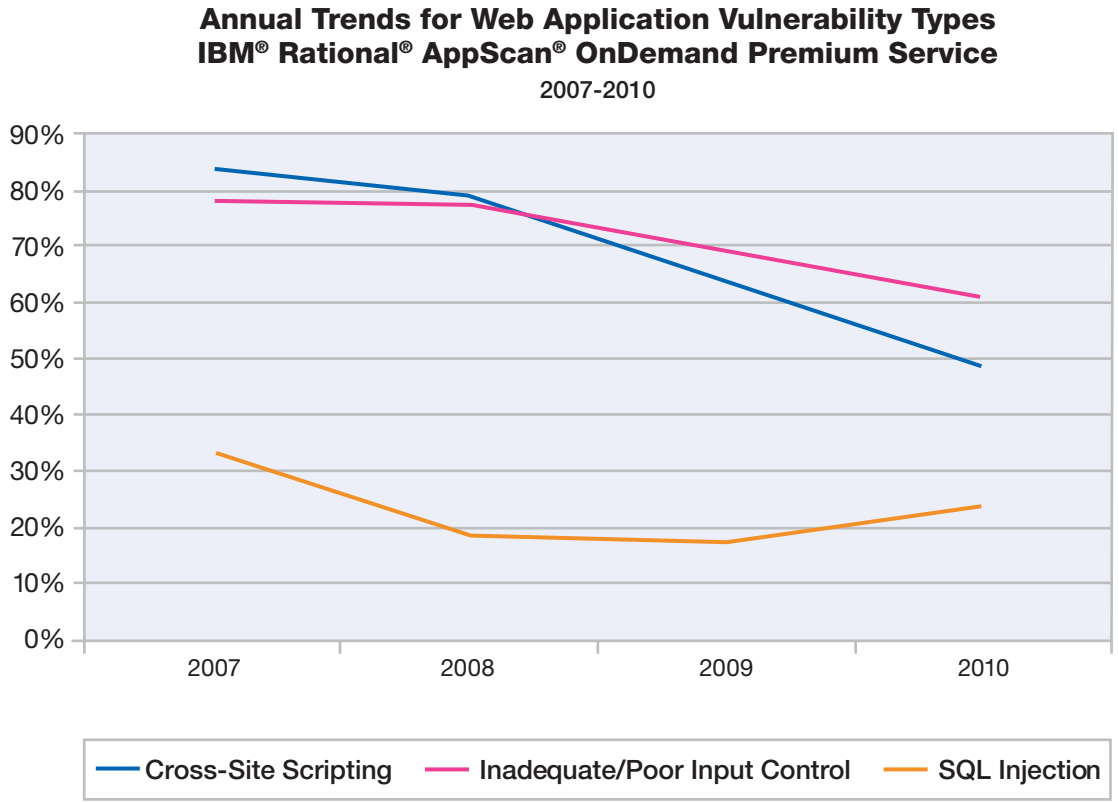


Figure 68: Annual Trends for Web Application Vulnerability Types IBM® Rational® AppScan® OnDemand Premium Service – 2007-2010

ANNUAL TRENDS								
Vulnerability Type	**2007**		**2008**		**2009**		**2010**	
	Avg. vulnerability per test	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**
Cross-Site Request Forgery	1.9	22%	1.8	20%	7.9	59%	3.8	53%
Cross-Site Scripting	12.7	83%	17.9	79%	40.8	64%	5.8	49%
Error Message Information Leak	46.9	83%	22.6	74%	23.5	68%	15.3	56%
Improper Access Control	3.9	56%	2.4	67%	0.8	30%	0.9	31%
Improper Application Deployment	2.6	50%	3.2	54%	3.0	51%	1.9	33%
Improper Use of SSL	28.9	50%	23.8	74%	38.8	51%	26.4	60%
Inadequate / Poor Input Control	14.4	78%	28.1	77%	44.4	69%	10.5	61%
Information Disclosure	6.6	61%	8.7	63%	12.9	64%	16.6	84%
Insufficient Web Server Configuration	16.5	72%	5.4	46%	1.4	31%	4.4	44%
Non Standard Encryption	7.3	28%	2.4	17%	2.5	35%	1.6	22%
SQL injection	1.3	33%	5.3	19%	1.7	18%	2.3	23%

Table 14: Annual trends for Web application vulnerability types, 2007 – 2010, IBM® Rational® AppScan® OnDemand Premium Service

Business segments

As in 2009 we were able to split out our 2010 statistics by business segments. Where the number of data points would allow, we were able to split out data for five business segments.

In 2010, financial applications were again the best performing segment. Financial applications were found to not only have lower percentages attributed to the likelihood of finding each of the vulnerabilities covered, but they also have very low numbers for the instances of each finding found for a given test. So while XSS and SQL injection might be found in some financial applications, it would typically be an isolated occurrence and not a flaw seen throughout the application. The same is not true for applications for industrial and IT organizations.

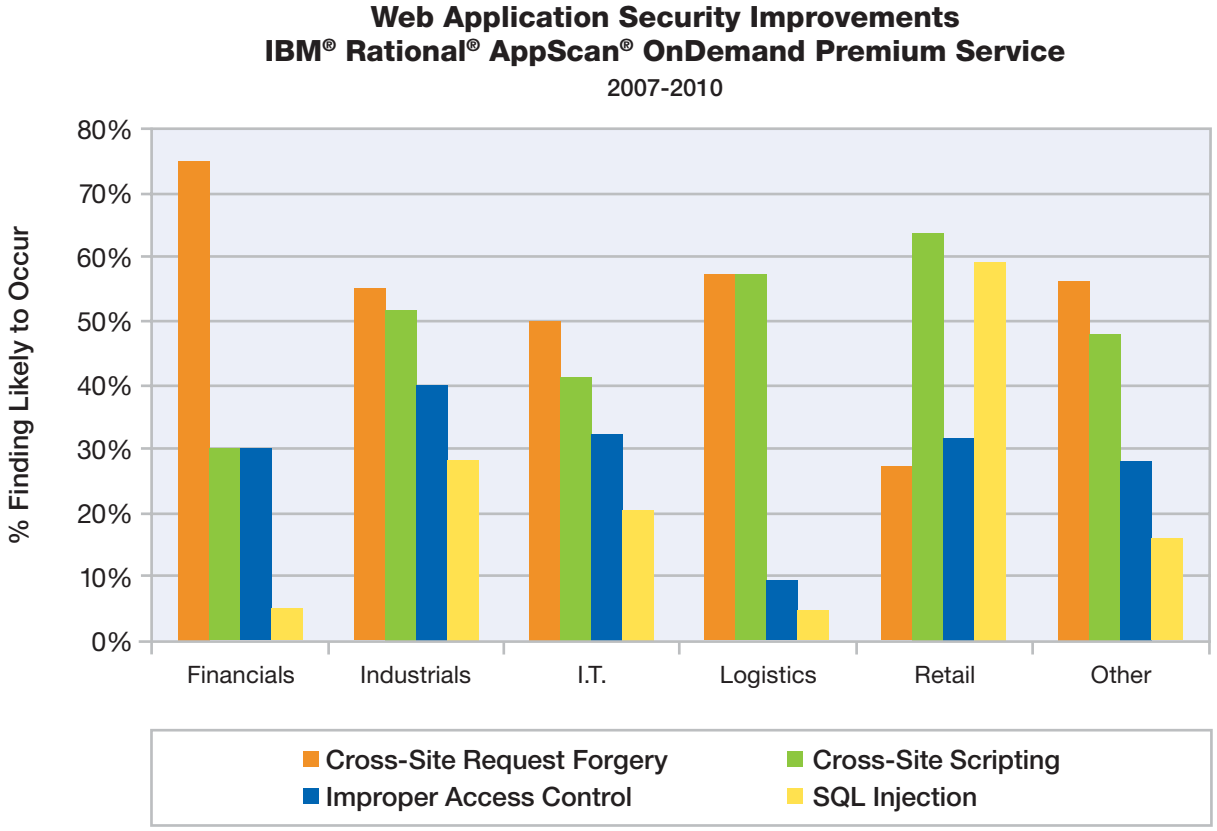


Figure 69: Web Application Security Improvements IBM® Rational® AppScan® OnDemand Premium Service – 2007-2010

	BUSINESS SEGMENT											
Vulnerability Type	**Financials**		**Industrials**		**Information Tech.**		**Logistics**		**Retail**		**Other**	
	Avg. vuln per test	**% one vuln likely to occur**	**Avg. vuln per test**	**% one vuln likely to occur**	**Avg. vuln per test**	**% one vuln likely to occur**	**Avg. vuln per test**	**% one vuln likely to occur**	**Avg. vuln per test**	**% one vuln likely to occur**	**Avg. vuln per test**	**% one vuln likely to occur**
Cross-Site Request Forgery	6.3	75%	2.6	55%	4.2	50%	9.2	57%	0.5	27%	2.4	56%
Cross-Site Scripting	0.4	30%	7.6	52%	6.4	41%	1.7	57%	2.6	64%	11.0	48%
Error Message Information Leak	10.9	80%	23.2	58%	14.6	47%	0.7	33%	17.8	59%	11.2	60%
Improper Access Control	0.4	30%	1.2	40%	0.7	32%	0.1	10%	2.2	32%	0.4	28%
Improper Application Deployment	2.4	55%	1.3	32%	2.1	24%	1.6	24%	0.4	27%	3.9	44%
Improper Use of SSL	32.1	90%	15.6	33%	19.4	50%	45.7	81%	20.0	73%	46.6	88%
Inadequate / Poor Input Control	3.5	40%	11.8	63%	13.8	65%	1.6	48%	11.3	82%	15.1	60%
Information Disclosure	10.9	75%	22.5	92%	17.4	82%	13.4	90%	7.8	82%	16.2	76%
Insufficient Web Server Configuration	1.0	50%	8.1	58%	4.4	44%	0.5	24%	2.3	27%	3.4	36%
Non Standard Encryption	2.1	10%	1.7	23%	0.7	24%	0.3	10%	4.6	41%	0.4	20%
SQL injection	0.1	5%	1.4	28%	5.0	21%	0.0	5%	7.0	59%	0.3	16%

Table 15: Most Prevalent Web Application Vulnerabilities by Industry, IBM® Rational® AppScan® OnDemand Premium Service

Application security test cycle

For the first time we collated data relating to the actual test cycle that was being conducted. This allowed us to see the correlation between the initial test of an application and the follow-up retest. In a pleasing way, the trend between these two statistics is that there is a significant decline in the likelihood of finding vulnerabilities in the retest. In many cases this reduction is more than half that of the original. This demonstrates the importance not only of testing applications, but also that follow up and mitigation are equally important.

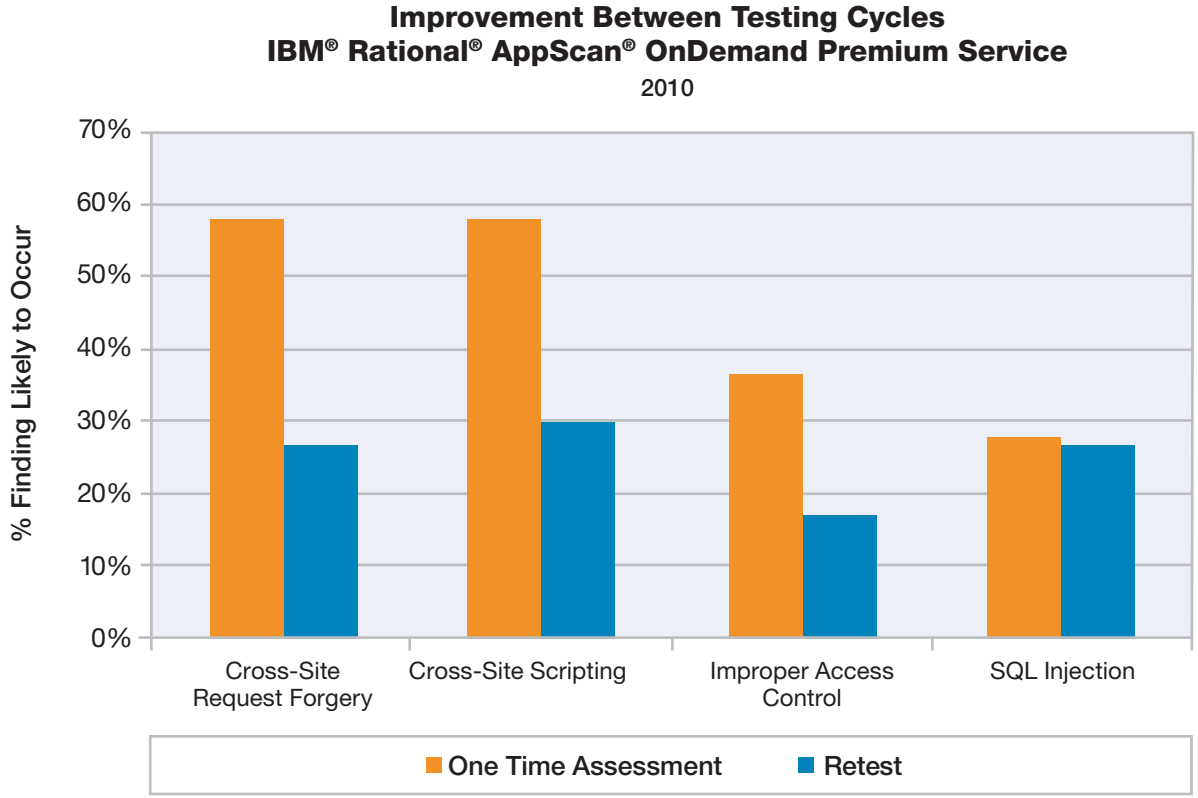


Figure 70: Improvement Between Testing Cycles IBM® Rational® AppScan® OnDemand Premium Service – 2010

SECURITY TEST CYCLE						
Vulnerability Type	**One Time Assessment**		**Quarterly Assessment**		**Retest**	
	Avg. vulnerability per test	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**
Cross-Site Request Forgery	3.2	58%	7.8	58%	0.6	27%
Cross-Site Scripting	8.8	58%	1.0	35%	0.8	30%
Error Message Information Leak	22.5	63%	3.4	43%	4.5	43%
Improper Access Control	1.2	37%	0.4	25%	0.3	17%
Improper Application Deployment	2.4	35%	1.5	28%	0.4	30%
Improper Use of SSL	27.2	54%	35.5	83%	11.3	53%
Inadequate / Poor Input Control	15.8	74%	1.6	43%	2.3	33%
Information Disclosure	21.3	86%	10.3	78%	7.1	83%
Insufficient Web Server Configuration	6.0	48%	1.3	33%	2.7	40%
Non Standard Encryption	1.5	25%	1.3	13%	2.2	23%
SQL injection	3.3	28%	0.1	8%	1.4	27%

Table 16: Security test cycles by vulnerability type, IBM® Rational® AppScan® OnDemand Premium Service 2010

Application technology

Another new statistic for us in 2010 was taken from looking at the technology of the application. We were only able to split this across three types but this still showed some interesting results. ASP.NET applications were clearly more susceptible to SQL injection than Java or PHP. The likely reason is that ASP.NET applications would typically use SQL Server as a backend database. SQL injection is better documented and easier to detect in this technology.

PHP overall performed best of the three technologies. However, it is worth highlighting that our data is taken entirely from commercial applications.

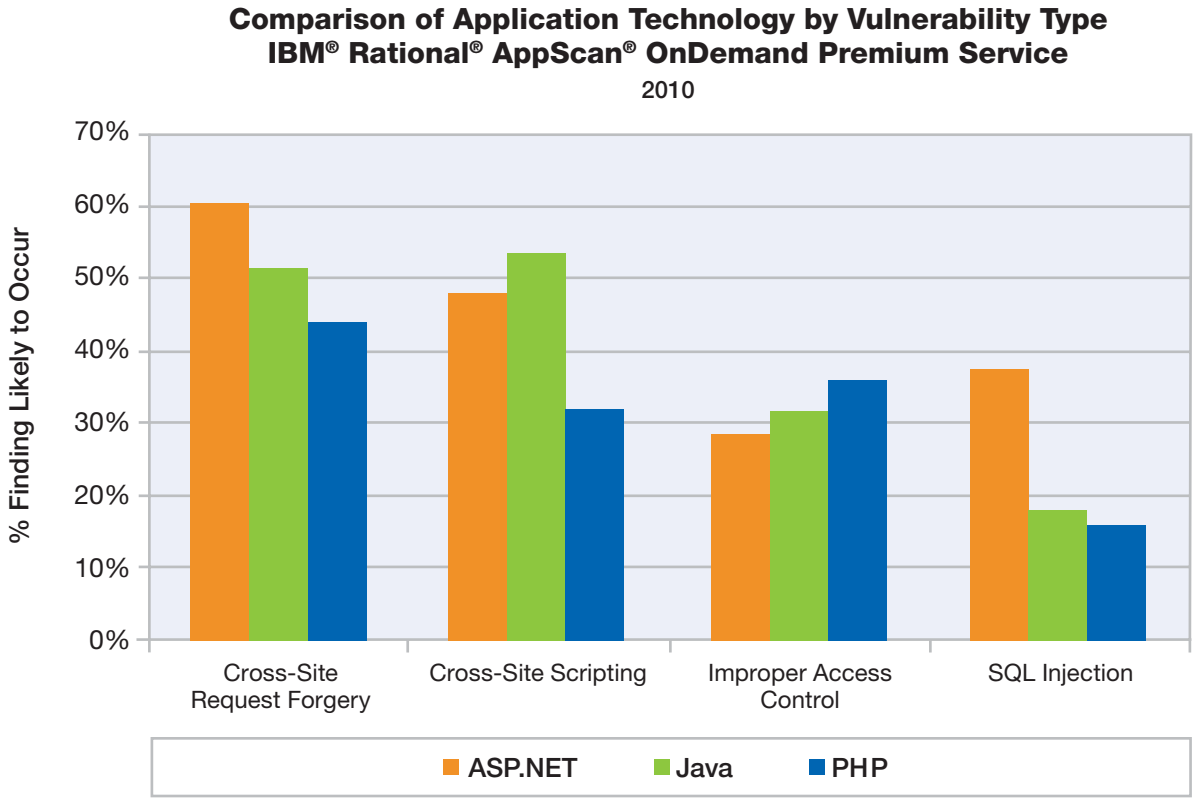


Figure 71: Comparison of Application Technology by Vulnerability Type IBM® Rational® AppScan® OnDemand Premium Service – 2010

APPLICATION TECHNOLOGY						
Vulnerability Type	**ASP.NET**		**Java**		**PHP**	
	Avg. vulnerability per test	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**
Cross-Site Request Forgery	2.8	61%	4.4	51%	3.4	44%
Cross-Site Scripting	4.9	48%	7.2	53%	1.9	32%
Error Message Information Leak	23.6	71%	13.7	51%	3.4	40%
Improper Access Control	1.1	29%	0.8	32%	0.8	36%
Improper Application Deployment	2.5	48%	1.5	26%	2.0	28%
Improper Use of SSL	28.2	64%	28.8	55%	12.4	72%
Inadequate / Poor Input Control	10.6	66%	12.1	59%	3.7	56%
Information Disclosure	24.7	84%	14.5	88%	6.6	72%
Insufficient Web Server Configuration	3.0	50%	5.7	41%	2.3	44%
Non Standard Encryption	2.7	30%	1.1	17%	0.7	24%
SQL injection	3.3	38%	2.3	18%	0.2	16%

Table 17: Comparison of application technology by vulnerability type, IBM® Rational® AppScan® OnDemand Premium Service 2010

Hybrid analysis sheds light on vulnerability blind spot

Background and methodology

In the past ten years, many whitepapers, research articles, and Blog posts have been published on the subject of server-side web application vulnerabilities such as SQL injection, cross-site scripting, and HTTP response splitting. In addition, several projects such as the WASC web hacking incident database or the WASC statistics projects have tried to estimate the incidence of such issues in the real world.

On the other hand, there is a dearth of information and statistics on the incidence of client-side JavaScript™ vulnerabilities in web applications, even though these vulnerabilities can be just as severe as their server-side counterparts. We suspect that the main reason for this lack of information is that client-side vulnerabilities may be harder to locate, and require deep knowledge of JavaScript and the ability to perform code review of HTML pages and JavaScript files.

As Web 2.0, AJAX applications, and Rich Internet Applications (RIAs) become more common, client-side JavaScript vulnerabilities may become more relevant, with a potential rise in the amount of such issues being exploited by malicious hackers.

This summary presents the results of research performed by the IBM Rational application security group into the prevalence of client-side JavaScript vulnerabilities, using a new IBM technology called JavaScript Security Analyzer (JSA). JSA performs hybrid analysis by applying static taint analysis on JavaScript code collected from web pages and extracted by an automated deep web-crawl process. From our perspective, this kind of analysis is superior to—and more accurate than—regular static taint analysis of JavaScript code because it includes the entire JavaScript codebase in its natural environment: fully rendered HTML pages and the browser's Document Object Model (DOM).

The research used a sample group of approximately 675 websites, consisting of all the Fortune 500 companies and another 175 handpicked websites, including IT, web application security vendors, and social networking sites. In order to avoid damage to the sites or interference with their regular behavior, we used a non-intrusive web crawler, similar to that of a web search engine, which retrieved approximately 200 web pages and JavaScript files per site into a repository. We then used the JavaScript Security Analyzer to analyze these pages offline for client-side JavaScript vulnerabilities. We concentrated on two main types of issues: DOM-based cross-site scripting, and open redirects.

JavaScript analyzer results

The results of our research were quite disturbing: about 98 sites (14 percent) of the 675 sites suffer from many severe client-side JavaScript issues, which could allow malicious hackers to perform attacks such as:

- Infecting users of these sites with malware and viruses.
- Hijacking users' web sessions and performing actions on their behalf.
- Performing phishing attacks on users of these sites.
- Spoofing web contents.

The troubling fact about these statistics is that most organizations have no efficient process or automated solution to assist them with the task of locating these types of issues.

Our research also showed that 38 percent of the vulnerable sites suffered from these vulnerabilities as a result of using third party JavaScript code such as:

- Marketing campaign JavaScript snippets.
- Flash embedding JavaScript snippets.
- Deep linking JavaScript libraries for Adobe® Flash and AJAX applications.
- Social networking JavaScript snippets.

Percentage of Sites Vulnerable to Client-Side JavaScript Issues

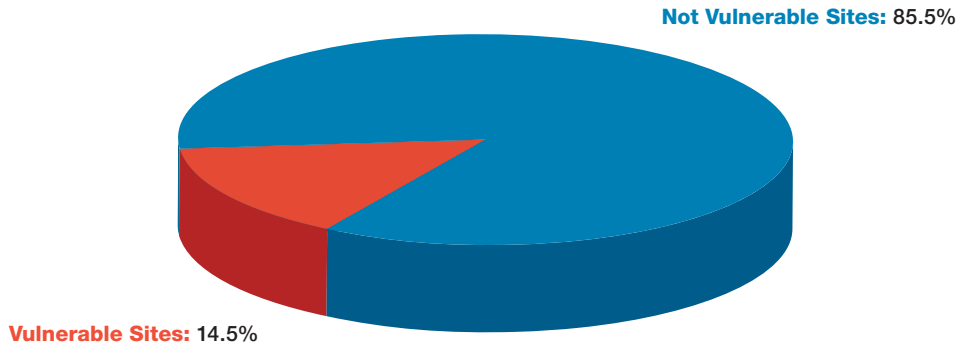


Figure 72: Percentage of Sites Vulnerable to Client-Side JavaScript Issues

Vulnerable Third-Party JavaScript Code Versus In-House Written Code

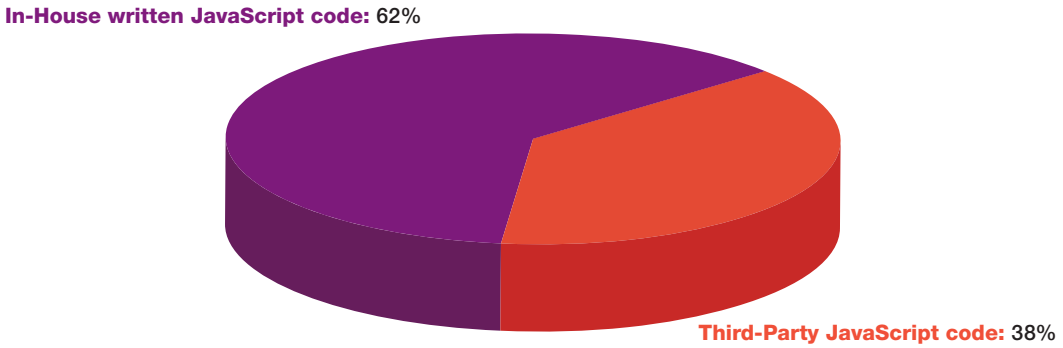


Figure 73: Vulnerable Third-Party JavaScript Code Versus In-House Written Code

Of the 98 vulnerable sites, 92 sites (94 percent) suffered from DOM-based cross-site scripting issues, whereas only 11 sites (11 percent) suffered from open redirects. The total amount of DOM-based cross-site scripting issues found was 2370, while only 221 open redirects were found.

Based on the dataset that we analyzed, we may extrapolate that the likelihood that a random page on the Internet contains a client-side JavaScript vulnerability[24] is approximately one in 55.

To summarize, from the information uncovered by this research we conclude that client-side vulnerabilities are quite common in modern web applications, especially those that rely on JavaScript for performing client-side logic—i.e. Web 2.0, AJAX, and Rich Internet Applications. In addition, a substantial number of the existing JavaScript client-side vulnerabilities on the Internet are introduced from 3rd party code that is not developed in-house, and usually is not reviewed for security issues.

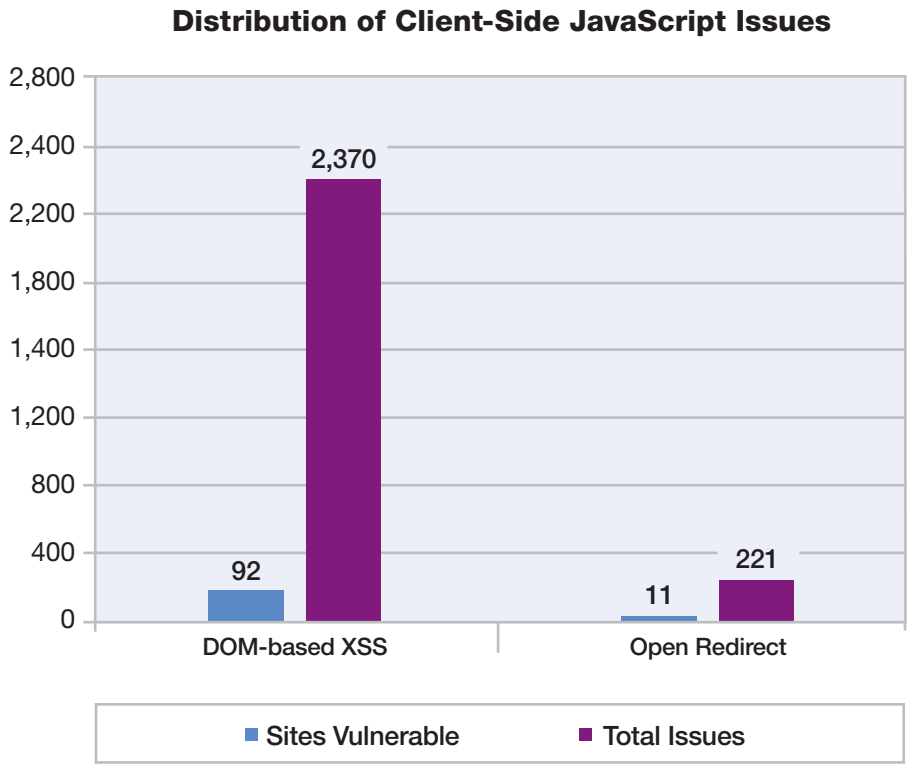


Figure 74: Distribution of Client-Side JavaScript Issues

24 Information about the prevalence of client-side JavaScript vulnerabilities was included from a Rational research paper titled "Close Encounters of the Third Kind" (http://www.ibm.com/common/ssi/cgi-bin/ssialias?infotype=SA&subtype=WH&appname=SWGE_RA_RA_USEN&htmlfid=RAW14252USEN&attachment=RAW14252USEN.PDF).

Web application hack-ability and efficient defense

IBM Security provides both scanning products and services. The value of this combination is that, in aggregate, IBM can show how effective companies are in securing their web applications. While these numbers do not have direct bearing on your business, which has its own risk picture, they do provide a comparative view which is useful.

The following Web Application Vulnerability scanning is from IBM Professional services, and these vulnerability numbers represent vulnerabilities found by both Rational® AppScan® as well as manual site analysis by a professional penetration tester.

Figure 75 to the right shows the likelihood that each vulnerability will occur within a web application. One thing to understand is that some of these scans are repeat scans, so some of the decline shown is due to fixed vulnerabilities over time.

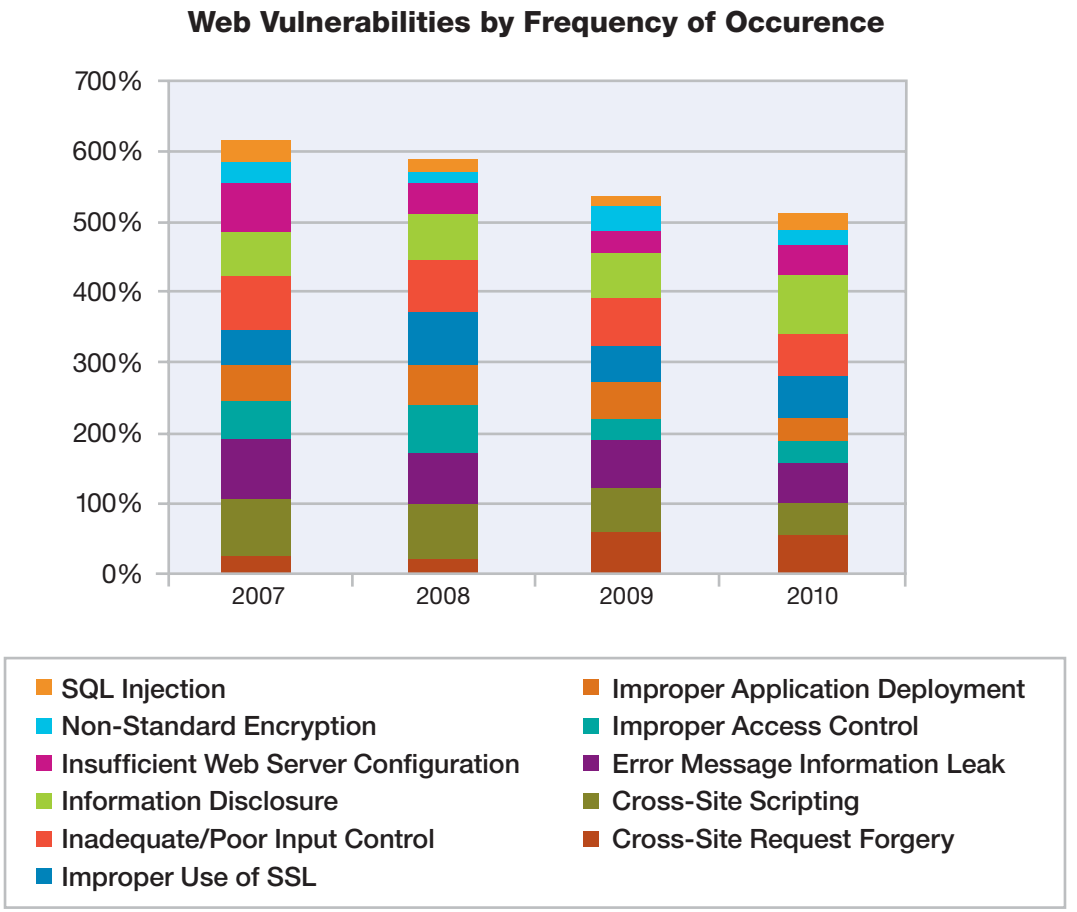


Figure 75: Web Vulnerabilities by Frequency of Occurence

If you want to compare your web vulnerability levels with other companies in your business segment, the chart below shows the average number of instances of a given vulnerability type across industries.

BUSINESS SEGMENT						
Vulnerability Type	**Financials**		**Industrials**		**Information Tech.**	
	Avg. vulnerability per test	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**
Cross-Site Request Forgery	6.3	75%	2.6	55%	4.2	50%
Cross-Site Scripting	0.4	30%	7.6	52%	6.4	41%
Error Message Information Leak	10.9	80%	23.2	58%	14.6	47%
Improper Access Control	0.4	30%	1.2	40%	0.7	32%
Improper Application Deployment	2.4	55%	1.3	32%	2.1	24%
Improper Use of SSL	32.1	90%	15.6	33%	19.4	50%
Inadequate / Poor Input Control	3.5	40%	11.8	63%	13.8	65%
Information Disclosure	10.9	75%	22.5	92%	17.4	82%
Insufficient Web Server Configuration	1.0	50%	8.1	58%	4.4	44%
Non Standard Encryption	2.1	10%	1.7	23%	0.7	24%
SQL injection	0.1	5%	1.4	28%	5.0	21%

Table 18: Vulnerability type for Financials, Industrials, Information Technology, IBM® Rational® AppScan® OnDemand Premium Service 2010

BUSINESS SEGMENT						
Vulnerability Type	**Logistics**		**Retail**		**Other**	
	Avg. vulnerability per test	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**
Cross-Site Request Forgery	9.2	57%	0.5	27%	2.4	56%
Cross-Site Scripting	1.7	57%	2.6	64%	11.0	48%
Error Message Information Leak	0.7	33%	17.8	59%	11.2	60%
Improper Access Control	0.1	10%	2.2	32%	0.4	28%
Improper Application Deployment	1.6	24%	0.4	27%	3.9	44%
Improper Use of SSL	45.7	81%	20.0	73%	46.6	88%
Inadequate / Poor Input Control	1.6	48%	11.3	82%	15.1	60%
Information Disclosure	13.4	90%	7.8	82%	16.2	76%
Insufficient Web Server Configuration	0.5	24%	2.3	27%	3.4	36%
Non Standard Encryption	0.3	10%	4.6	41%	0.4	20%
SQL injection	0.0	5%	7.0	59%	0.3	16%

Table 19: Vulnerability type for Logistics, Retail, Other, IBM® Rational® AppScan® OnDemand Premium Service 2010

If you are considering what technology to use for your next web application, these numbers may help you focus your research.

Vulnerability Type	ASP.NET		Java		PHP	
	Avg. vulnerability per test	% one vulnerability likely to occur	Avg. vulnerability per test	% one vulnerability likely to occur	Avg. vulnerability per test	% one vulnerability likely to occur
Cross-Site Request Forgery	2.8	61%	4.4	51%	3.4	44%
Cross-Site Scripting	4.9	48%	7.2	53%	1.9	32%
Error Message Information Leak	23.6	71%	13.7	51%	3.4	40%
Improper Access Control	1.1	29%	0.8	32%	0.8	36%
Improper Application Deployment	2.5	48%	1.5	26%	2.0	28%
Improper Use of SSL	28.2	64%	28.8	55%	12.4	72%
Inadequate / Poor Input Control	10.6	66%	12.1	59%	3.7	56%
Information Disclosure	24.7	84%	14.5	88%	6.6	72%
Insufficient Web Server Configuration	3.0	50%	5.7	41%	2.3	44%
Non Standard Encryption	2.7	30%	1.1	17%	0.7	24%
SQL injection	3.3	38%	2.3	18%	0.2	16%

Table 20: Vulnerability type by web application (ASP.NET, Java, PHP), IBM® Rational® AppScan® OnDemand Premium Service 2010

At first review, these numbers may seem to be of limited use. While every security fanatic holds the theoretical concept of zero vulnerability as a laudable but perhaps impossible goal, there is value in looking at a comparative vulnerability. Understanding this stems from understanding the nature of your adversary, the attacker. Some may claim attackers as being lazy, but it is demonstrated that attackers range from disciplined ascetics focused only on the acquisition of hacking skills to the lazy buyers of hacking products. One thing is certain, attackers pursue efficiency as the following examples illustrate.

- Attackers use scanning tools and automated propagation tools which are designed to use any and all vectors to fulfill one simple intent: Give control of as many computers as possible to a master.
- They use cached pages on search sites to assess your vulnerability so that they can "probe the ghost of your defenses" without probing you directly. Your cached page can tell them what to attack without directly examining your live web pages.
- Attack business sites rank targets, building search engines for hacking targets. So the most vulnerable targets are attacked the most. This is where the comparative view starts to make sense, in that a less vulnerable website will be ranked lower and therefore hacked less.
- There is a self-sustaining cycle where vulnerable websites allow the propagation of bots, which then generate more fake sites with malware, etc. This cycle is self-reinforcing.

Understanding that hacking is mainly about efficiency, we can prioritize and strategize our web application defense to be as efficient as possible. Our unobtainable goal of zero vulnerability (unplugged, powered off, and placed inside a Faraday cage) can shift to becoming a relatively inefficient target so that it takes more effort to compromise your company rather than another. You can use numbers in the previous chart to have an idea of how attractive your business servers may be to attackers.

Avoid the Net cast by automation

Automated systems sweep the net for easily exploited websites. Typically, these automated attacks are mitigated most effectively by a separate web access control system and Intrusion Prevention System (IPS) with web application protection capabilities.

For web applications, a good choice is to separate your authentication solution from your web application. This can provide you with vulnerability mitigation for several types of web application vulnerabilities at once. Separate authentication also makes access control itself more efficient for administrators to manage than from within the web application code.

When it comes to intrusion prevention, efficiency should be measured in actions taken over vulnerabilities blocked. The perfect Intrusion Detection System has an efficiency ratio approaching 0, where turning it on results in perfect protection. This of course is driven by the accuracy of detection. Threat prevention accuracy is driven in turn by security research, so accuracy should be viewed as a “historical trend” of pre-emptiveness.

The most efficient threat-mitigation systems block whole classes of threats with a few detection algorithms. Part of the value of assessing accuracy as a historical track record is taking into account the background and motivations of the researchers.

These tools rapidly can close down vulnerabilities, giving you more time to fix your vulnerabilities efficiently.

Fix vulnerabilities efficiently

Vulnerability prioritization is a balance between the difficulty of the fix versus the ease of the attack. This is where professional penetration testing and vulnerability assessment services provide additional value because they identify relationships that help you prioritize. Vulnerabilities in web applications are often related, one hard-to-fix vulnerability may be mitigated by fixing several easy-to-fix vulnerabilities. For example, request forgery is often difficult to fix, but to be more effective, it is often combined with link injection as a vehicle for delivering malicious content. In addition to identifying complex relationships, the professional penetration tester can find vulnerabilities that are recognized only by intelligent human probing.

Clearly, those vulnerabilities which are blocked by Intrusion Prevention and access control are less important to fix, especially if the fix is difficult, but it is always a good idea to fix broken applications, if for no other reason than to help your application developers avoid the same mistakes in the future.

The best defense against the elite

If you avoid the net cast by automation, you should fix the vulnerabilities you can, and make your remaining vulnerabilities difficult to access; and you hopefully will be left exposed only to the hacker elite. From here you can continuously work toward the unobtainable “Zero Vulnerability” posture with relative safety.

Section IV—Emerging Trends in Security

The Emerging Trends in Security section takes a look at fast developing technology that presses upon enterprises considering whether or not it is time to make investments in these future areas. We explain where threats and exploits are being utilized in these early technology adoptions and how enterprises can stay focused.

Mobile security trends

As enterprises approach the huge potential in efficiency that mobile computing has to offer, the two primary hurdles they will likely face are complexity (due to proliferation of platforms) and security. This section explores the approaches, strategy, and suggested controls as a perspective on the external threat landscape in this area.

In approaching the mobile security topic, there are two fundamental observations to consider. First, most of what is considered best practice around securing mobile devices is still not nearly as well defined as it is in the corresponding personal computing space. Second, the underlying platforms themselves are substantially untested and likely contain years of vulnerability discovery ahead of them.

2010 saw significant increases in the number of vulnerabilities disclosed for mobile devices as well as the number of public exploits released for those vulnerabilities, but it's important to keep these increases in perspective. Many of the vulnerabilities impacted shared software components that are used by both mobile and desktop software. The vulnerability research that is driving these disclosures is not necessarily mobile-centric.

Likewise, many of the public exploits that have been released for these vulnerabilities are not actually designed to function properly on mobile platforms, although they could be retooled to do so by an interested party.

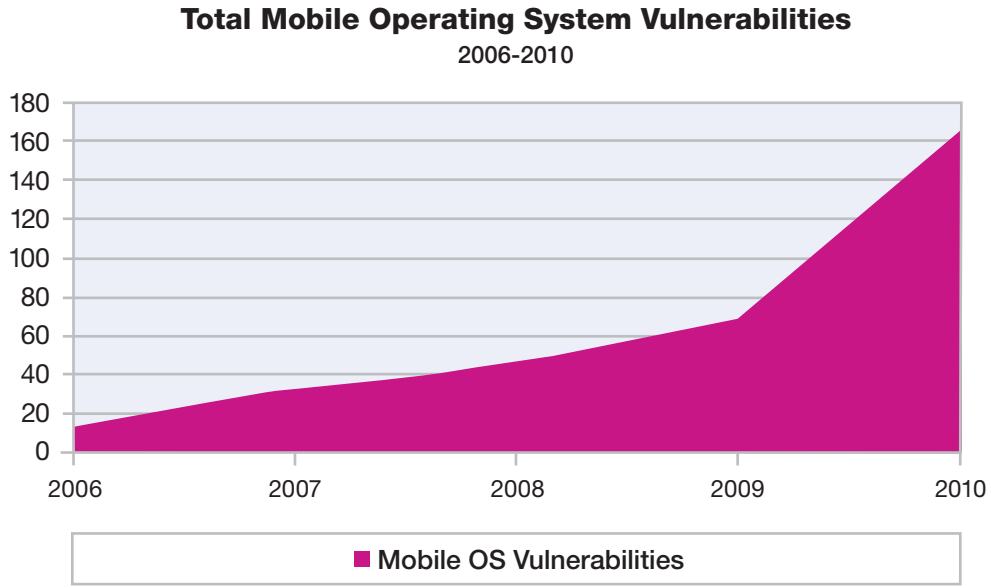


Figure 76: Total Mobile Operating System Vulnerabilities – 2006-2010

Nevertheless, there have been exploits released this year that are designed to function on various popular mobile platforms. One of the motivations of these exploit writers is a desire by mobile device users to "jailbreak" or "root" their devices to enable various kinds of functionality not intended by the manufacturers. This motivation drives the creation of mature, reliable exploit code that is widely disseminated and can be readily repurposed for malicious use. For example, early in 2011 malicious applications were distributed in the Android app market that used widely disseminated exploit code to obtain root access to devices and steal information. The vulnerabilities exploited by these malicious applications had been publicly disclosed for months at the time of the attacks. While attacks like this are not yet common place, they may happen more frequently in the future. It's also worth pointing out that the use of mobile devices in an enterprise environment brings other software systems into play, such as enterprise management servers and desktop sync software, which have also been subject to vulnerability disclosures and exploit releases.

We aren't seeing a lot of widespread attack activity targeting these vulnerabilities today, because mobile devices likely do not represent the same kind of financial opportunity that desktop machines do for the sort of individuals who create large Internet botnets. As e-commerce involving mobile phones increases in the future, it may bring with it a greater financial motivation to target phones, and an associated increase in malware attacks. However, mobile devices do represent opportunities for sophisticated, targeted attackers today. There are a number of vulnerabilities to target, and there is exploit information available. Malicious software on the devices can be used to spy on users, access sensitive information on the phones, and reach back into corporate networks. Therefore, enterprises should take the risk of targeted malware on phones seriously.

Because of these risks, enterprises may be apprehensive to move forward with significant enablement of multiple mobile device platforms. However, in addition to the potential efficiency benefits of enablement, it may be more useful to implement effective management technologies rather than provide technical controls needed to prevent the forward movement that will be attempted without their support anyway. It will likely become more expensive to implement technical controls to help ensure enterprise data is not finding its way to employee smartphones in an ad hoc fashion. Investing that same funding into properly securing some level of additional platforms to enable this trend and its subsequent efficiency gains may make the most amount of sense for many environments.

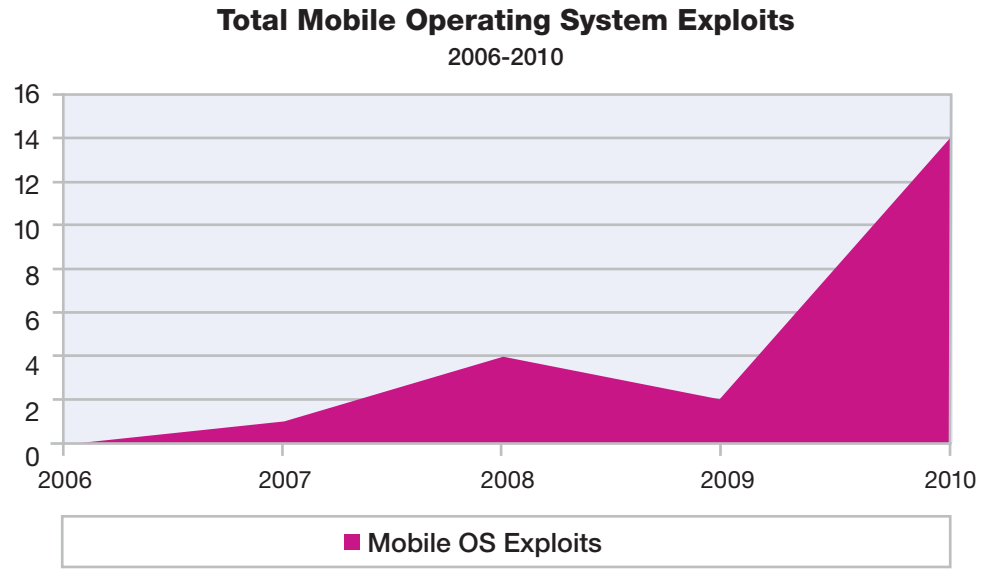


Figure 77: Total Mobile Operating System Exploits – 2006-2010

Effective controls to manage mobile devices

Existing enterprise security standards serve to help protect the integrity of our data and its corresponding IT infrastructure. Therefore, it should be relatively straightforward for the enterprise to identify the required controls for a given data classification. The data-focused approach should provide the foundation of the appropriate security standards to adequately protect this same data on mobile devices while taking into account the unique aspects of mobile technology. This enterprise data is of no less value because it now resides on the latest, shiny new smartphone rather than on existing personal computers or servers.

As enterprises arrive at the specific controls they need to enforce, it is vital to establish correct assumptions on the various classifications of data that will end up residing on their devices. This can be approached in different ways including identifying classification based on employee roles or services that are expected for device support. Regardless of the approach, it is paramount that this classification is established in order to clearly define the resulting controls that are required.

This statement ideally results in a fairly small set of controls required to host, transmit, and process this data so the controls can be clearly defined in employee security standards as well as implemented and enforced via technology.

For example, here is a typical set of controls.

- A device password of adequate strength to protect the data classifications expected to reside on the device.
- A timeout and lockout feature controlled by the device password and set for a period of minimal time. This is typically anywhere from five to 30 minutes; the shorter the better from a security perspective.
- Device configuration such that any data stored on the device is removed after “X” failed login attempts, or the device is managed by a remote service with this ability. If both controls are possible, they should both be used. This data removal should include data stored on memory media (i.e., flash memory) used by the device if possible.
- Password prompt on the device should pause for an incremental time after each unsuccessful login attempt to protect against brute-force login attempts if possible.
- Install and run an anti-malware program on any device that has access to the enterprise infrastructure or has access to enterprise data.
- Install and run a firewall program on the device if possible. Limiting access into the enterprise is an effective means of decreasing risk.
- Remote access for synchronization of data or access to enterprise infrastructure should always go through an approved Remote Access Service (RAS) gateway using adequate access credentials. It is a sound security practice to minimize or discourage the practice of making internal services available externally. Doing so simply increases attack surface area.
- Configure Bluetooth so that it is not discoverable and it will connect only with paired devices on all handheld devices supporting these features.

If devices cannot meet these minimum requirements, they should not be suitable for enterprise use. Ideally, technology should be implemented to properly configure devices for employees as part of the boarding process. This establishes a trusted relationship at the completion of the boarding process.

As you review these defined controls, notice that there are some controls unique to smartphones. The requirement to either remotely or locally remove information is a compensatory control to address the unique nature of smartphones. Because of their size and common use cases, enterprises should expect that loss and theft will be higher than they’ve typically witnessed in laptop programs. The reality is that even the most conscientious employee can use their smartphone in an airport, cab, hotel, or anywhere they go because that is the nature and benefit of the technology.

In reviewing the myriad of platforms that have become available in the last couple of years, the primary observation from a security perspective is that platform vendors have designed their products to appeal to consumers with the enterprise being a secondary concern. Most smartphone platforms did not lend themselves to immediate enterprise use in their initial versions. Nor did they support the typical controls that an enterprise would expect. In fairness, nearly all vendors have recognized this and have begun to embrace that their customers desire to use their devices across both their work and personal lives. As a result, typically as platforms hit version two or three, they include most or all of the minimum enterprise requirements. It is particularly important that enterprises consider patch management of these devices as a part of their overall strategy for managing them. As discussed above, the desire to "jailbreak" or "root" the devices has been one of the drivers for the public dissemination of reliable exploit code for mobile devices, and this sort of exploit code has been used in malicious attacks.

Although it is the responsibility of the ecosystem of mobile device makers and telecom companies to make sure that updates are available that fix these vulnerabilities, those updates may have to be manually installed by end users. Experience shows that manual end user update processes are inconsistently complied with and users who aren't keeping up with updates may have devices that are exposed to attack. One way to combat this problem is to develop a mechanism for regularly reminding corporate mobile device end users that installing updates is an important part of keeping their device and their corporate data secure.

It is very likely that at least initially, as software updates become more important and more frequent to fix exploited vulnerabilities that enterprises may only be able to rely on their MDM (Mobile Device Management) solutions to simply limit synchronization to updated versions. Currently, the platform vendor/hardware vendor/carrier ecosystem has not embraced the notion of frequent updates that can be distributed by third parties, like enterprises, in order for them to more closely manage vulnerabilities on their enterprise devices. Obviously, as this moves forward, it may vary from platform to platform, adding an additional challenge of inconsistency for the enterprise.

Encryption

While encryption of data at rest is not required for some types of information in some industries, it should be used for a subset of specific types of data in nearly every enterprise. This is driven by legislation as well as by customer expectation so we'll continue to see this apply to at least a portion of employees for every enterprise. Whether enterprises leverage native encryption capabilities that may exist in some platforms or seek some of the third party encryption solutions that exist, it is crucial to thoroughly understand the implementation you've selected to help ensure that it meets the specific encryption requirements defined in your security policy.

Note that nearly all data encryption approaches for smartphones have been software-based and do not provide an ideal architecture for the typical smartphone. It is hard to determine if this is simply a point in time in the development of mobile devices and more will eventually include hardware-based encryption capabilities. This concern may also be mitigated as processor capacity continues to increase in smartphones and we see both faster and multiple processors in these devices.

Until the summer of 2010, some felt smartphone malware was an urban legend but as a result of multiple security research disclosures that summer, there is now more recognition that this is both possible and likely common moving forward. Enterprises should not discount this threat because it is not as pervasive as the existing personal computer threat landscape.

It is valuable to maintain an information-based objective approach as we look at the current threat landscape in this arena. While the threat of mobile malware has existed as long as the devices have been available, it remains far less prevalent than malware attacks against many other devices. In fact,

most of the activity that drives the most malware today remains focused on targeting Windows XP computing devices. This should not be a surprise—they exist in the hundreds of millions and are typically manned by a wide range of user expertise. We should expect this target to exist as it does today until the prevalence of XP devices begins to decrease as the 2014 Windows XP end-of-life support date arrives. Until then, XP will remain a primary target, especially with common malware development kits available.

One of the reasons Windows XP grew to the primary attack target is simple pervasiveness. Windows XP market share drove this attractiveness. The discovery of numerous vulnerabilities allowed it to grow and the existence of malware development kits allowed it to flourish. If you apply this same logic to the current smartphone landscape, you would note that at present, there is no single dominant platform. As there become clear winners in this space, we should expect them to be targeted.

In discussing smartphone malware, we may see a slightly different attack approach than we've seen in the personal computing space. Specifically, we may see malware introduced voluntarily by the device owner by using "vetted" application hosting in one of the many platform-specific application stores. This approach is already evidenced in existing malware and should be expected to increase as the number of available applications skyrockets. This will also challenge the end user because of the nature of smartphone application stores.

Unlike personal computers where this approach isn't prevalent, users likely will perceive the application store as a trusted source of software for their device. This couldn't be farther from the truth, with no existing application store providing secure code reviews. In fact, most do not provide any code review whatsoever, simply providing a place for developers who complete the registration process (which may include a minimal fee) to sell or give away their work. While it is undoubtedly possible to remotely compromise a smartphone device by socially engineering a user into clicking a link or visiting a URL, these attacks require remote code execution vulnerabilities, unlike the application store approach. It is likely that malicious behaviors in what appear to be trustworthy applications may provide an easy vector.

We should also expect that many of the same malware components we see in desktop malware will exist in their mobile counterparts. Components like keystroke loggers and proxies that redirect traffic and steal information have already been observed in smartphone malware. Multiple types of Premium SMS toll fraud malware exist; these are unique to smartphones and represent an easy way of generating quick revenue for the attacker.

Remote Access Service

Since smartphones in their essence exist as mobile devices and are typically outside of both the enterprise infrastructure and premise, a secure remote access service is a fundamental enabler of enterprise mobile computing.

In an ideal circumstance, a Remote Access Service (RAS) would only allow access to those devices it could demonstrate as trustworthy, rejecting all others. In addition, given the specific defined use cases for mobile devices, risk can be lowered by limiting this access to those destinations and services needed by the device and restricting those that are not required. RAS is another area where the desire for platform diversity becomes a challenge. Ideally, the enterprise would desire adoption of common, industry-standard secure access solutions that are commonly supported in many or most platforms.

Enterprise selection of RAS service should also focus on the technology selection that is best suited to smartphone devices. Typically, most personal computer RAS services use IPsec (Internet Protocol Security) as a means to establish an authenticated, secure tunnel across the Internet between the personal computer and the enterprise gateway. This approach has supported the needed, secure algorithms to help provide confidence that data in transit was well protected between the two

points. The obvious approach would be to transport that same approach to smartphone devices. Many smartphone platforms include IPsec VPN clients natively and work with most industry standard gateways. The benefit of this approach is that existing infrastructure can be leveraged, using the same level of security required.

The downside to this approach, when used with smartphones, is a real issue with device battery life and usability. Maintaining a constant tunnel between device and gateway, which is needed to synchronize data, quickly saps battery life. The alternative approach is to manage the use of this tunnel, leaving it connected only long enough to synchronize or access data, and then turning it off. Unfortunately, this loses many of the benefits gained by mobile efficiency.

An alternate approach is the use of Secure Sockets Layer (SSL) as a tunneling protocol within the remote access solution. While SSL is able to support similar encryption algorithms as IPsec (in terms of bit strength), it exists natively in http (s). SSL can provide an on-demand secure connection into the enterprise that does not require the mobile device to maintain a constant secure tunnel; it is only needed for actual data exchange. The primary concern with the use of SSL in a remote access service is in terms of the gateway. In most cases, this function is simply a reverse-proxy SSL-based exchange that is easily compromised and provides little security isolation. That said, there are SSL-based gateways available that do provide security functions which allow for the discovery of a trusted device (hence preferable from a security perspective) while still maintaining the battery-friendly, user-friendly approach observed with an on-demand secure access service.

Future security vision

Nirvana, as it applies to future of smartphone security and enterprise use, is likely the ability for smartphone devices and associated platforms to support dual personas on a single device. Since much of the smartphone growth within the enterprise likely will be comprised of employee-owned devices, the ability for enterprise data and controls to peacefully co-exist on a personal smartphone is the most desired state. In today's environment, enterprises should ensure control of their data regardless of where it is and this includes employee-owned smartphones. As a result, enterprise requirements should be applied to all smartphones enabled to access or store this data, regardless of owner liability. The ideal future state would allow the enterprise to properly secure access to its data and infrastructure to the degree required while allowing the individual to decide the security controls for their data and access to personally-subscribed services. For the enterprise, this would mean that all enterprise data, applications, and network access to and from the enterprise would be secured in their prescribed manner but would be enforced to only that "container" where those applications, associated data, and connectivity existed. Outside of the container, the user would be free to decide what kind of controls the device itself should contain and what applications they were comfortable with without regard to any impact on enterprise data, applications, and access.

The need for this approach and separation is necessary as we look at the future enterprise use of smartphones. Certainly, while starting with the need for malware prevention, we shouldn't expect that enterprise protection ends there. It is only a matter of time before things like intrusion prevention and data leakage prevention are requirements on smartphones as they've become on personal computers. Given the likely relatively limited nature of computing resources present on smartphones, the most viable approach to these needs is to push the execution of them into the enterprise remote access connection in a way that helps ensure that all connection to and from the device is forced through a common service that performs this inspection before allowing the traffic to its ultimate destination. To some degree, this inspection can be exacerbated by finite segregation between enterprise data and applications, access, and personal use but ultimately, even if only the enterprise portion needs this level of inspection, the most favorable approach will be to push a lot of this into the cloud or, in more likely terms, into the remote access enterprise connection.

The evolving state of security in the cloud

While security is still considered one of the major inhibitors to cloud adoption, organizations are increasingly adopting cloud-based technologies to address competitive market needs. This contradiction highlights the fact that many of the perceived challenges associated with cloud computing have been of less concern for a large subset of the market that have already adopted the cloud. We are seeing a shift in perception as cloud adoption evolves and knowledge increases. A recent study from Appirio focused on the state of the public cloud from the perspective of corporations that are using the public cloud for one or more service. Of the 155 medium-to-large companies responding to their survey, 28 percent agreed that security is the number one misconception about cloud computing and 39 percent said that cloud computing would be a pivotal enabler of an overall business transformation for their organization.[25]

Unlike other emerging technologies, the interest in security as it relates to cloud computing began close to its inception, and concerns about cloud security have received considerable attention in the marketplace. This has translated into hesitancy on the part of some organizations to aggressively adopt the cloud. In fact, many organizations are looking at a private cloud implementation as their initial foray into cloud computing in order to maintain control over data processing and security. Although perceptions about cloud computing in general may be changing, the fact remains that an organization's willingness to utilize the public cloud for mission-critical work usually depends on their understanding of potential risks and their assessment of whether their data can be adequately safeguarded. They also may rely on their experience with and knowledge of specific cloud-based solutions. In fact, we see greater adoption of cloud technologies with which the market has become more familiar. For example, although email is clearly a business critical application and can contain confidential data, many organizations have already leveraged some form of web-based email as part of a collaboration solution. The question for organizations is not whether the cloud as a whole is secure, but whether the organization is comfortable placing their workload on the cloud. As shown in Figure 78, the relevant component of the adoption curve is that most workloads can be suitable for cloud-based technologies. Whether that technology is adopted depends on the business benefits of the organization and their perception of the risks.

25 "State of the Public Cloud: The Cloud Adopters' Perspective" published October 2010, Appirio.

As with many outsourcing technologies, public cloud computing requires that the subscriber trust the provider to manipulate and handle their data with appropriate security measures. This trust relationship is paramount in cloud computing given that many providers are unable, or in some cases, unwilling to share their security controls or the details of the environment for the very purpose of maintaining security. Security best practices guidelines specific to cloud computing have begun to emerge from organizations such as the Cloud Security Alliance (CSA), which has a stated focus of providing security and privacy guidance for subscribers of cloud computing. This helps organizations evaluate their risk tolerance for using the cloud. With the increasing challenge of compliance that subscribers are facing, organizations should leverage industry recognized best practices while establishing their strategy around security and their use of cloud-based technology.

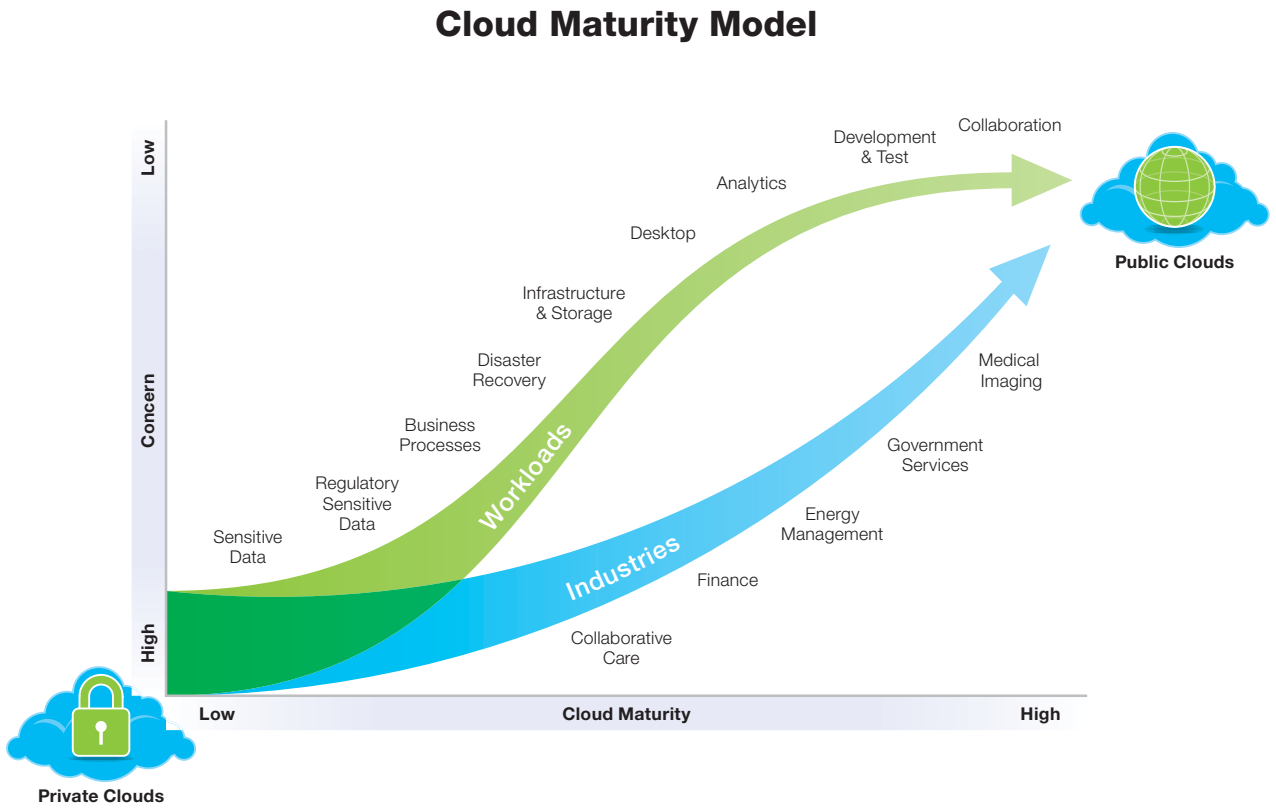


Figure 78: Cloud Maturity Model

Design elements for security in the cloud

Secure by design

Cloud computing is largely driven by financial and operational efficiency motivators. As a result, organizations should build security into the fabric of their cloud activities to achieve the expected returns. Retroactive attempts to apply security later in the cloud life cycle often result in diffusing the value of cloud computing. For example, if an organization wants to use the cloud, public or private, as a platform for delivering cloud-based application services, but has not ensured that the targeted application has been securely designed and implemented, then regardless of the controls their provider has put into place, the application vulnerabilities could leave the solution open to unintended data loss or compromise. Extending existing security policies and standards, leveraging sound physical security protections already in place, and assessing systems and applications for security weaknesses are examples of security design elements that should be included when establishing a secure cloud environment.

Purpose-built security

IBM believes that there is no "one-size-fits-all" approach to security within clouds. Rather there are common sets of foundational security controls which apply to all types of clouds. On top of these foundational controls, organizations should implement workload-specific controls that align with the work being done in that particular cloud. For example, in a cloud solution dedicated to workplace collaboration, anti-spam is certainly an appropriate and needed control. However, in a cloud designed for development, anti-spam probably is not a control necessary to reduce risk associated with the workload. This approach allows the cloud provider to address the specific security needs of each cloud solution and control costs, which should translate into cost savings for their subscribers. The delivery and deployment models (SaaS, IaaS, PaaS, etc.) can also determine the types of security controls that are appropriate based on differences in attributes such as data flow, integration points, and user access scenarios.

Improving security via the cloud

Although a vast amount of public attention has been given to the security risks of cloud computing, it is likely that for many organizations the cloud could be considered more secure than their traditional legacy environment. Cloud providers may contribute security capabilities and skills that subscribers do not or cannot support within their own organizations. Cloud adoption is typically aligned with specific initiatives, and as such, the security requirements are narrowly focused and thus can be more deliberate. As such, security can be applied more appropriately and effectively to that workload or task than was applied as part of the organization's enterprise-wide security program.

Cloud computing can also allow organizations to apply layers of security that they previously were not able to implement due to lack of skilled resources or budget by actually moving security as a workload into the cloud. Cloud-based security services not only can offer customers cost savings over performing that function in house, but may allow some organizations to take on new security controls that they otherwise would not have added to their security management program, such as ongoing vulnerability scanning.

> Security in the cloud is a product of ongoing due diligence rather than a point in time statement. Organizations should plan on engaging in security over the life cycle of their cloud activities with the same level of diligence they execute within their enterprise environments.

Cloud providers who understand security threats and are able to adapt as threats evolve, are best equipped to help subscribers strengthen their security posture via the cloud. Ongoing gap assessments against best practices for secure cloud computing and testing for weaknesses against external attack via penetration testing are ways that cloud providers can assess and maintain their security posture.

Organizations should understand the implications of their cloud initiative in terms of security and privacy. Organizations new to the cloud should look towards seasoned experts to help them consume cloud-based technologies and security vendors, like IBM, can help these organizations plot out their security requirements and help ensure that their security strategy for cloud computing is sound.

Cloud computing opportunities

As acceptance of cloud computing advances, we expect to see the cloud being used in new ways that can serve to advance security. For example, IBM is exploring the use of advanced analytics to help organizations identify threats to their environments and respond to those threats without impacting business value. These advanced analytics capabilities are being developed to allow the processing of millions of events per second to identify the key threats or the needle in a haystack which an organization should focus on from a security perspective. IBM is also leveraging social network concepts such as crowd sourcing to evaluate the impacts of collective group experiences and knowledge to identify and address vulnerabilities. Finally, IBM is evaluating emerging endpoints such as mobile technologies to provide protection from new avenues of attack against cloud subscribers.

Summary

As cloud adoption continues to grow, and cloud providers apply controls appropriate to the function and purpose of their cloud solution, acceptance of the cloud, even the public cloud, as a platform for handling increasingly sensitive and mission critical workloads is expected to grow. Building security into the foundation of each cloud initiative should be the joint responsibility of the cloud subscriber and their providers. This requires a deep understanding of the security requirements surrounding that initiative, and a commitment to meet those requirements without applying security controls that are unnecessary or ineffective. If vendors and subscribers are able to do this, then the efficiencies and cost savings that cloud computing affords can be better realized.

As security concerns quell regarding cloud computing, more organizations may take advantage of the security benefits that can be gained from cloud computing either as a beneficiary of the security controls the provider offers for their specific security initiative or as a consumer of cloud-based security services. In the meantime, organizations should continue to seek guidance from security vendors like IBM for help in evaluating and developing their cloud security strategy, assessing the controls around their cloud initiatives, and providing them with secure solutions for enabling cloud computing within their organization.

IBM Corporation
Software Group
Route 100
Somers, NY 10589 U.S.A.

Produced in the United States of America
March 2011

WGL03007-USEN-00