IBM X-Force 2012
Mid-year Trend and Risk Report

September 2012

IBM

Contributors

Producing the IBM 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 attention and contribution to the publication of this report.

Contributor	Title
Brian McGee	Visual Designer - User Experience Group/Usability
Bryan Ivey	Team Lead, MSS Cyber Threat and Intelligence Analyst
Carsten Hagemann	X-Force Software Engineer, Content Security
Chadd Horanburg	Cyber Threat Intelligence Analyst
Cynthia Schneider	Technical Editor, IBM Security Systems
David Merrill	STSM, IBM Chief Information Security Office, CISA
Dr. Jens Thamm	Database Management Content Security
Gina Stefanelli	X-Force Marketing Manager
Jason Kravitz	Techline Specialist for IBM Security Systems
Larry Oliver	Senior Cyber Threat/Security Intelligence Analyst
Leslie Horacek	X-Force Threat Response Manager
Marc Noske	Database Administration, Content Security
Mark E. Wallis	Senior Information Developer, IBM Security Systems
Mark Yason	X-Force Advanced Research
Michael Applebaum	Director of Product Marketing, Q1 Labs
Mike Warfield	Senior Wizard, X-Force
Nishad Herath	X-Force Advanced Research
Paul M. Sabanal	X-Force Advanced Research
Ralf Iffert	Manager X-Force Content Security
Randy Stone	Engagement Lead, Emergency Response Service
Rob Hall	Product Manager - Sterling Connect:Enterprise, Sterling Secure Proxy
Robert Freeman	Manager, X-Force Advanced Research
Rod Gifford	Product Marketing Manager, Sterling Connect:Enterprise, Sterling Secure Proxy
Scott Moore	X-Force Software Developer and X-Force Database Team Lead
Thomas Millar	Senior Incident Response Analyst

About IBM X-Force

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, IBM X-Force also delivers security content to help protect IBM customers from these threats.

DEDICATION

The IBM X-Force 2012 Mid-year Trend and Risk Report is dedicated to the fond memory of our friend and colleague, ***Don Hall****. The Director of Product Development, for Advanced Threat Platforms at IBM Security Systems, Don oversaw a global team of engineers, including the X-Force security research and development teams who contribute to the production of this report. A passionate champion for his team, and a dedicated technology leader, the contributions Don made to security and to IBM will be greatly missed.*

IBM Security collaboration

IBM Security provides a broad spectrum of security competency.

- The IBM X-Force research and development team discovers, analyzes, monitors, and records a broad range of computer security threats, vulnerabilities, and the latest trends and methods used by attackers. Other groups within IBM use this rich data to develop protection techniques for our customers.
- The IBM X-Force content security team independently scours and categorizes the web by crawling, independent discoveries, and through the feeds provided by IBM Managed Security Services (MSS).
- 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.
- IBM Professional Security Services (PSS) delivers enterprise-wide security assessment, design, and deployment services to help build effective information security solutions.
- The QRadar Security Intelligence Platform from Q1 Labs, an IBM company, offers an integrated solution for SIEM, log management, configuration management, and anomaly detection. It provides a unified dashboard and real-time insight into security and compliance risks across people, data, applications, and infrastructure.
- IBM Sterling Secure Proxy is a demilitarized zone (DMZ)-based application proxy that protects your file transfers from the public Internet. IBM Sterling Connect:Direct® is one of the leading solutions for secure, point-to-point file transfers. It has been optimized for high-volume, reliable data delivery of files within and between enterprises, and provides script-based automation, scheduling, and alert notifications for unattended 24x7 operations.

Contents

Contributors 2

About IBM X-Force **2**
IBM Security collaboration 3

Section I—Threats 6

Executive overview **6**

2012 highlights **8**
Threats 8
Operational security practices 9
Software development security practices 10
Emerging trends in security 10

IBM Managed Security Services—A global threat landscape **11**
Hand in hand: Cross-site scripting and SQL injection 11
Obfuscation 12

MSS—2012 top high-volume signatures **14**
SQL injection 15
SQL Slammer worm 16
PsExec_Service_Accessed 17
Directory Traversal 18
Cross-site scripting (XSS) 19
SNMP Crack 20
SSH brute force 21
HTTP Unix passwords 22
Shell command injection 23
Return of web browser exploitation 24

Trending in the dark—the afterglow of an attack? **25**
Spoofed denial-of-service attacks 25
Targets of denial-of-service attacks 27

Mac malware—major outbreak and targeted attacks **29**
Flashback 29
Mac APT 29
Conclusion 30

Web content trends **31**
Analysis methodology 31
IPv6 deployment for websites 31
Anonymous proxies 34
Malicious websites 36

Spam and phishing **38**
Spam volume stabilized at low level 38
Major spam trends during the last 12 months 39
Common top-level domains in URL spam 43
Spam country of origin trends 44
Spammers' weekend activities 45
Grum botnet take down in July 2012 46
Email scam and phishing 48

Section II—Operational security practices 52

Combating Advanced Persistent Threats (APTs) with security intelligence and anomaly detection **52**
Understanding advanced persistent threats 52
Security intelligence: Uniquely equipped to defend against APTs 54
Anomaly detection: The security intelligence lynchpin of APT defense efforts 56
Best practices for anomaly detection 57
Conclusion 57

Vulnerability disclosures in the first half of 2012 **58**
Web applications 58
Continuing decline in exploit count 62
CVSS scoring 65
Vulnerabilities in enterprise software 66
Wrap-up 69

Contents

Sandboxes: Another line of defense **70**
What is a sandbox? 70
How sandboxes work 70
Sandboxes can help you 71
What you can do now 71
What we can expect 72
Attackers will adapt 72
Final thoughts 72
Auditing made easier with UNIX shell history time stamping **73**
Evaluating the cyber terrain with OCOKA **77**
Observation 78
Concealment 79
Obstacles 80
Key terrain 81
Avenues of approach 82
Using perimeter security to take the risk out of file transfers **83**
Securing your perimeter 84
Best practices 86

Section III—Software development security practices 87
Email password—the keys to your personal online identity **87**
How important is your email password? 87
Once more into the breach 87
Why does this matter? 87
What happens next? 87
Forgot your password? Click here to reset 88
“Don’t use the same password on different sites” 88
Rules and regulations vs. the real world 88
What is a secure password? 88
An example 89
Remembering your passwords 89
Security questions 89
Two-factor authentication 89
Putting it all together 90
Secure password hashing—when faster is not always better **91**
When slower is better 91
Consider the options 92
A hash of a hash 92
More complex passwords 93
Go slowly 94
Faster, cheaper and powerfully parallel 95

Section IV—Emerging trends in security 97
Influences of initial bring your own device (BYOD) in most enterprises **97**
State of security 98
Making BYOD work 99
Identification and authentication 99
Access authorization 100
Information protection 100
Operating system and application integrity 100
Assurance 101
Incident response 101
BYOD program definition and review 101
Best practices in mobile security 102
State of mobile security technologies 102
Approach trends by industry 104
Mobile platform vulnerability management 104

Section I—Threats

In this section we explore threat-related topics and describe the enterprise attacks that security specialists face. We discuss malicious activity observed by IBM and how we help protect networks from those threats. We also update you on the latest attack trends that IBM has identified.

Executive overview

Early in 2011, IBM X-Force declared it the year of the security breach. Enterprises both large and small were targeted. In 2012, the trend has continued and the topic of security breaches quickly rose to the top of discussion lists from board rooms to blogs and to major media. Executives who were held accountable for critical corporate, customer, employee, investor and/or partner data wanted to reconcile and understand just how well they might be doing in this combustible environment of attack activity. They continued to ask the hard questions about how to secure an enterprise that is interconnected by means of cloud, mobile, and outsourcing technologies. They also asked who within the organization was managing security, so that steps towards a plan of action could be discussed.

As a security research organization, IBM X-Force has traditionally viewed security breaches with a technical focus. However, we have modified our view of attacks and breaches over time to encompass a greater business context. The overall breach trend continues into 2012, as several major high profile businesses have had to deal with the fallout of leaked passwords and other personal data. The healthcare industry in particular seems to be hit hard.

While security products and technology could have mitigated many of these unfortunate events, we are seeing more than ever how systems interconnectedness, poor policy enforcement, and human error, is far more influential than any single security vulnerability.

We've seen several headlines regarding cases where digital identities were decimated, not through malware, key loggers, password cracking, or even through access of the victim's computer or device. Instead, the bad guys accomplish their nefarious deeds by culling a small amount of personal data from public sources, using clever social engineering tricks and depending upon the loose policies of a handful of companies who we trust with our private data. Now, more than ever, the delicate balance between security, convenience, and privacy takes center stage.

In one case, attackers bypassed two-factor authentication—commonly thought to be almost failsafe—simply by convincing a mobile phone provider to relocate a user's voicemail, giving attackers the data they needed to reset a password. In another, the last four digits of a credit card number, which was easily visible on one site, was used by another service as a key piece of identification data, used to reset the account. For each one of these types of high profile incidents, there are similar breaches, going on beneath the radar.

Through the disclosure of breaches in 2012, we continue to see SQL injection reigning as the top attack technique. In addition, attackers seem to be taking advantage of cross-site scripting vulnerabilities for web applications. Over 51% of all web application vulnerabilities reported so far in 2012 are now categorized as cross-site scripting.

Even with all of this abundant attack activity, we have witnessed bright spots as well. Spam and phishing levels remain low with the take down of botnets in 2011, and as recently as July 2012, we witnessed yet another botnet take down with the removal of Grum. The data clearly demonstrates declines from this activity. Positive web trends continue with the adoption of IPv6 technology. Currently, enterprises and governments taking advantage of IPv6 find less malicious activity occurring, although we don't know when attackers will decide to adopt IPv6 technology.

At the mid-year point in 2012 we see an upward trend in overall vulnerabilities, with a possibility of an all-time high by year end. Even so, IBM X-Force data continues to demonstrate declines in true exploits, with only 9.7% of all publically disclosed vulnerabilities subjected to exploits. By making headway in certain areas, we find ourselves at a crossroad of change. Improvements in past software design and technology are combining with the adoption of new technologies such as personal mobile devices and tablets blending into the enterprise.

A more holistic approach to the entire ecosystem is required. Users should become more aware of how visible their personal data is online, more aware of who has access to it, and more aware of how it can be used against them. This affects not only their social networking, but also their choices of mobile application selection and usage. As an increasing trend, mobile applications are requiring a significant amount of permissions that dilute the ability of users to discern potentially malicious intent. Furthermore, as consumers and corporations move critical data into the cloud, it is even more important to audit and understand how this data is accessed.

We have moved from the office, to the corporate network perimeter, to linked businesses, to a world of interconnected devices and services. A lapse in policy or technology at any point in the system can and will shake the whole foundation. IBM X-Force is confident that our IBM X-Force Trend and Risk Report will help to arm you with the awareness you need to make the right decisions for your business.

Now let's consider some of the highlights that occurred in the first half of 2012.

2012 highlights

Threats

Malware and the malicious web

- Any major global event, whether it is an election or a catastrophe, will lead to Search Engine Optimizations (SEOs) created by many different people for a variety of goals, both genuine and malicious. The current headlines provide excellent sources of "bait" to use in spam, SEO attacks and phishing, or spear phishing campaigns. They are also excellent opportunities for attackers armed with web browser exploit kits such as Blackhole. (page 11)
- One method for completely subverting the victim's computer is to arm a trusted URL or site with a malicious payload via cross-site scripting vulnerabilities. The websites of many well-established and trustworthy organizations are still susceptible to non-persistent cross-site scripting. (page 11)
- Since the last report, we have seen steady growth in SQL injection, which is keeping pace with the increased usage of cross-site scripting and directory traversal commands, such as HTTP "DotDot" commands. These three exploit types become very powerful when they are used together. (page 11)
- At the end of 2011 we discussed how the emergence of new Mac malware variants will more and more resemble Windows counterparts. Looking back at the first half of 2012, it appears that we were correct. In the last few months we have seen major developments in the Mac malware world including the Flashback outbreak and the discovery of advanced persistent threat (APT) Mac malware. (page 29)
- In our last IBM X-Force Trend and Risk Report, we mentioned the technical difficulty in exploiting OS X software is a major factor in preventing mass exploitation. Flashback infections bypass OS security by using multi-platform exploits through Java vulnerabilities. That is, the exploitation technique and most of the code involved is the same, regardless of whether the target is Windows or Mac. Some security vendors have set up sinkholes to determine the number of Flashback infections, and estimates are as high as 600,000 machines. (page 30)
- Another major development in Mac malware in the first half of the year is the discovery of targeted malware (Mac APT). Some initial variants used Java exploit CVE-2011-3544 to spread. This exploit is the Java Applet Rhino Script Engine Vulnerability—the same one used by Flashback. This targeted malware's purpose is to steal user data. (page 30)

Web content trends, spam, and phishing

- IPv6 Day was June 6th 2012, with many organizations implementing permanent IPv6 deployments. While full adoption is still low, IBM X-Force data demonstrates that Web 2.0 and legitimate sites are currently the most IPv6 ready. Websites with content such as hacking sites, illegal drugs sites, anonymous proxies, pornography, and gambling sites have been slower to adopt IPv6. This might be because of the additional technical efforts that are required in order to be IPv6 ready, or possibly so they can continue to reach as many users as possible. (page 31-33)
- Anonymous proxy registrations continue to hold steady in the first half of 2012, with three times as many anonymous proxies newly registered today as compared to previous years. More than two thirds of all anonymous proxies ran on the .tk domain (the top-level domain of Tokelau, a territory of New Zealand). (page 35)
- The United States continues to reign as the top host for malicious links with more than 43% of all malware links hosted. Germany takes the second place position, hosting 9.2%. Rounding out the top list is Russia, at number three for the first time, and China dropping from the top of the list to number four. Nearly 50% of all malware links are placed on pornography or gambling websites. (page 36)

- At the end of 2011, we saw the rebirth of image-based spam. Spammers continued to use this type of spam until the end of March 2012. At one time, more than 8% of all spam contained an image attachment. (page 39)
- Another new trend surfaced around the size of spam. Traditionally, spam messages were purposely kept small, to ensure spammers could send out as much as possible given their bandwidth. Today, we are seeing very large-sized messages, with the bulk of the size coming from large sections of irrelevant Cascading Style Sheets (CSS). A current theory is the extra data is being used as a way of evading detection as it does not seem to affect the message data or formatting. (page 41)
- India remains the top country for distributing spam, dominating the top of the list and setting an all-time record by sending out roughly 16% of all spam registered today. The USA, which fell below 3% in the spring of 2011, has increased in the spring of 2012. The USA currently accounts for more than 8% in third place after Vietnam. Rounding out the top five are Australia and South Korea, with Brazil coming in at number six, responsible for 6% of all spam distributed in the first half of 2012. (page 44)
- On July 18th, 2012, we witnessed the take down of the Grum botnet. Grum preferred clients in the USA, Vietnam, Australia, Germany, and Brazil, with these countries sending out 29.9% of the worldwide spam before the take down, but only 22.5% afterwards. (page 47)
- At the end of 2011, we began seeing the emergence of phishing-like emails that link to websites that do not necessarily perform a phishing attack. In 2012 this activity continued where parcel services were widely used to dupe users reaching more than 27% of the scam and phishing volume. Phishers also turned attention to nonprofit organizations, accounting for 66% and then dropping to 7% in the first two quarters of 2012. (page 49)

Operational security practices

Vulnerabilities and exploitation

- In the first half of 2012, we reported just over 4,400 new security vulnerabilities. If this trend continues throughout the rest of the year, the total projected vulnerabilities would be slightly more than the record we saw in 2010 approaching 9,000 total vulnerabilities. (page 58)
- The decline of reported SQL injection vulnerabilities continued in 2012 but cross-site scripting vulnerabilities increased again to a projected all-time high. Cross-site scripting is a term used to describe web application vulnerabilities that allow attackers to inject client-side script into web pages that are viewed by other users. Over 51% of all web application vulnerabilities reported so far in 2012 are now categorized as cross-site scripting. (page 59)
- IBM X-Force catalogs two categories of exploits. Simple snippets with proof-of-concept code are counted as exploits, but fully functional programs that can attack a computer are categorized separately as “true exploits.” The declining trend of true exploits continues into 2012, where based on data from the first six months, we project that only 9.7% of all publicly disclosed vulnerabilities will contain exploits. (page 62)
- IBM X-Force observed that vulnerabilities in Office and Portable Document Formats (PDF) declined sharply. IBM X-Force is confident that there is a strong relationship between the decline of PDF disclosures and the Adobe Acrobat Reader X sandbox. (page 67)

- IBM X-Force has seen great strides in the rate of patched vulnerabilities of the top ten vendors, which can be attributed to secure development practices and the continued implementation and improvement of Product Security Incident Response Team (PSIRT) programs. The top ten vendors have an impressive patch remedy rate of just over 94% of all vulnerabilities disclosed. (page 67)
- The rate of unpatched vulnerabilities (excluding the top ten vendors) for the first half of 2012, were the highest that IBM X-Force has seen since 2008. 47% of all vulnerabilities disclosed this year remain without a remedy, but this is primarily due to non-enterprise software. (page 68)

Software development security practices

Email password security

- The connection between websites, cloud-based services, and webmail provides a seamless experience from device to device, but users should be cautious about how these accounts are connected, the security of their password, and what private data has been provided for password recovery or account resetting. (page 87)
- Given the speed of password recovery tools, weak passwords can be discovered from leaked database hashes in seconds. The best solution for web developers is to use a hashing function that is designed for secure password storage. It should use a salt and the hash transformation itself should take a relatively long time, making it much more difficult to recover plain text passwords. A salt is just an additional element, such as a random string of text combined with the password before it is sent to the hashing function. (page 91)

Emerging trends in security

Mobile malware

- In the first half of 2012, reported mobile vulnerabilities and exploits are down to the lowest levels since 2008. IBM X-Force thinks there are multiple reasons for this. First, mobile operating system developers continue to invest in both in-house discoveries of vulnerabilities as well as enhancements to their security models to prevent vulnerabilities from being exploited. As is typically the case with a relatively new area like mobile, we see a pattern. First there is a spike in discoveries, with easier bugs found quickly, and then the ones that are more difficult to exploit are left. There is often a lag in time between researchers and attackers discovering techniques to overcome previously perceived limitations. (page 64)
- The state of mobile device security is in flux. While there are reports of exotic mobile malware, such as TigerBot/Android. Bmaster on Android, and Zeus/ZITMO on multiple mobile platforms, most smartphone users still appear to be the most at risk from premium SMS scams and the like. These scams work by sending SMS messages to premium phone numbers in a variety of different countries automatically from installed applications. (page 98)

Mobile—bring your own device (BYOD)

- To make BYOD work within your company, a thorough and clear policy should be in place before the first employee-owned device is added to the company's infrastructure. This policy should cover all aspects of the relationship between the company and the employee's device, and include buy-in from all parties. (page 99)
- As mobile devices become a primary computing device for many—both in enterprise as well as the Internet at large—we might find that patching of vulnerable devices becomes a primary security concern, since this area has had the least progress made in the past year or so. (page 105)

IBM Managed Security Services—A global threat landscape

IBM Managed Security Services (MSS) monitors tens of billions of events per day in more than 130 countries, 24 hours a day, and 365 days a year. The global presence of IBM MSS provides a first-hand view of current threats and our analysts use this wealth of data to derive an understanding of the cyber threat landscape. This section provides updates on our view of the top threats that are discussed throughout this report. Threat trend identification is vital to establishing future security strategy and understanding the significance of the threats to our computing environment.

Hand in hand: Cross-site scripting and SQL injection

Any major global event, whether it is an election or a catastrophe, will lead to Search Engine Optimizations (SEOs) created by many different people for a variety of goals, both genuine and malicious. We have seen the effect on social media sites every time there is a disaster, or a celebrity event, or a political scandal. We witnessed many of these types of events this year including the 2012 election, London Olympic Games, and the much-cited Mayan Prophecy. All provided an excellent source of "bait" to use in spam, SEO attacks, and phishing, or spear phishing campaigns. They are also excellent opportunities for attackers armed with web browser exploit kits such as Blackhole.

One method of subverting a victim's computer is to provide a URL that sends the user to a vulnerable website the user trusts. Many websites of well-known and trustworthy organizations remain susceptible to non-persistent cross-site scripting, usually employing a specially crafted URL. With the growing use of HTML5, SQL injection on the client-side is now possible as well, since HTML5 has become the new de facto web access method. This means that attackers might be able to access local storage through the HTML5 thick features, and if there is a local version of a loaded SQL database, then SQL injection becomes another valid method for infecting the victim's computer.

Since our last **IBM X-Force Trend and Risk Report**, we continue to see steady growth in SQL injection, keeping pace with the growth of cross-site scripting, and directory traversal commands such as HTTP "DotDot" commands. These three exploit types become very powerful when they are used together. Because there are so many ways to mix these three techniques together, we do not enumerate all the methods currently in play.

What we can assert, however, is that SQL injection and cross-site scripting are growing rapidly as favored attack methods, and that our trending information matches the assertion. We will continue to watch all three events for opportunities to correlate and improve reporting on this new approach.

Obfuscation

In the world of cyber threats, obfuscation is a technique to hide or to mask the sources and methods of a security relevant event. New obfuscation methods are constantly evolving in an attempt to evade intrusion prevention systems (IPS) and anti-virus software. IBM Security Network IPS has special detection algorithms that assist us in monitoring these techniques around the world. The toughest type of obfuscation to deal with is based around encryption, because it limits what we can determine about the information that is being transmitted. On the other hand, encrypted information appearing in unexpected places is often a "tell" by itself, because it identifies a suspicious source and destination that require further examination.

Matching Trends Between Cross-site Scripting Events and SQL Injection Events
July 2011 through June 2012

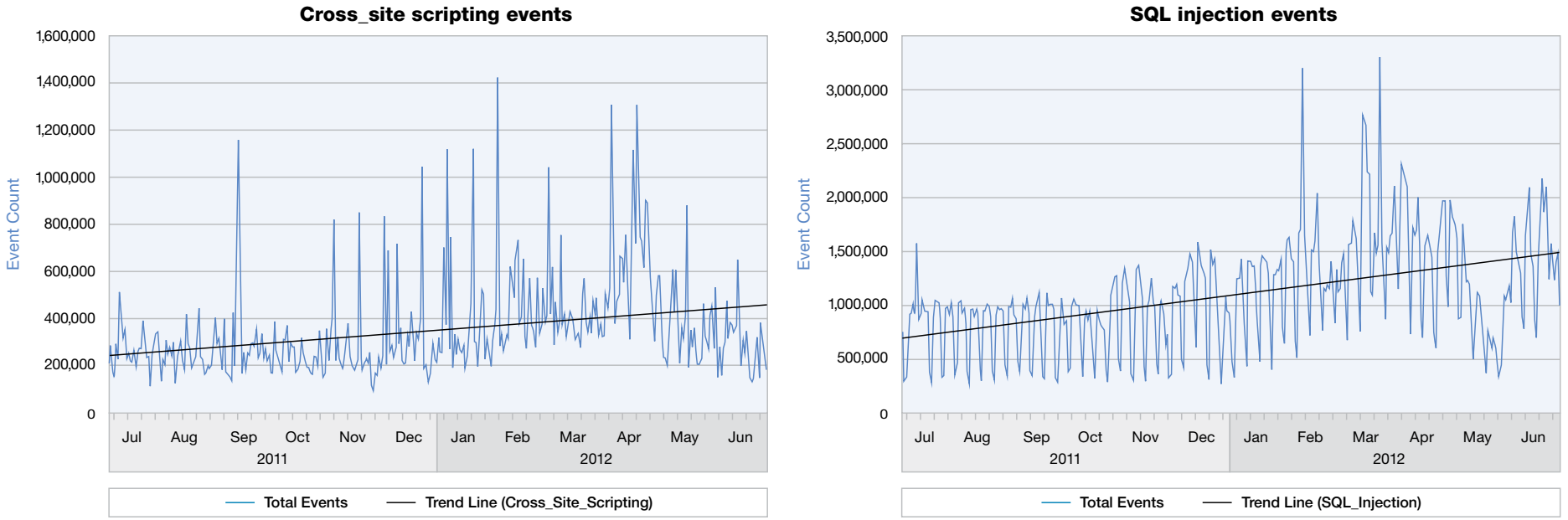


Figure 1: Matching trends between Cross-site Scripting Events and SQL Injection Events - July 2011 to June 2012

We are seeing the increased use of encryption by computer criminals to hide their exploits and to make it harder for network security systems to detect them. This includes HTTPS as well as native encryption features in various document formats and obfuscation using scripting languages. As the chart clearly demonstrates, the presence and volume of potentially obfuscated traffic is extremely variable, and extremely persistent. The image represents a composite of nearly 30 separate obfuscation heuristics. We expect that the use of obfuscation techniques will continue as technologies that identify exploits, malware, and data leakage improve. Additionally, as new applications are deployed, and as new technologies (cloud services, mobile applications, etc) emerge and influence how we communicate using the Internet, there will be more reason to hide potential attacks, raising the stakes each day.

We continue to develop and deploy techniques to keep pace with the growth of obfuscation techniques, and will continue to update our customers on these trends.

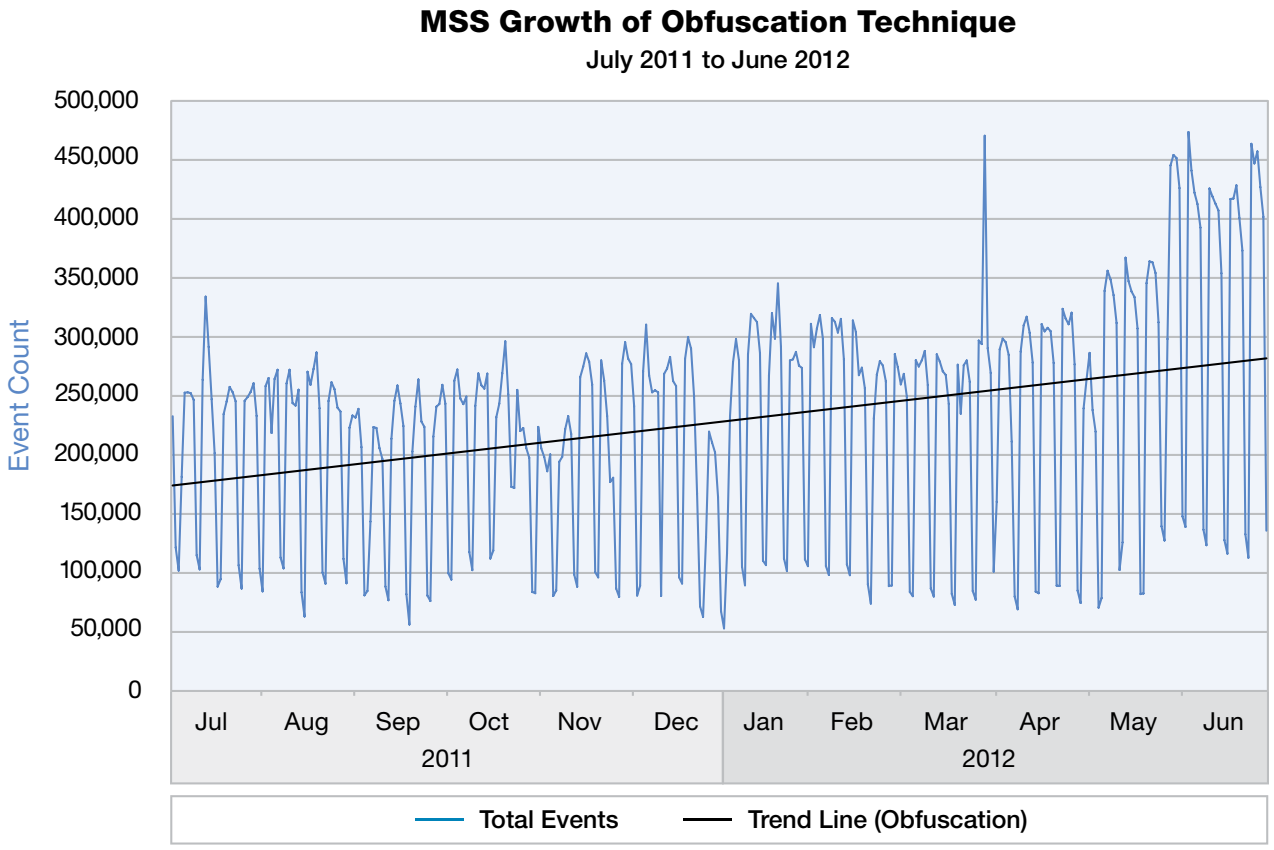


Figure 2: MSS Growth of Obfuscation Technique - July 2011 to June 2012

MSS—2012 top high-volume signatures

Table 1: The Top MSS high volume signatures table shows the relative placement of the ten most significant Managed Security Services signatures and their trending direction for 2012, as compared to year-end 2011 and year-end 2010. Seven of the top ten signatures from year-end 2011 have retained a spot on the 2012 mid-year list. We will highlight some of the significant changes first.

The downward trajectory of the SQL_Injection signature reversed in 2011 and continues to increase, thus retaining its position as the highest volume signature.

The SQL Slammer worm signature, SQL_SSRP_Slammer_Worm, has been on the decline throughout the year and may even drop off our top ten list during the next iteration of this report. We still do not know the exact reason for the dramatic and sustained decline.

At the same time, the PsExec_Service_Accessed signature has returned to the lineup of high volume signatures. This popular system administration tool is at position three after being absent from the list for a year.

Like several of the other signatures, the volume of the HTTP_Get_DotDot_Data signature continues its upward trend, climbing from fifth highest position to fourth.

Event Name	2012 Rank	Trend	2011 Rank	Trend	2010 Rank	Trend
SQL_Injection	1	Up	1	Up	2	Down
SQL_SSRP_Slammer_Worm	2	Slightly Down	3	Slightly Down	1	Down
PsExec_Service_Accessed	3	Slightly Up			3	Slightly Up
HTTP_GET_DotDot_Data	4	Up	5	Up		
Cross_Site_Scripting	5	Slightly Up	6	Slightly Up		
SNMP_Crack	6	Down	4	Down		
SSH_Brute_Force	7	Slightly Up	7	Slightly Up	4	Slightly Up
HTTP_Unix_Passwords	8	Up	8	Up	6	Slightly Up
Shell_Command_Injection	9	Slightly Up	9	Up		
JavaScript_Shellcode_Detected	10	Up				

Table 1: Top MSS High Volume Signatures and Trend Line - 2012 H1

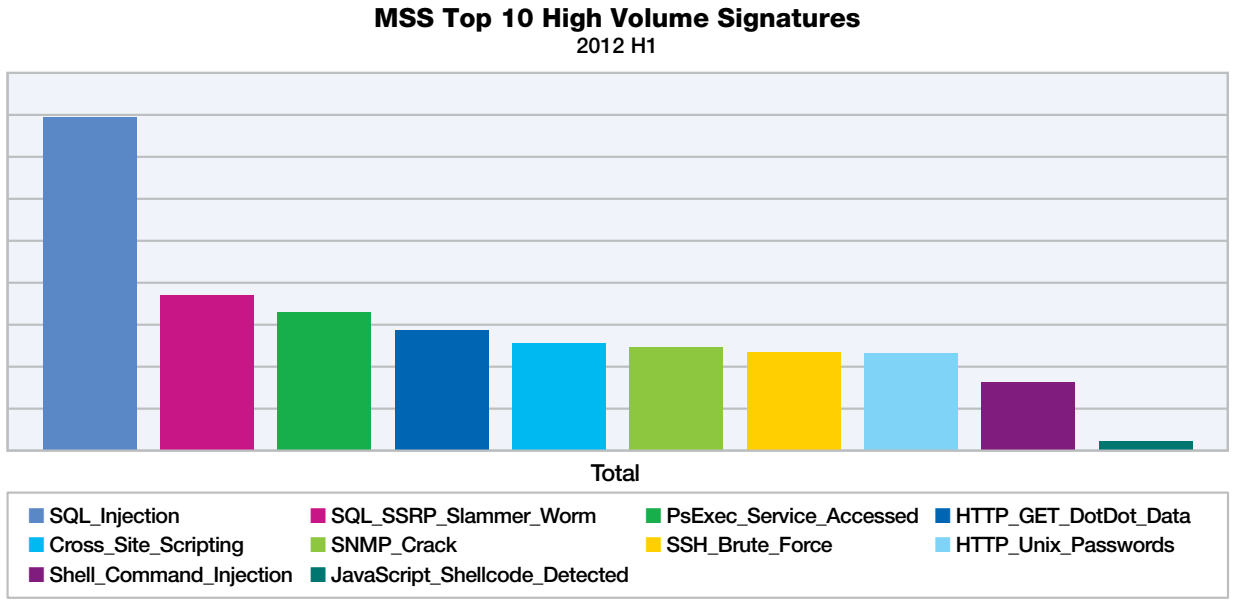


Figure 3: MSS Top 10 High Volume Signatures - 2012 H1

SQL injection

The SQL_Injection signature ranked second highest in 2010, and climbed to highest in 2011 with an indication of a continued upward trend. 2011 was a banner year for exploiting SQL weaknesses. At year-end, the trend line for SQL injection activity had begun to flatten out as the hacktivist activity had begun to quiet down. There was the usual jump around the retail holidays in November and December, but the trend was looking to flatten out.

The hacktivist groups, Anonymous and Lulzsec, had a major presence in SQL injection tactics early in 2011 and continued to hone their skills with new injection attack vectors. However, their activity levels had entered a brief lull that was recognizable.

Using tools such as LizaMoon, the attacker community made strides during 2011 in automating the identification of potentially weak systems and has continued to refine their exploitation methods. In 2012, we are seeing even higher levels of SQL injection attempts and the expansion rate of this type of attack appears to be higher than at the end of 2011. The net result of all this activity has kept SQL injection in the highest position for the first half of 2012.

The IBM X-Force 2011 Year-End Trend and Risk Report contains a section, "The Continuing Threat of SQL Injection," that provides additional insight into the SQL injection threat and identifies actions that can be taken to help protect against attack. This article should be required reading for anyone unfamiliar with this attack and its associated exploit mechanisms.

As discussed earlier in this report, attackers continue to combine different technologies together, creating a layered attack from which they may have a greater chance of success and can be difficult to defend against. SQL injection is one of the most common exploits found in these tool kits, especially when combined with other commonplace exploits, such as shell command injection, or cross-site scripting.

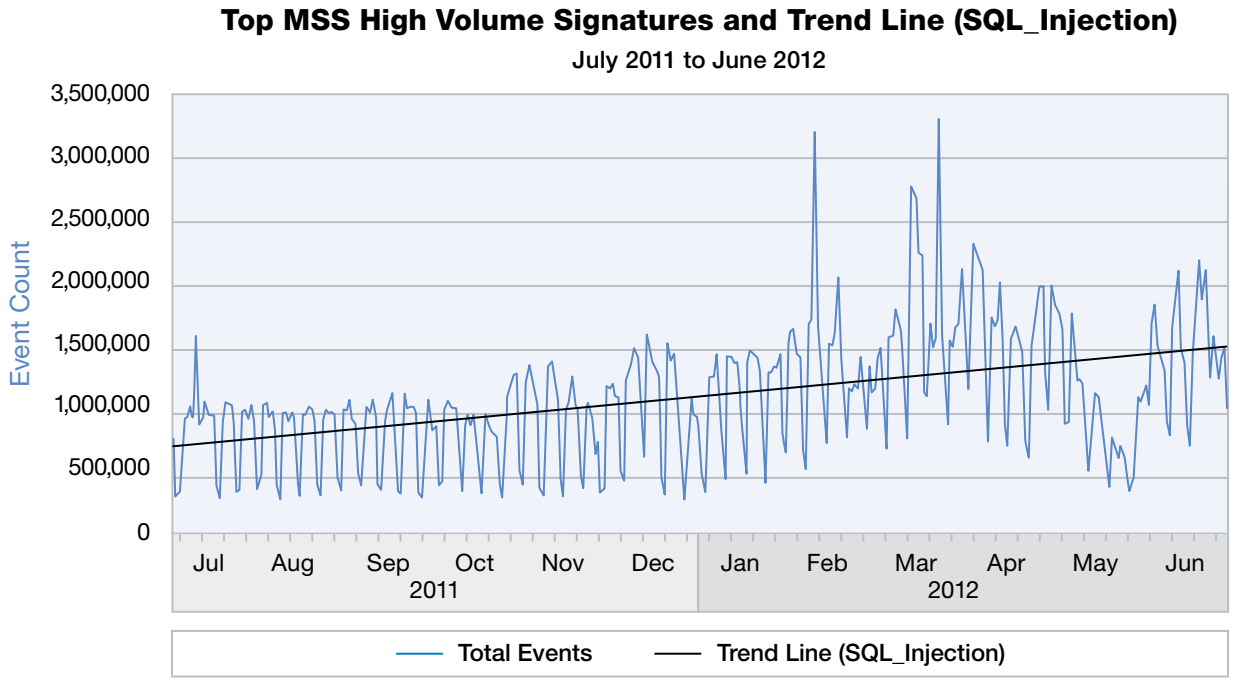


Figure 4: Top MSS High Volume Signatures and Trend Line (SQL_Injection) - July 2011 to June 2012

SQL Slammer worm

The second most common signature we've seen relates to the SQL slammer worm. The SQL Slammer worm has proven to be one of the most durable examples of Internet malware. The end of January of 2012 marked the ninth anniversary of the release of the Slammer worm. But Slammer does seem to be fading away. As discussed in the **IBM X-Force 2011 Mid-year Trend and Risk Report** article "The day that SQL Slammer disappeared", SQL Slammer activity dropped precipitously in March 2011. Since that time Slammer has almost completely disappeared. Although Slammer is currently ranked third in the 2012 Mid-Year Report, it might be completely gone by the time the next report is released. As forecasted, the drop continues, and likely will be removed from the Top High Volume Signatures list in the next report.

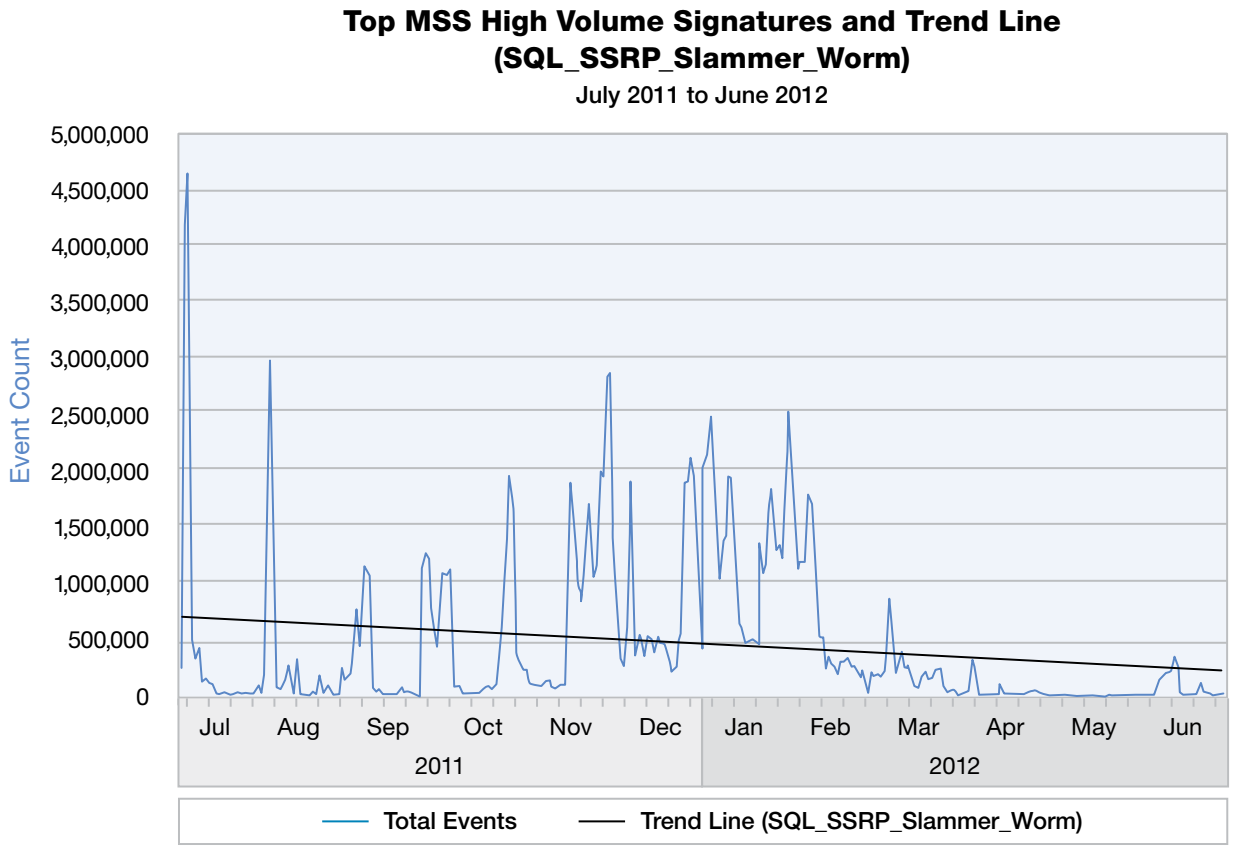


Figure 5: Top MSS High Volume Signatures and Trend Line (SQL_SSRP_Slammer_ Worm) - July 2011 to June 2012

PsExec_Service_Accessed

The signature in the fourth spot, PsExec_Service_Accessed, is a bit of a "blast from the past" because it placed third on the list of High Volume signatures at the end of 2010.

Note that the PsExec software is a part of a legitimate application package, owned by Microsoft and supported as part of Windows Sysinternals. It is a command-line based remote administration tool, much like a lightweight version of telnet, and functions properly without installing any code on the target system. PsExec handles the whole thing.

However, worms and advanced threats sometimes 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 the Sysinternals suite of software is used in your organization, you should ensure that best security practices are employed.

Our heuristic signature detects the invocation of the PsExec server handler and will report attempts to use the tool. This does not always mean that an attack or any malware has been detected, but any time this signature shows up, it is a good idea to confirm that the use is appropriate.

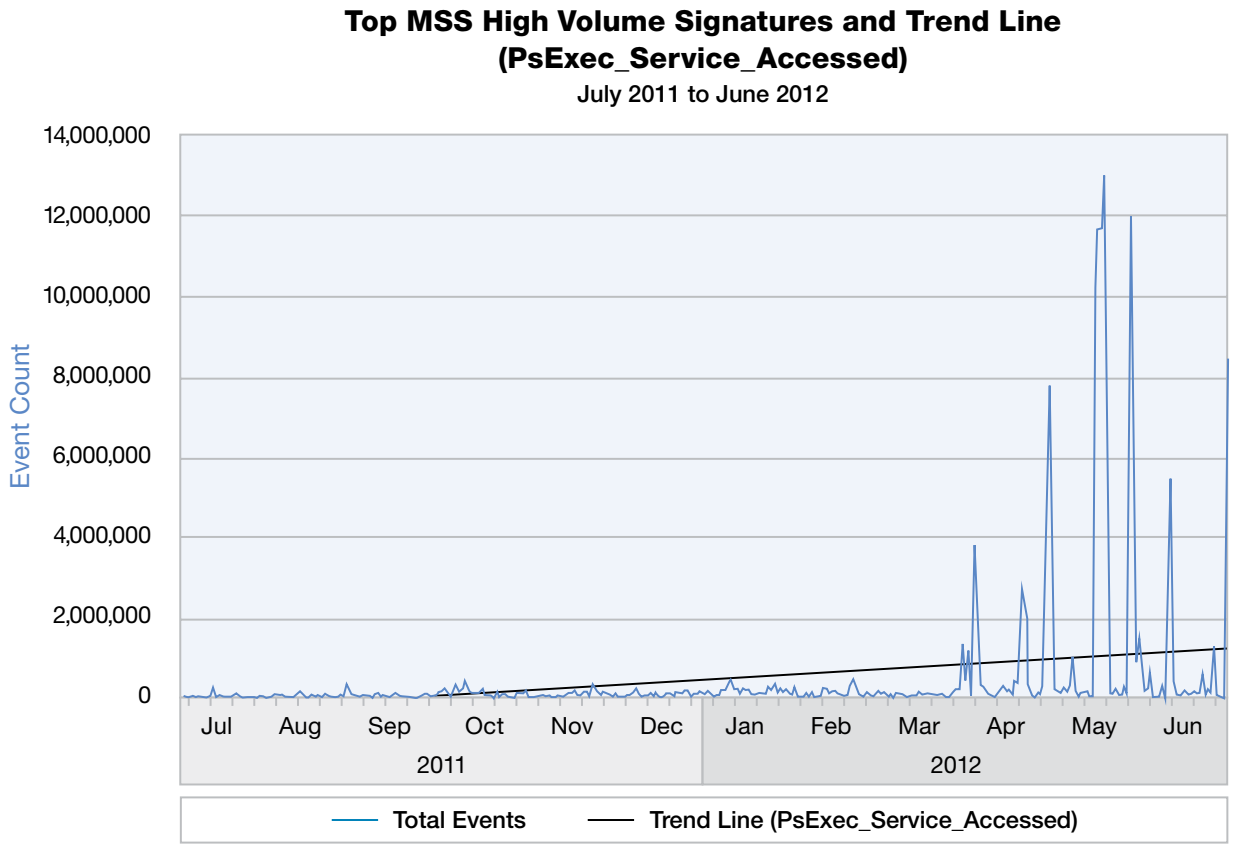


Figure 6: Top MSS High Volume Signatures and Trend Line (PsExec_Service_Accessed) - July 2011 to June 2012

Directory Traversal

The fourth most common signature we've seen is HTTP_GET_DotDot_Data and its relation to the directory traversal attack method. This is a truly old attack method, but it is still quite effective because it is based on the persistent features of most operating system shells.

This allows an attacker to traverse directories on vulnerable web servers. The ability to move from directory to directory can provide a lot of information to the attacker about the location of the programs on the server.

The only credible defense against this technique is to filter incoming user input to identify and disable unwanted abilities, and to restrict the access privilege level of the web-serving processes.

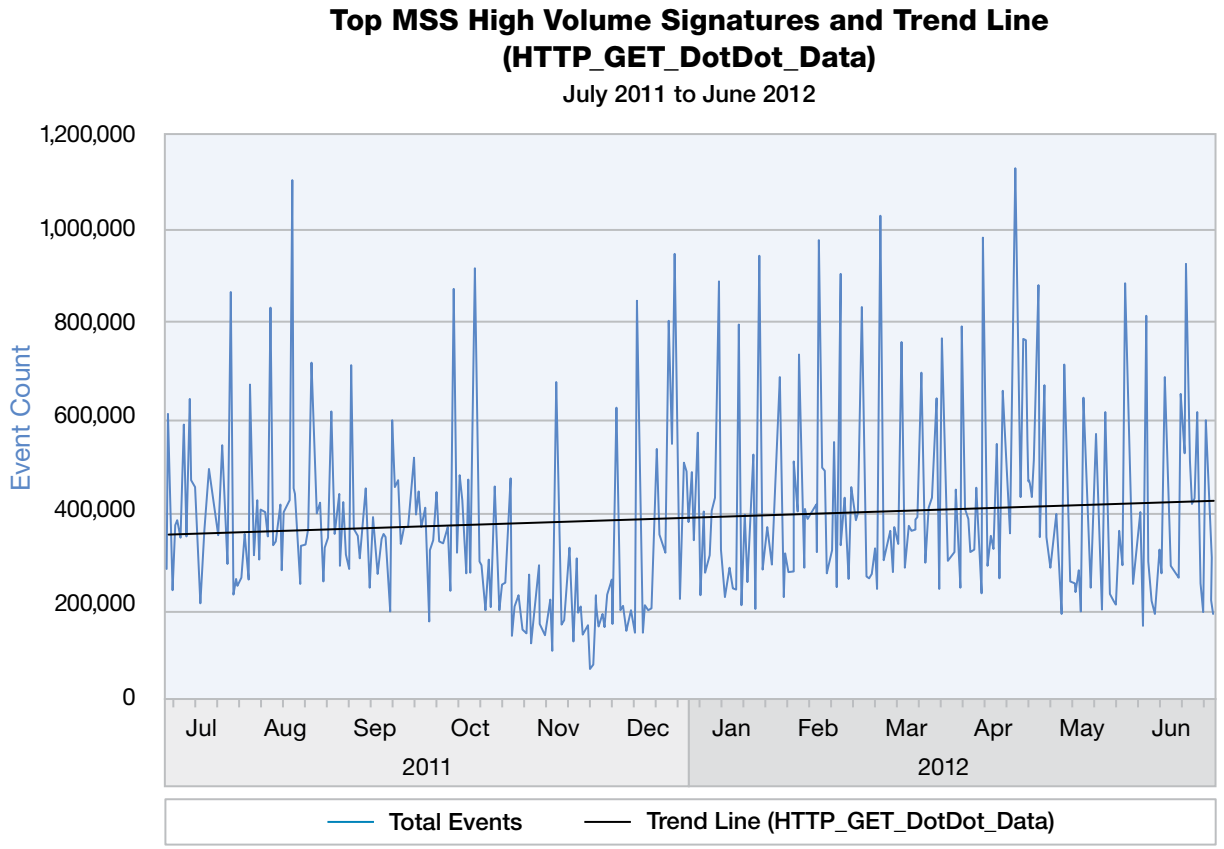


Figure 7: Top MSS High Volume Signatures and Trend Line (HTTP_GET_DotDot_Data) - July 2011 to June 2012

Cross-site scripting (XSS)

Cross-site scripting has been one of the most persistent exploits of the Internet era. A cross-site scripting attack injects client-side script into web pages, potentially subverting the client computer. This attack works on any web browsing technology, including mobile devices. The attack is extremely popular and can pose a significant security risk.

First documented in 1999, cross-site scripting was originally a problem unique to the Unix environment. Before the year was out a second variant of the exploit was documented. At this time, there are more than 6,000 variants of this vulnerability, with uses ranging from hijacking a browser session to a total system web-server-based takeover.

The Cross_Site_Scripting signature falls into position number five in our list of the top ten signatures tracked by volume. Reducing exposure to this risk normally involves careful validation of the server side code. New browser technologies show some promise for reducing the effectiveness of this vulnerability and user education on the client-side is vital.

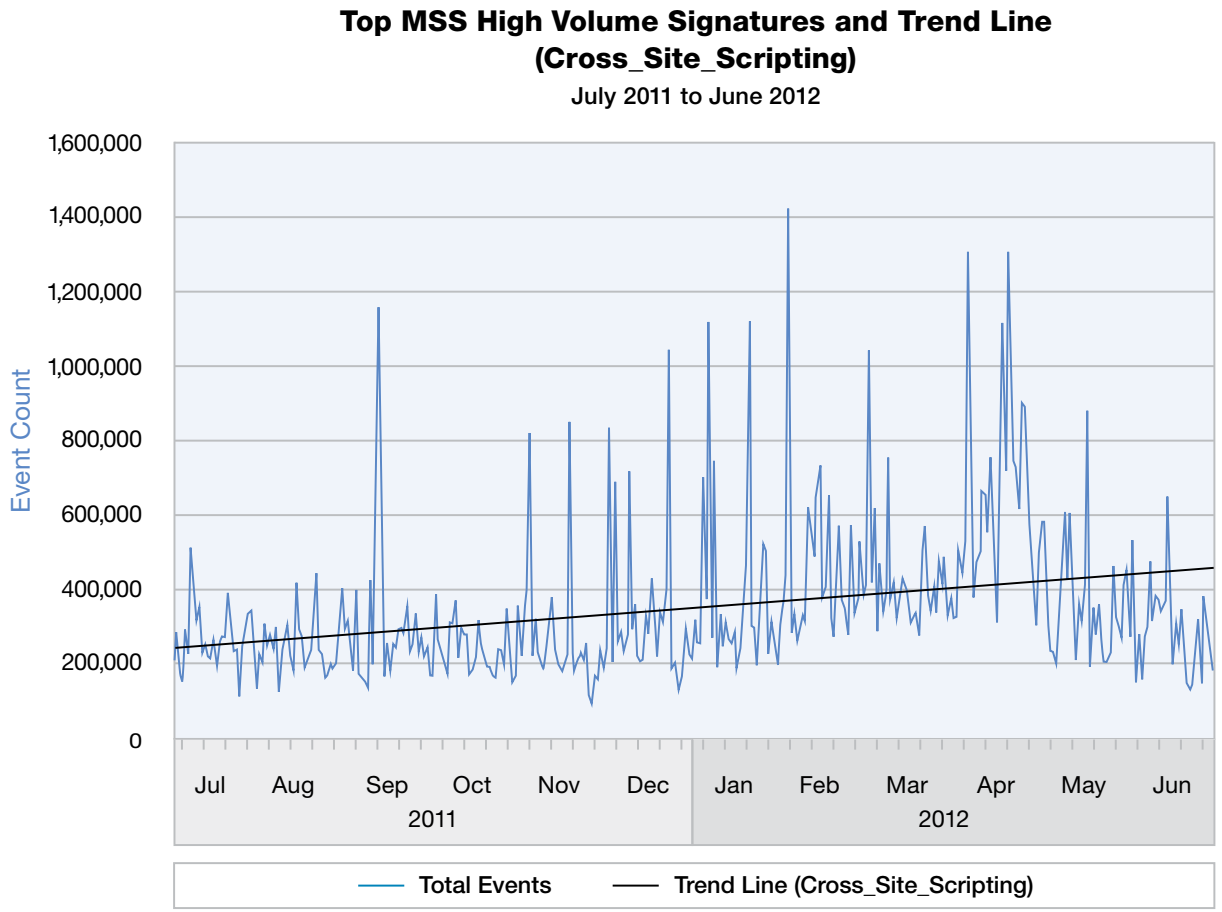


Figure 8: Top MSS High Volume Signatures and Trend Line (Cross_Site_Scripting) - July 2011 to June 2012

SNMP Crack

The SNMP_Crack signature is one of several signatures that are geared to detect brute-force attempts against fairly weak security. The Simple Network Management Protocol (SNMP) was developed for use within a trusted environment and community strings were intended to help keep things sorted out, not to provide authentication across public networks. Intended to assist network administrators, SNMP can be found on operating systems, hubs, switches, and routers in an Internet Protocol environment.

The SNMP_Crack signature is triggered when a large number of SNMP messages with different community strings are detected in a short time period. This is a suspicious finding, one that might indicate a brute-force community string guessing attack. As a matter of best practices, SNMP is normally prohibited through firewalls to prevent an external entity from using SNMP to perform a discovery scan on your protected network.

Since SNMP services are configured with default community strings, a potential attacker can search for those community strings first. Failing to gain information using default strings, the attacker might attempt a brute force search for valid community strings. Unless it is absolutely required, we recommend that SNMP be blocked at the external perimeter. We also recommend that the need for SNMP be evaluated in its entirety, and that you consider disabling the protocol if it is not required. If it is truly required, then consider migrating to SNMPv3 for stronger authentication.

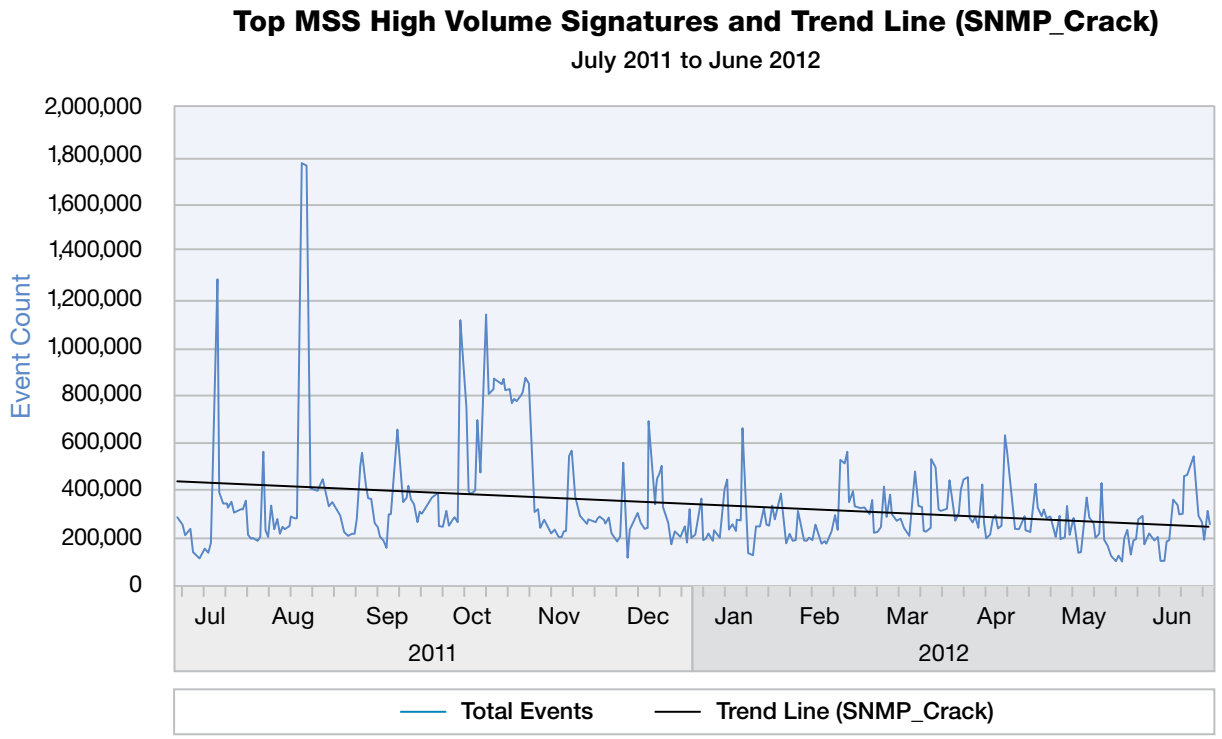


Figure 9: Top MSS High Volume Signatures and Trend Line (SNMP_Crack) - July 2011 to June 2012

SSH brute force

While the growth of this event was significant at the end of 2011, the activity level seems to have reached a plateau. Much like HTTP_Unix_Passwords, this signature is not absolutely indicative of an attack, but bears watching.

Because of the nature of this signature, it is not possible to tell whether a brute force or dictionary style attack might be happening since all of the traffic to be examined is encrypted. So this signature is about large numbers of SSH Server Identifications happening in a short period of time from a specific source address. Depending upon configuration, this could be a vulnerability scanner checking a system, a tool checking for weak passwords, or a full on brute force dictionary style attack. Since we cannot see inside the encrypted packets there is no proper way to determine the intent of a small amount of activity. Counts of the Server Identification requests will tend to be an indicator of the sort of communication being attempted, with high counts being highly suspicious.

Our recommendations remain the same: disable direct login to privileged accounts, enforce username and password security, and consider multi-factor authentication for particularly sensitive systems.

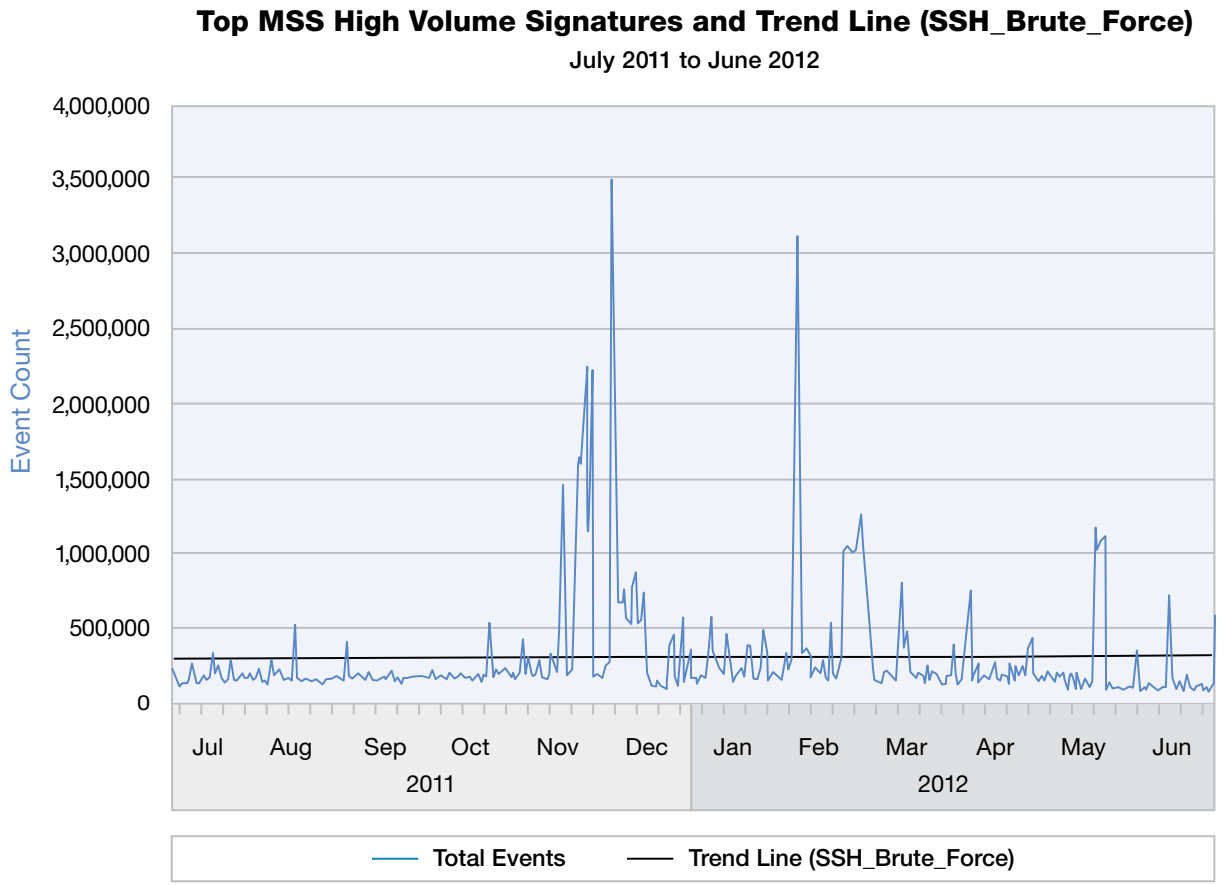


Figure 10: Top MSS High Volume Signatures and Trend Line (SSH_Brute_Force) - July 2011 to June 2012

HTTP Unix passwords

This signature identifies attempts to access the password (i.e., /etc/passwd and /etc/shadow) file on Unix systems through the HTTP protocol. While the HTTP_Unix_Passwords signature remains in the top high-volume list and continues to see an upward trend, it dropped from sixth place in 2010 to eighth place in 2011. It remains in eighth place in 2012. As with several of the other signatures, the event count continues to grow, but the sheer number of additional high-risk events are overtaking these signatures.

Relatively speaking, the HTTP Unix password attack is ancient, but it continues to be effective, which is partially why it continues to grow. Attempts to gain access to /etc/passwd can be made through many protocols, so other signatures, such as HTTP_Unix_Passwd_File_Accessed or FTP_Unix_Password_File_Accessed, might also be present. Clearly, gaining access to system password files, and then attempting to break these protections with hash tools, rainbow tables, or brute force attacks are still considered to be worthy pursuits, and yield desirable results for the attackers.

The HTTP_Unix_Passwords signature remains in the top high-volume list and continues to see an upward trend, currently holding the number eight spot in the list.

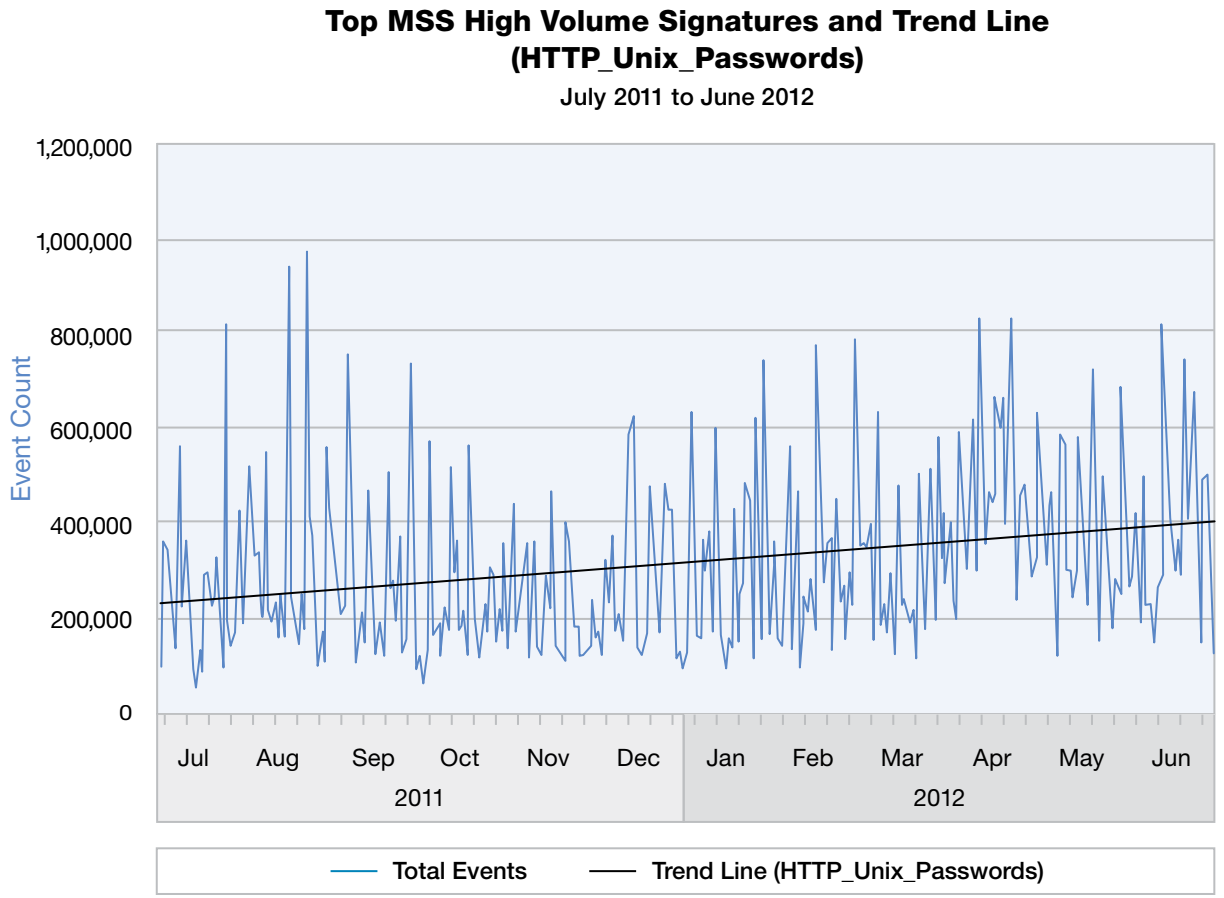


Figure 11: Top MSS High Volume Signatures and Trend Line (HTTP_Unix_Passwords) - July 2011 to June 2012

Shell command injection

Shell command injection is a form of Remote Command Execution (not to be confused with Remote Code Execution) that has become a steady presence across all customer types. Managed Security Services (MSS) is seeing a slow but very steady growth of these attack attempts, and we anticipate further growth of them.

Like SQL injection, this is an easy way for an attacker to gain a foothold on a server. Once that foothold is established, the attacker can gain a strategic advantage that provides a launching point for attacking other systems from inside the perimeter defenses. Exploitation is pervasive and far too frequently successful. Already compromised machines running "rogue" PHP (such as C99 Shell) also tend to be exposed through the same heuristic described below. C99 is a remote administration tool that is not exclusively malicious, but is often preferred by attackers since it is readily available.

The Shell_Command_Injection signature is a set of heuristics to detect Unix shell command injection attempts by scoring various combinations of commands and symbols commonly used when executing shell commands. In the default configuration, shell commands are scored only when a tuning parameter is matched, or when a directory traversal attempt is detected. In both of these cases, an attempt is made to score shell commands and symbols.

The primary defense against this attack is to validate user input to the server, eliminating shell commands. Restricting or eliminating the server software access to shell commands (like wget, passwd, dir, ls, and so on) can also reduce the effectiveness of the attack in case it succeeds.

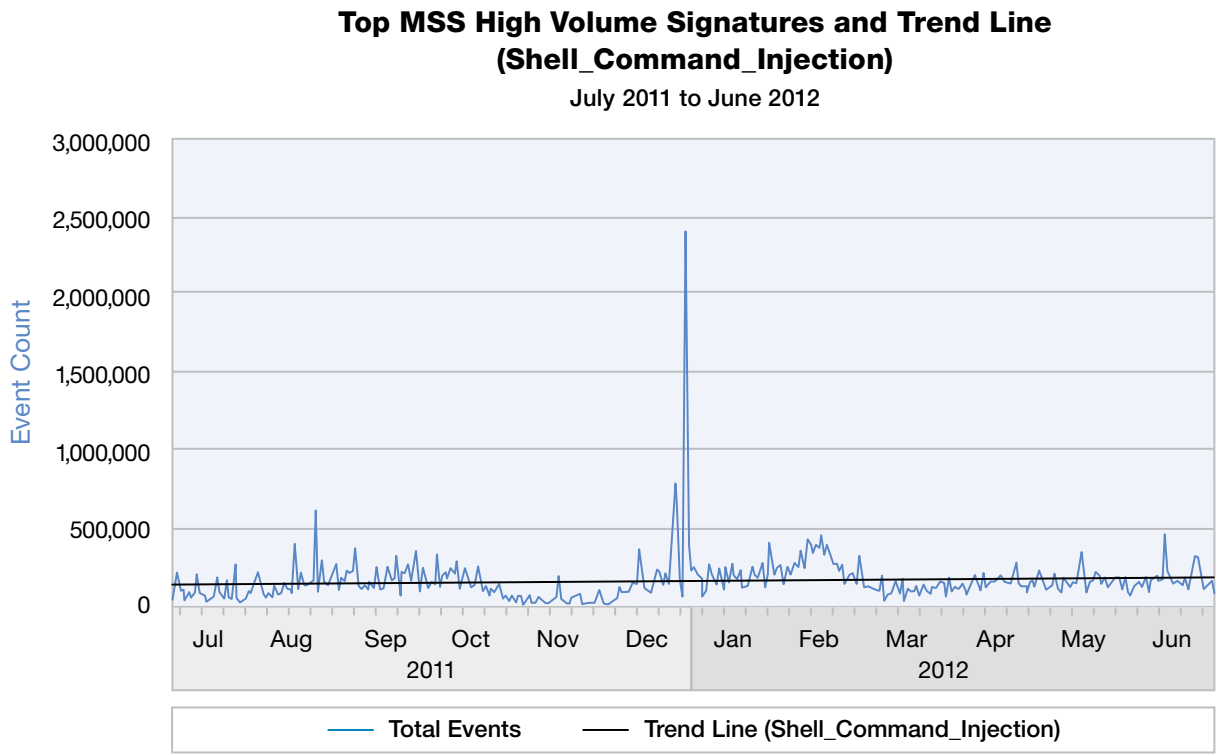


Figure 12: Top MSS High Volume Signatures and Trend Line (Shell_Command_Injection) - July 2011 to June 2012

Return of web browser exploitation

Recently, we identified a spike in browser exploitation though increased reports of the JavaScript_Shellcode_ Detected signature. This signature detects transmission of machine code encoded within JavaScript and presumed to be shellcode to exploit a vulnerability that may or may not be known yet. In other words, this is part of our "ahead of the threat" coverage. We have observed extremely small numbers of false positives with this signature since its inception in 2006. A dramatic increase in this signature being triggered is probably due to a rise in web browser exploit toolkits. In turn, this may be due to an increase in web application attack campaigns looking for vulnerable servers to serve malicious links and, in some cases, to serve the actual malicious code.

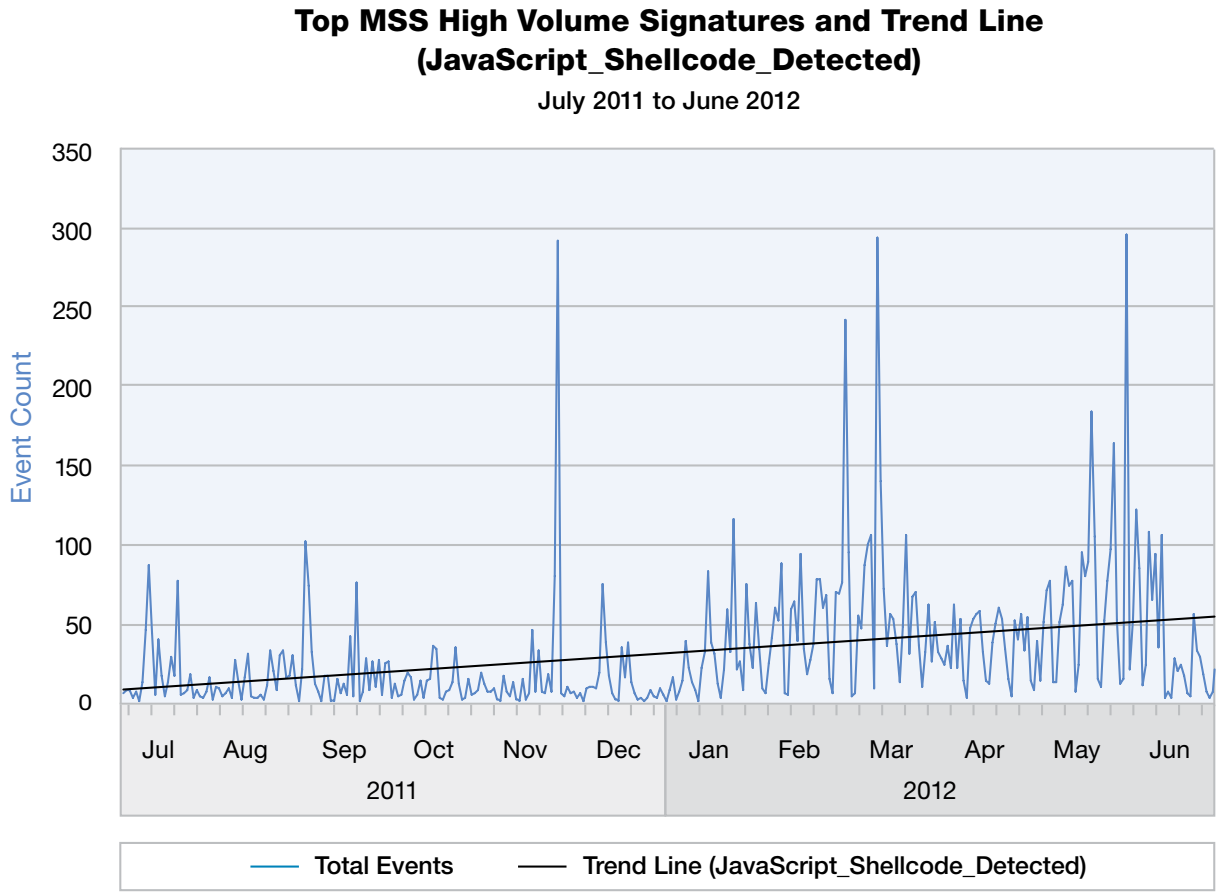


Figure 13: Top MSS High Volume Signatures and Trend Line (JavaScript_Shellcode_Detected) - July 2011 to June 2012

Trending in the dark—the afterglow of an attack?

One of the many data resources that IBM security analysts use to determine trending is a darknet. A darknet is a large range of IP addresses on the Internet that have never had services running on them. A darknet is also referred to as a black-hole network or a network telescope. 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. Traffic coming into this address range is often associated with malicious activity. The 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 trends begin to emerge. The first trend is the gradual rise in backscatter activity (Figure 14). Backscatter is actually a side effect of spoofed denial-of-service (DoS) attacks. Attackers who launch denial-of-service attacks on the Internet often put fake 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 determine the origin of an attack, to effectively block it, or to distinguish it between the spoofed packets and legitimate packets from real users. This is often used in denial-of-service and distributed-denial-of-service (DDoS) attacks to hide the true origin of an attack and to evade simple packet filters. The victim's system may respond to these spoofed packets as if they were legitimate, and send a response to the faked addresses, possibly tying up resources on their systems. These responses are known as backscatter. If an attacker randomly selects an IP address in our darknet range, and the victim responds, we collect and archive that response. By studying these responses and their patterns, we can learn and track things about denial-of-service activity on the Internet.

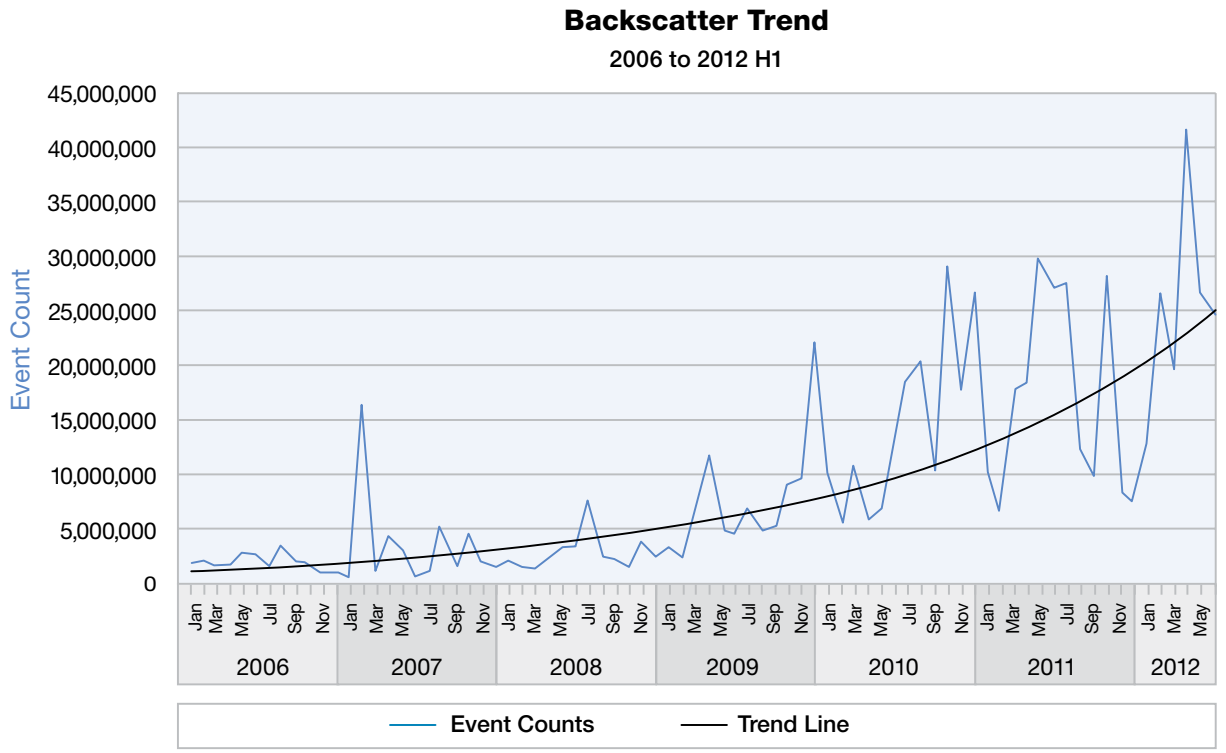


Figure 14: Backscatter Trend - 2006 to 2012 H1

In the IBM X-Force darknet, each SYN-ACK (response) backscatter packet that is received is likely an indicator that an attacker has sent a spoofed SYN (request) packet to a well-known service port on the machine under attack, spoofed from one of the IBM X-Force darknet addresses. While there has been a gradual increase in backscatter activity since 2006, there was a large jump between the years 2008 and 2009. Part of this increase was due to a significant spike in activity in 2009—the largest, percentage-wise, during this period. This trend of increased backscatter traffic continued into 2010, with another large jump, and on into 2011. At the close of Q2 2010, the average count for the first half of 2010 was slightly higher than the total average for 2009, at just over 16.5 million. At the close of the year 2010, we saw that this number had jumped to over 18 million. By mid-year 2011, we were seeing monthly spikes as high as 30 million. While the volume dropped off somewhat in the later part of 2011, 2012 has now seen backscatter spikes of up to 42 million. Figure 2 indicates the increase in annual volume from 2006 through 2011 of spoofed denial-of-service attacks on the Internet.

What can we deduce from this gradual rise and, in some instances, large jumps of backscatter activity? The majority of the backscatter data results from denial-of-service (DoS) attacks, so we can speculate that there has been a steady increase in spoofed DoS attacks since 2006. However, backscatter is subject to a high degree of variability due to the nature of what is collected and what is occurring. Some intense periods of backscatter may be the result of internecine warfare within and between various attacker camps. During this warfare, one group may attempt to block or take over the resources of another group. These "shelling matches" between warring camps can result in a sudden increase in backscatter traffic and in 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, December 2009 and, most recently, in April 2012 as shown in Figure 15.

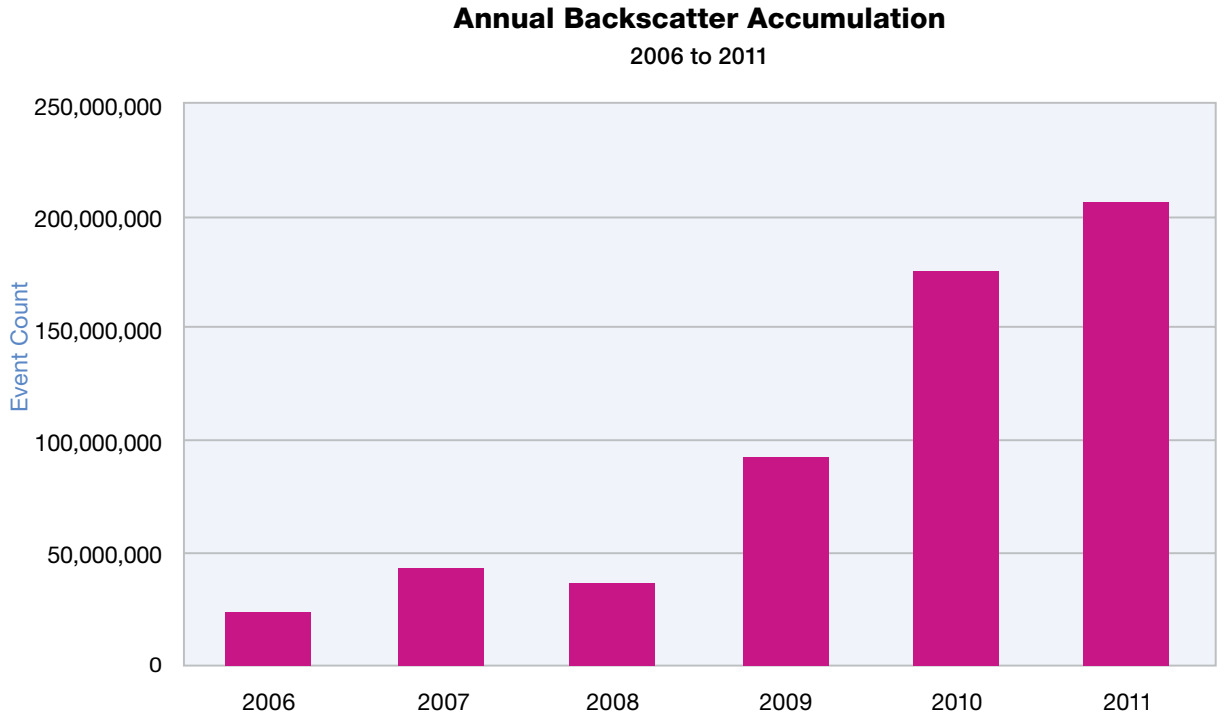


Figure 15: Annual Backscatter Accumulation - 2006 to 2011

Targets of denial-of-service attacks

The nature of a spoofed denial-of-service attack makes it difficult to determine the origin of the attack. 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 addresses. When looking at backscatter in the IBM X-Force darknet, it is clear that the origins of the attack are spoofed, but the target locale of the attack can be determined. Examining the sources of the backscatter provides information about the targets of spoofed denial-of-service attacks. Figure 16 shows the top target countries originating backscatter for the first half of 2012, as determined by using the WorldIP database that maps addresses to countries.

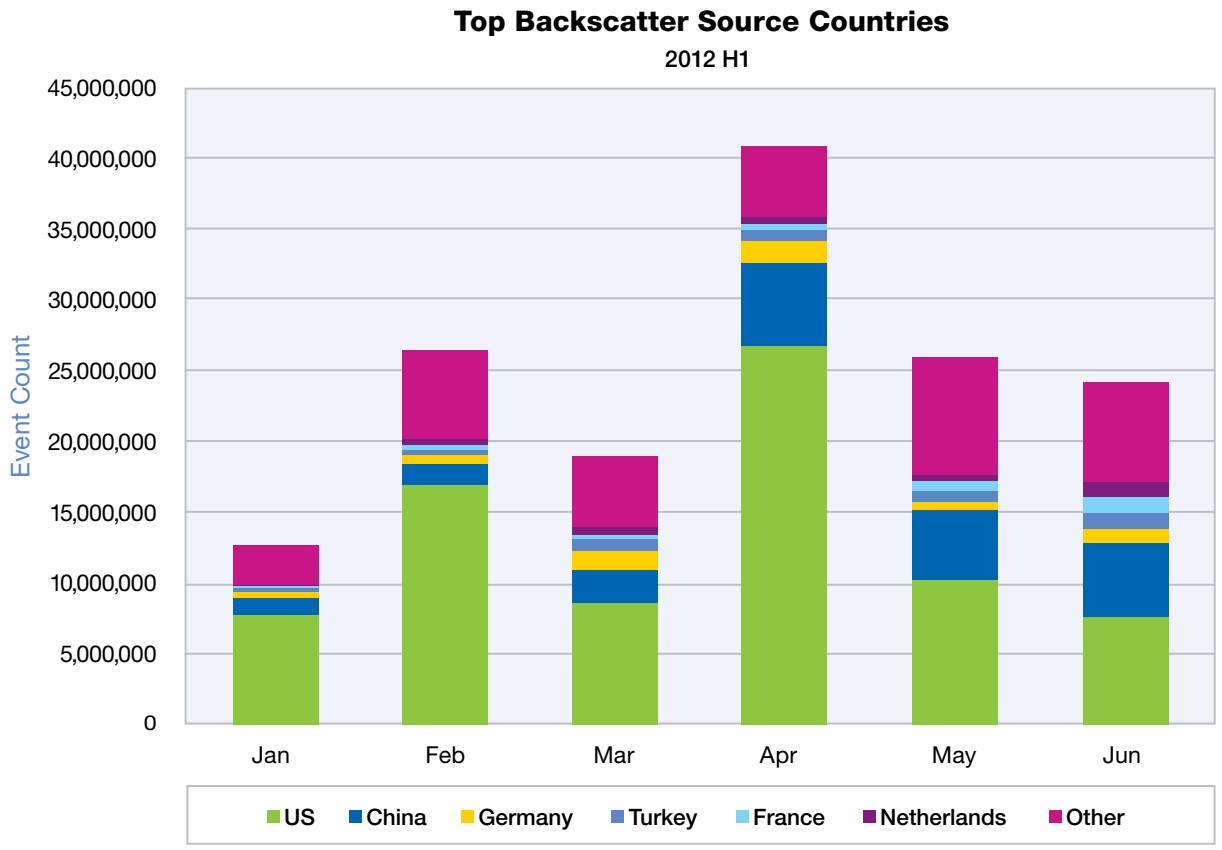


Figure 16: Top Backscatter Source Countries - 2012 H1

There is a fairly clear trend in the data. The United States is by far the largest source, China is second, and Germany is a distant third. Other individual countries are even farther behind. The United States and China have the first and second largest counts of IPv4 addresses allocated to them, so their ranking as backscatter sources isn't surprising. If any IP address is as likely to be a target as any other, then one would expect to see Japan, South Korea, or the UK in the top handful as well. But the attacks are highly variable and can strike anywhere as can be seen by the large number of "Other" hits representing targets other than the top countries in this chart and the counts drop off rapidly after the US and China.

In many cases, the "Other" category includes countries that we can determine, but which have lower counts than the top countries, while much of the "Other" category contains blocks of addresses for which accurate country code information is not available. Many of the IPv4 addresses in the "Other" category are legacy addresses from the early days of the Internet, when tracking data was not as clean, but which still represent a sizable chunk of the Internet address space.

Because the "Other" category includes the backscatter collected from all but the top six countries specifically identified in the chart, the totally height of each of the bars represents the total backscatter traffic collected. The large spike in April of 2012 clearly stands out, along with a smaller spike in February 2012. These spikes are well tracked by swings in the US backscatter activity, although correlation while the China backscatter is not so clear, with June of 2012 seeing the US and China much closer in backscatter traffic.

Mac malware—major outbreak and targeted attacks

In the last IBM X-Force Trend and Risk Report we discussed the emergence of Mac malware. We also predicted that more Mac malware would come out in 2012, and that this malware will resemble its Windows counterpart more and more. Looking back at the first half of 2012, it appears that we were correct.

In the last few months we have seen some major developments in the Mac malware world: the Flashback outbreak and the discovery of advanced persistent threat (APT) Mac malware. Let's take a closer look at these developments.

Flashback

The first variant of Flashback was discovered in September of 2011. Several variants were released after that, but the variants released this year were somewhat special. They share most of the features of the previous ones but what made them so successful this time is their method of delivery.

While the earlier Flashback variants relied on social engineering tactics to lure users to install them, the newer ones also employed drive-by-download techniques that are common in the Windows malware world. Flashback achieved this through compromised Wordpress blog sites that were modified to host redirect links to sites that contain the exploits.

In the last report, we mentioned that the technical difficulty in exploiting OS X software is a major factor in preventing mass exploitation. Flashback works around this by using multi-platform exploits through Java vulnerabilities. That is, the exploitation technique and most of the code involved is the same, regardless of whether the target is Windows or Mac.

Flashback first used two Java exploits back in February, CVE-2011-3544 (Java Applet Rhino Script Engine Vulnerability) and CVE-2008-5353 (Java Calendar Deserialization Vulnerability), but these exploits had been patched by then, and so this variant never achieved widespread infection. Things changed, however, when Flashback started using a CVE-2012-0507 (Java AtomicReferenceArray Type Violation Vulnerability) exploit in March. This vulnerability was already patched by Oracle the month before, but the Apple version of Java was not updated yet, leaving a lot of Mac machines vulnerable to this exploit. The resulting mass infection was enormous, and Flashback became the most widespread Mac malware to date. Some security vendors have set up sinkholes to determine the number of Flashback infections, and estimates are as high as 600,000 machines.

This Flashback outbreak also shed light on the main purpose of this malware, which is to earn revenue through click-jacking. After it is installed, Flashback hooks into the browser and intercepts either a Google ad click or a Google search. If a Google ad is clicked, the query parameters are sent to the Command and Control (C&C) server instead of a Google server. If a Google search is detected, the search parameters are sent to the C&C server, and then the C&C server responds with the Flashback author's own pay-per-click URLs, instead of the actual Google search results.

Mac APT

Another major development in Mac malware in the first half of the year is the discovery of targeted malware.

First there's the Tibet malware, which was discovered in March. The first variants used Java exploit CVE-2011-3544 (Java Applet Rhino Script Engine Vulnerability), also used by Flashback, to spread. Its main purpose is to copy and download a

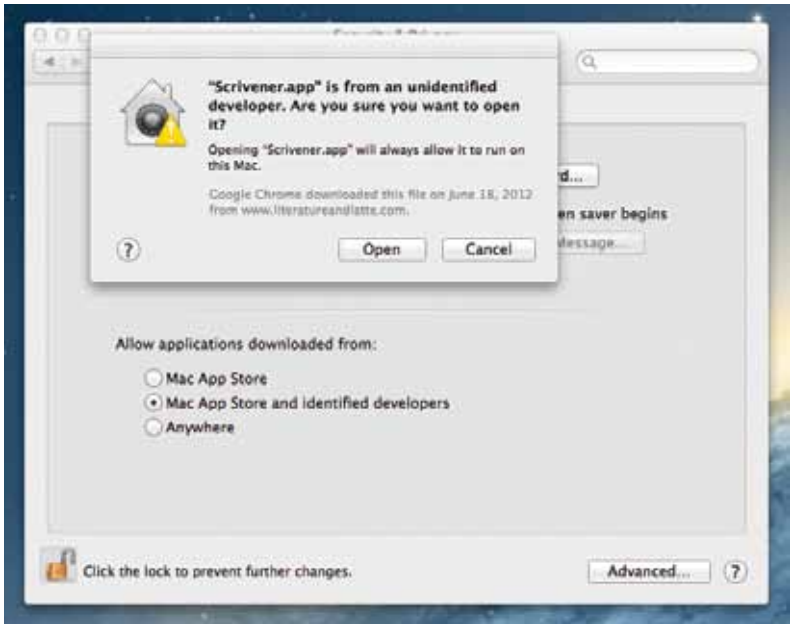
user's data. It was spread through links in emails that specifically targeted Tibetan non-governmental organizations (NGO). The next variants use a different method of delivery. These variants use an MS Word vulnerability, CVE-2009-0563 (MS Word Record Parsing Buffer Overflow Vulnerability). This vulnerability was patched way back in 2009. It affects the 2004 and 2008 versions of Word for Mac, but does not affect Word for Mac 2011. The Word doc files contained text discussing the political situation in Tibet, which led researchers to speculate that just as with the first variant, it targets Tibetan NGOs.

Another targeted malware attack is the SabPub backdoor, which was first discovered in April. The first variant did not initially show any sign that it was a targeted attack, although there were reports of emails pointing to URLs that hosted it. This malware uses the same Java exploit as Flashback, CVE-2012-0507 (Java AtomicReferenceArray Type Violation Vulnerability). The vulnerability was already patched so it didn't have as much impact as the Flashback variant that used the same exploit when it was first released. The next variant is similar to the Tibet malware in that it is delivered using the same Word doc exploit. As with the Tibet malware, the Word doc display text in Tibetan.

Conclusion

The Flashback malware outbreak has definitely put an end to the long-held belief that Macs are not susceptible to malware. We have come to a point where whatever comes next won't be as surprising anymore. In fact, at the time of this writing, a new Mac backdoor called Crisis has just been discovered, which features anti-reversing and rootkit capabilities. Crisis is the type of malware that we predicted in our last IBM X-Force Trend and Risk Report.

Apple recently released OS X Mountain Lion, which adds security features such as Gatekeeper and automatic security updates. Since June of 2012, Apple requires all applications that are submitted to the Mac App Store to have sandboxing enabled. These are great steps toward helping prevent the same mass infection we've seen with Flashback, but how well these improvements will thwart malware in the future remains to be seen.



Web content trends

The IBM Content data center constantly reviews and analyzes new web content data and analyzes 150 million new web pages and images each month. The data center has analyzed 17 billion web pages and images since 1999.

The IBM web filter database features 68 filter categories and 71 million entries with 150,000 new or updated entries added each day.

This section provides a review of the following:

- Analysis methodology
- IPv6 deployment for websites
- Anonymous proxies
- Malicious websites

Analysis methodology

IBM X-Force captures information about the distribution of content on the Internet by counting the hosts that are categorized in the IBM Security Systems web filter database. Counting hosts is an accepted method for determining content distribution and provides a realistic assessment. When using other methodologies—such as counting web pages and subpages—the results may differ.

IPv6 deployment for websites

To measure the IPv6 deployment for websites, we have performed DNS requests (to check for an AAAA record in DNS) for millions of hosts every week. As IPv4 runs out of space, we expected more and more Internet sites to switch to IPv6. However, when we looked at the numbers up to May of 2012, this expectation was not met. But in June we saw a significant increase, and the percentage of domains that have at least one host supporting IPv6 had reached 3% for the first time.

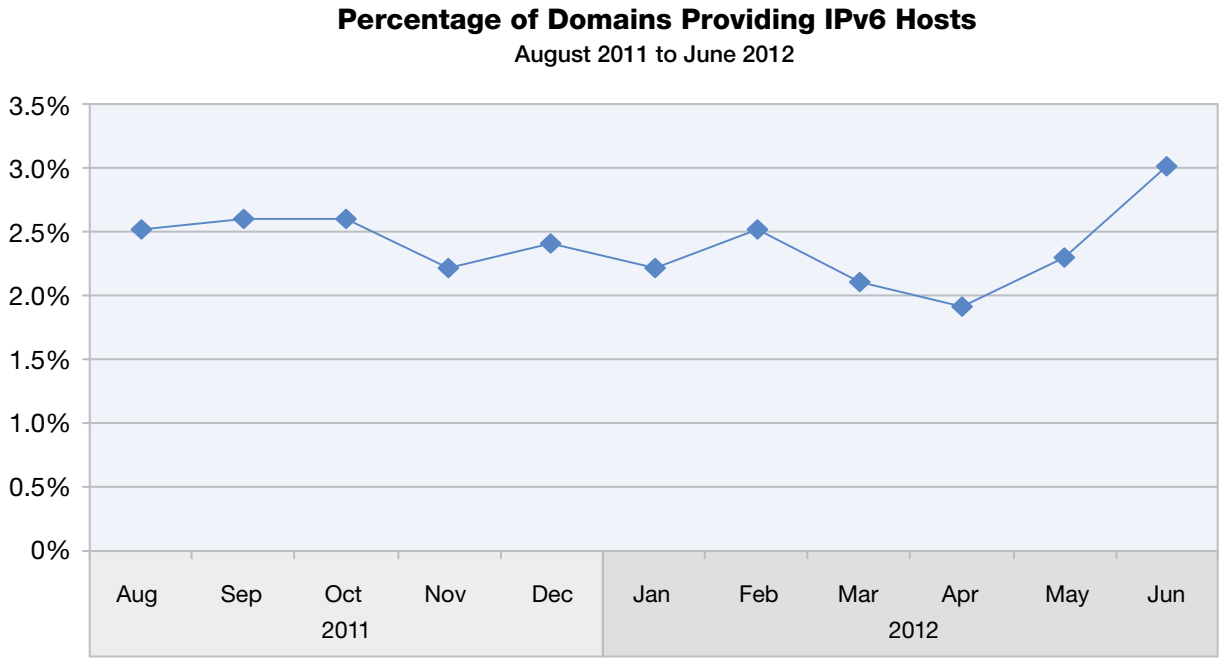


Figure 17: Percentage of Domains Providing IPv6 Hosts - August 2011 to June 2012

To analyze this increase, let's have a more detailed look at the numbers from May and June of 2012.

The change happened in week 23. In the middle of this week (June 6th) the 2012 IPv6 day[1] took place. This year many companies and organizations implemented permanent IPv6 deployments. Figure 18 demonstrates this clearly.

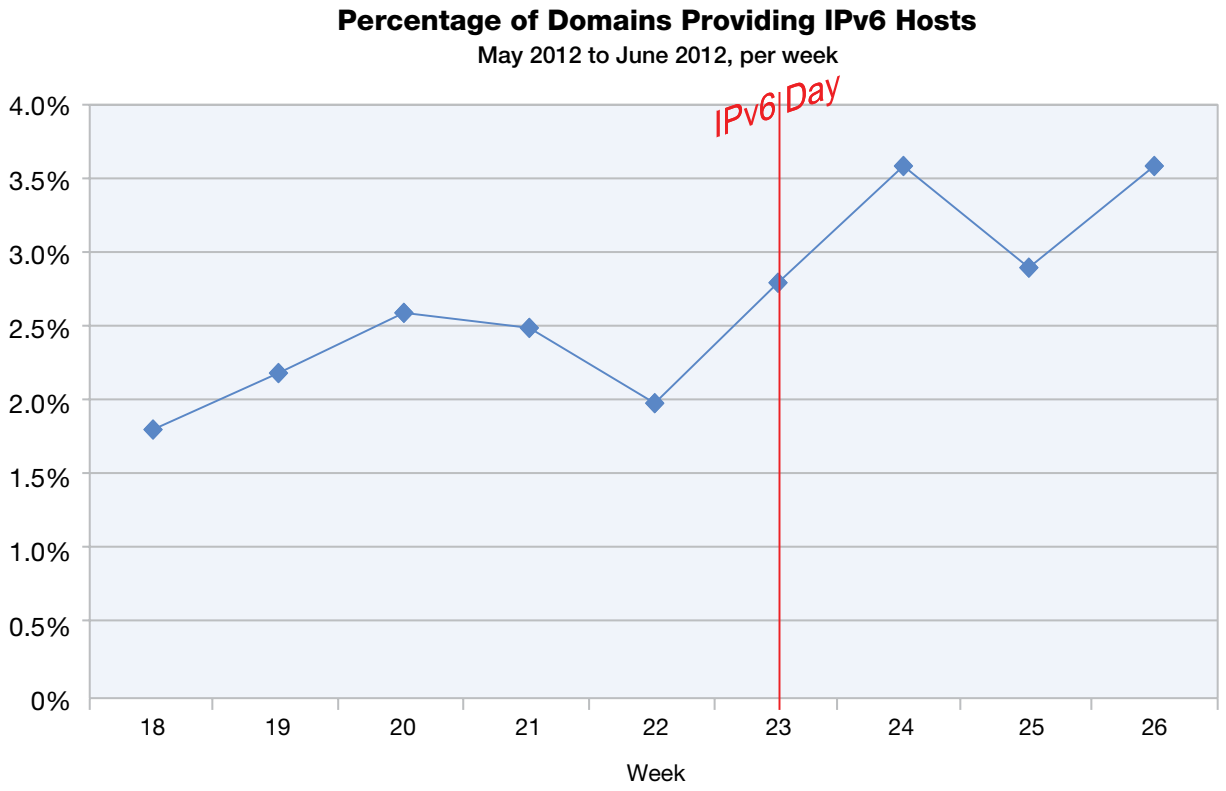


Figure 18: Percentage of Domains Providing IPv6 Hosts - May 2012 to June 2012, per week

1 See http://en.wikipedia.org/wiki/World_ipv6_day

Domains that provide at least one IPv6 supporting host can be called IPv6-ready. When we look at the types of categories[2] for IPv6-ready websites, another interesting trend appears.

- Web 2.0 sites, as well as governmental organizations, are the most IPv6-ready areas of the Internet.
- Many non-governmental organizations, search engines, portals, IT-sites, news sites, and blogs are well prepared.
- Consumer sites, such as classical web mailers, sports sites, computer games sites, shopping sites and dating sites are still above the average of 3% (the average in June, according to the chart).
- Websites with content such as illegal drugs sites, anonymous proxies, pornography, and gambling sites are particularly not IPv6-ready.
- Spam URLs represent the very bottom of the league.

Above-Average Readiness	%IPv6-ready	Lower-Average Readiness
Social Media	29.7%	
Social Networking	26.2%	
Governmental Organizations	14.5%	
Web Storage	9.3%	
Non-Governmental Organizations	9.3%	
Search Engines / Web Catalogs / Portals	9.3%	
Chat	8.6%	
Software / Hardware	8.3%	
News / Magazines	8.3%	
Blogs / Bulletin Boards	7.5%	
Webmail	6.5%	
Education	6.0%	
Sports	5.7%	
Computer Games	5.5%	
Shopping	5.5%	
Dating	4.8%	
	3.6%	Warez / Hacking / Computer Crime
Banking	3.3%	
	2.8%	Illegal Drugs
General Business	2.5%	
Travel	1.7%	
	1.4%	Illegal Activities
	1.3%	Anonymous Proxies
	1.3%	Malware
	1.1%	Pornography
	1.1%	Violence / Extreme
	0.8%	Gambling / Lottery
	0.5%	Spam URLs

Table 2: Percentage of IPv6-ready websites per category - June 2012

2 For a detailed description of the aforementioned website categories see http://filterdb.iss.net/categories/

So why are the bad guys dismissing IPv6 technology? One answer might be that many of the unwanted websites only exist for a few hours. This is particularly true for spam URLs, so these guys might want to avoid any additional technical efforts. Furthermore, the spammers want to reach as many users as possible, so there is no need to support IPv6, because everybody "speaks" IPv4 but only a few groups can "speak" IPv6.

It will be interesting to see if there is a significant increase of IPv6 support in the next few months and years.

Anonymous proxies

Increase of anonymous proxies

As the Internet becomes a more integrated part of our lives at home, at work, and at school, the organizations that are responsible for maintaining acceptable environments in these public settings increasingly find the need to control the places where people can browse.

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

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 to block anonymous proxies, then this activity (which would have normally been stopped) bypasses the filter and allows the user to reach the disallowed website.

The growth in newly registered anonymous proxy websites reflects this trend.

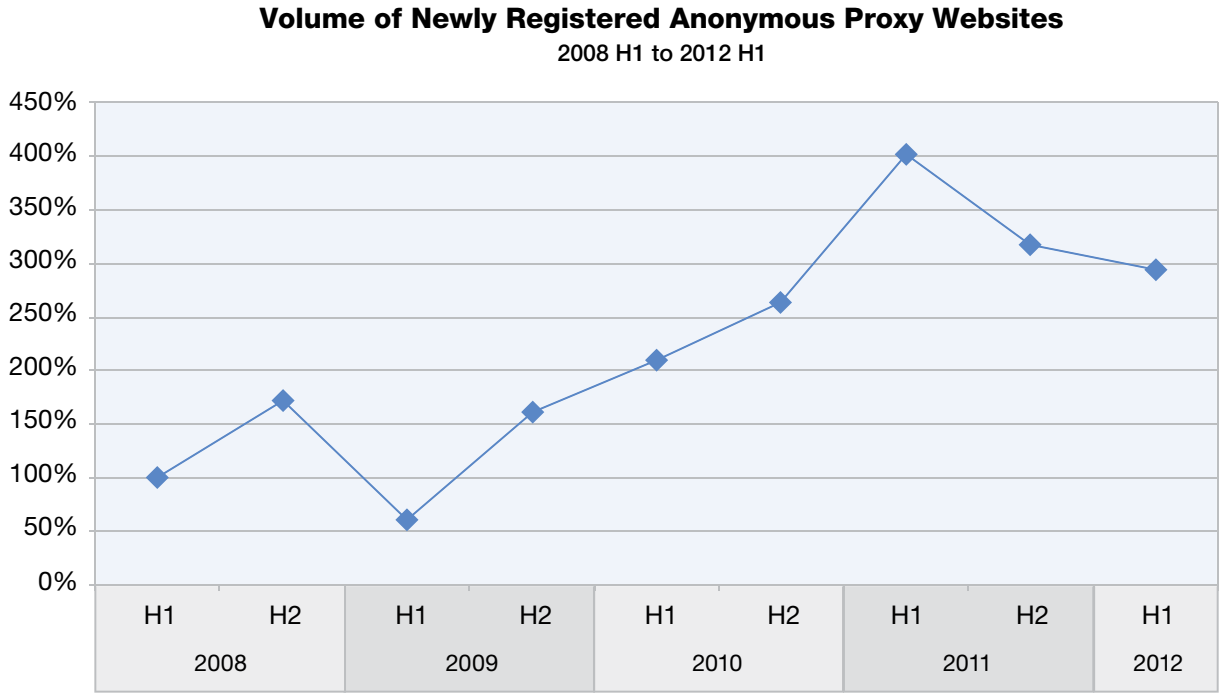


Figure 19: Volume of Newly Registered Anonymous Proxy Websites - 2008 H1 to 2012 H1

Four times as many anonymous proxies were registered in the first half of 2011, as compared to three years ago. In the second half of 2011 and in the first half of 2012, there were still about three times as many anonymous proxies newly registered as compared to three years ago. Once again we did not see another increase of this volume. Perhaps Internet activities are more focused on social networks. In many cases, these sites are not blocked at work or in schools so people no longer need to circumvent the content filtering system.

However, the use of social networking platforms issues a new challenge, particularly to companies that need to control which information is shared with other users and that need to prevent the sharing of confidential information. Thus, many companies are starting to use web application control systems, often as a part of next generation firewalls.

Anonymous proxies remain a critical type of website to track, because of the ease at which proxies allow people to hide potentially malicious intent.

Top-level domains of anonymous proxies

There are only a handful of top-level domains in use by anonymous proxies websites. Until the end of 2009 the .com and .info domains dominated, totaling more than 70% of all anonymous proxies. This has changed since the end of 2009, when the .cc domain (the top-level domain of the Cocos (Keeling) Islands, an Australian territory) entered the market, and then the .tk domain (the top-level domain of Tokelau, a territory of New Zealand) entered the market in the spring of 2010.

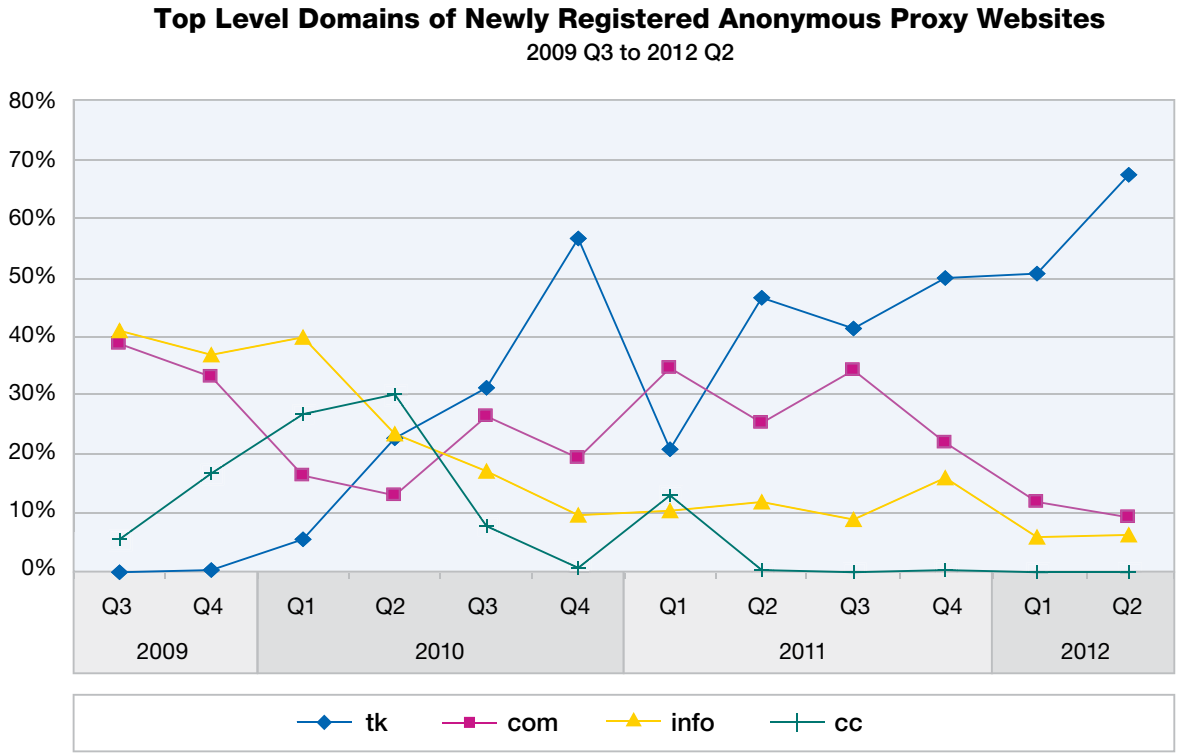


Figure 20: Top-Level Domains of Newly Registered Anonymous Proxy Websites - 2009 Q3 to 2012 Q2

As discussed in previous IBM X-Force Trend and Risk Reports, the domains of these top-level domains are free of charge.[3] Thus, today we see less than 10% of all anonymous proxies using the .info domain. The same is true for the .com domain. In the second quarter of 2012, more than two thirds of all anonymous proxies ran on the .tk domain.

Malicious websites

This section discusses the countries that are responsible for hosting malicious links and also discusses the types of websites that most often link to these malicious websites.

Geographical location of malicious web links

The United States continues to reign as the top host for malicious links. More than 43% of all malware links are hosted in the US. Germany is the new runner-up and hosts 9.2%. Russia appears in the top three for the first time. China was one of the top two until 2010, but is now at number four. France hosts 4% of malware links. Romania had two strong years in 2010 and 2011, when it reached about 8%, but it has since declined to 1.1%.

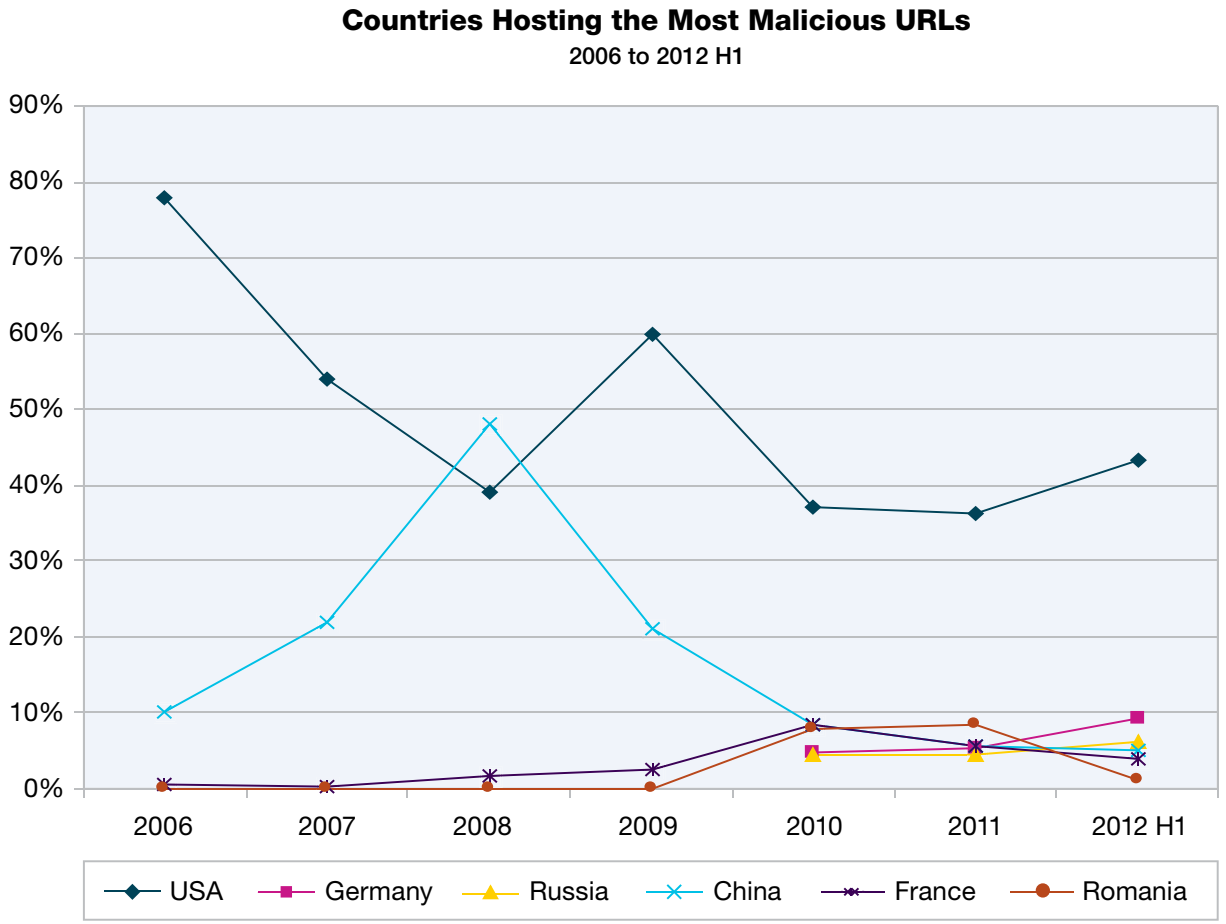


Figure 21: Countries Hosting the Most Malicious URLs - 2006 to 2012 H1

3 See http://www.co.cc/?lang=en and http://www.dot.tk/

Good websites with bad links

As mentioned at various points in our report and in past reports, attackers are focusing more and more on using the good name of trusted websites to lower the guard of end users and hide their attempts using 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 content.

Some of the top categories might not be surprising. For example, one might expect pornography and gambling to top the list. Together they now make up nearly 50% of all malicious links. However, the second-tier candidates fall into the more "trusted" category.

Blogs, bulletin boards, personal websites, and search engines fall into this second-tier category. Most of these websites allow users to upload content or to design their own websites. In other words, it is unlikely that these types of websites are intentionally hosting malicious links.

The following chart shows the history of the distribution of malware links.

When looking over the last three and a half years, interesting trends appear.

- Websites such as pornographic and gambling sites, were clearly dominant for more than a year and systematically distributed malware.
- Pornography sites were at the top, and accounted for more than one third of all malicious links.
- Gambling sites saw a decrease in malware for the first time, but still account for about 13% of all malware links. Although less than 0.6% of the adult population has a problem with gambling,[4] gambling sites are a popular target for malware distributors.
- Blogs and bulletin board malware has decreased to 7.6% in the last six months.
- Personal homepages—the classical Web 1.0 websites—continued to lose ground. One reason might be that personal homepages have fallen out of style due to Web 2.0 applications, such as profiles in social or business networks.
- Search engine, web catalog, and portal site malware decreased to 5.1%.

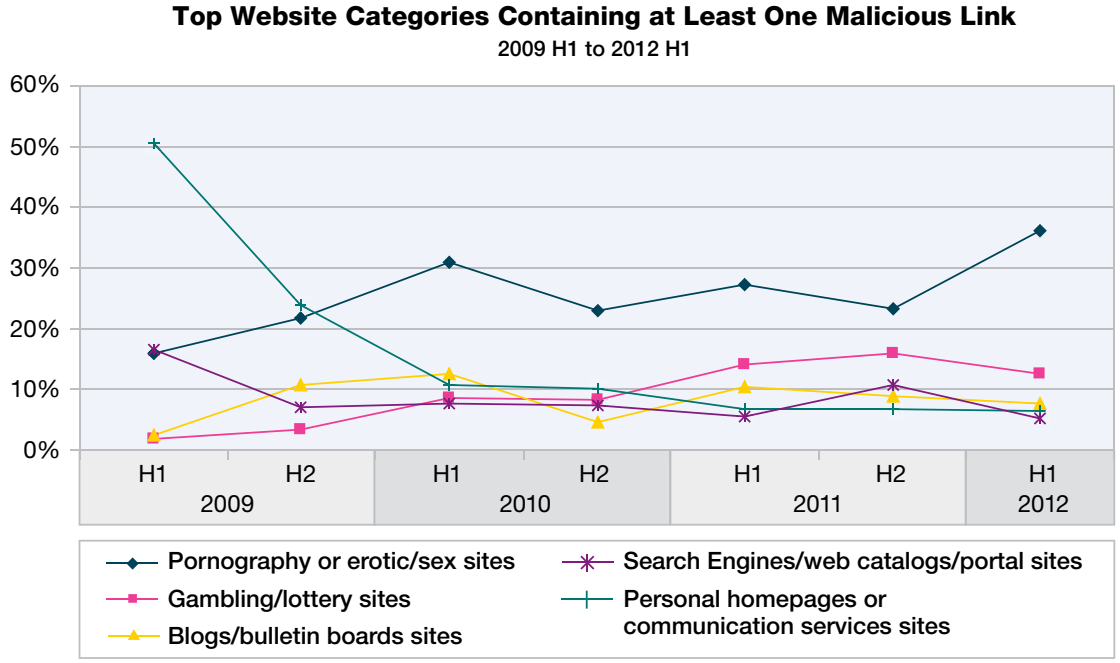


Figure 22: Top Website Categories Containing at Least One Malicious Link - 2009 H1 to 2012 H1

4 See http://en.wikipedia.org/wiki/Gambling_addiction#Prevalence

Spam and phishing

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 that 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, and 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. The updates are provided every five minutes.

This section addresses the following topics:

- Spam volume stabilized at low level
- Major spam trends during the last 12 months
- Common top-level domains in URL spam
- Spam country[5] of origin trends
- Spammers' weekend activities
- Grum botnet take down in July 2012
- Email scam and phishing

Spam volume stabilized at low level

In the spring and summer of last year we observed the same spam levels as at the beginning of 2009. After a short increase in September 2011, the volume decreased to the spring 2011 levels. In the first half of 2012 there were no major changes, and the spam volume stabilized at this low level.

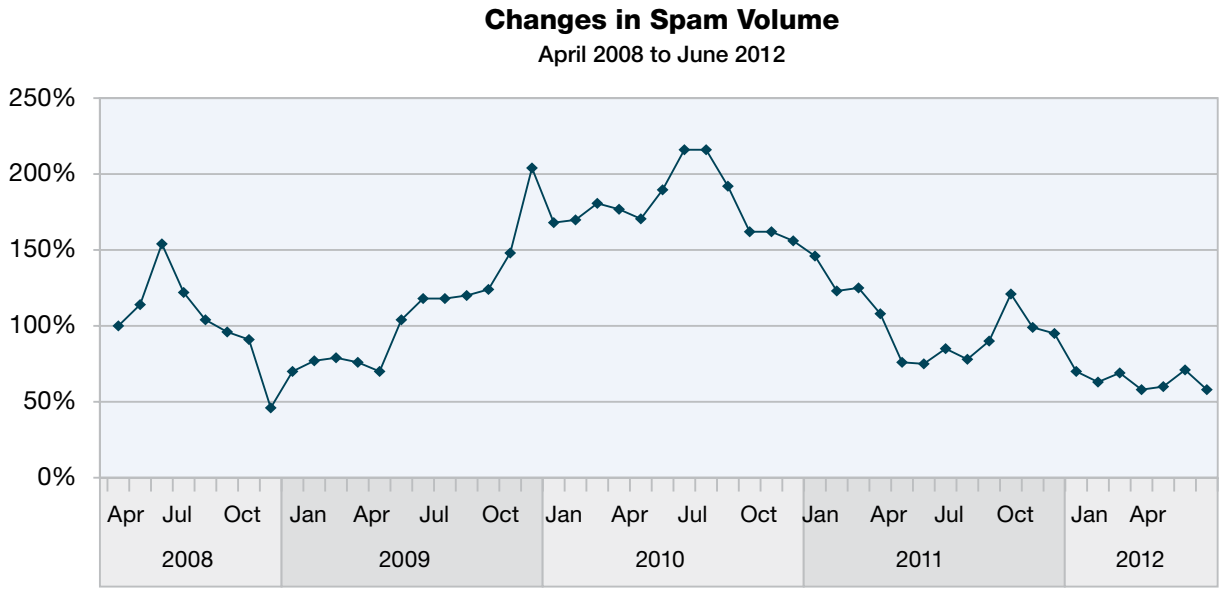


Figure 23: Changes in Spam Volume - April 2008 to June 2012

5 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 spam trends during the last 12 months

The following chart summarizes the major trends in spam we have observed since July 2011, by means of three parameters.

- **Image spam:** At the end of 2011 we saw the rebirth of image-based spam. Spammers continued to use this type of spam until end of March, 2012. Sometimes more than 8% of all spams contained an image attachment. Technically there were no changes in comparison to the image spams of December 2011.
- **ZIP/RAR spam:** In the second half of 2011 we saw several threats of ZIP/RAR spam. Both image spam and ZIP/RAR spam were discussed in detail in the IBM X-Force Trend and Risk Report. In the first quarter of 2012 there were no such threats. They have reoccurred since April of 2012, but on a much lower level. Technically there was nothing new on these ZIP or RAR attachments. Obviously they replaced the image spams of the first quarter.

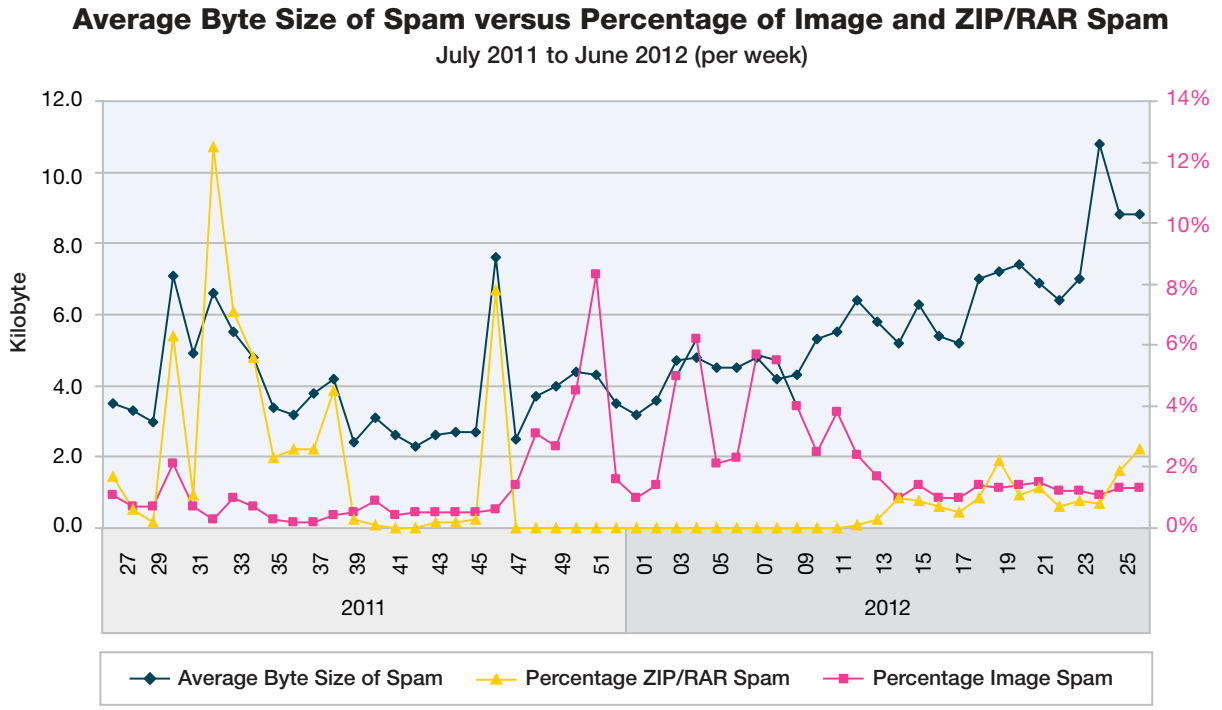
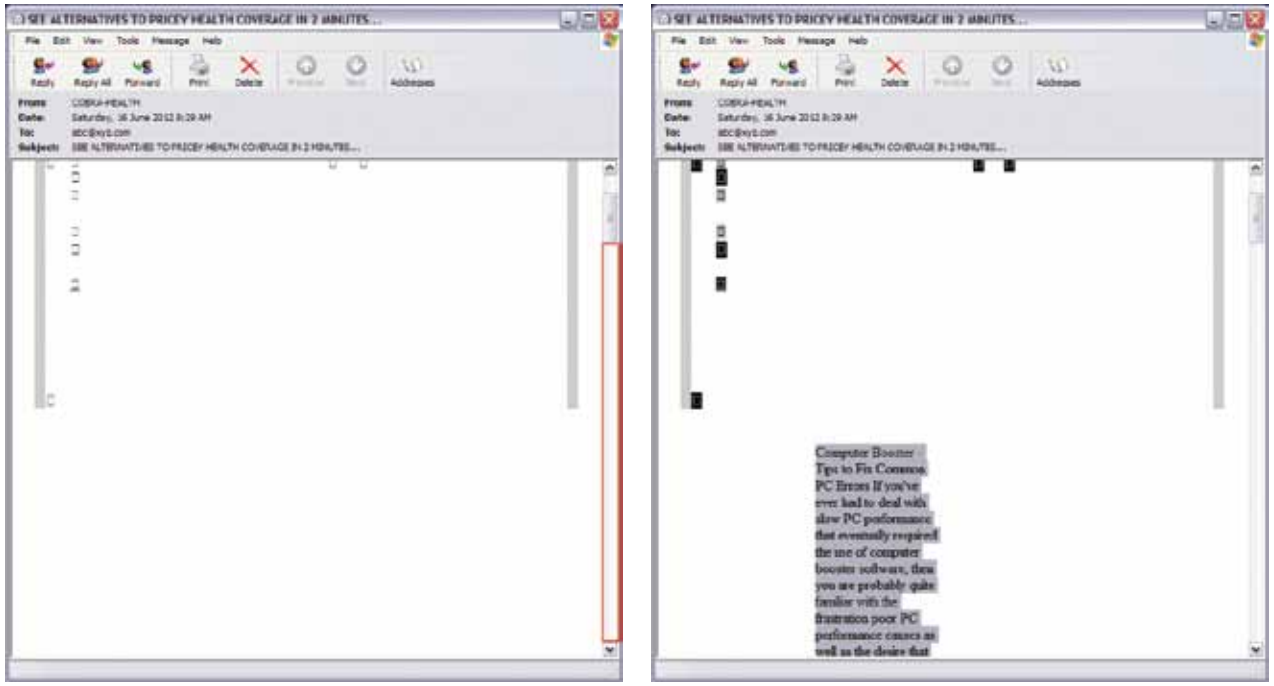


Figure 24: Average Byte Size of Spam versus Percentage of Image and ZIP/RAR Spam - July 2011 to June 2012, per week

- **Average byte size of spam:** Since mid-2010 the average size of spam has typically been about three or four kilobytes. But since the beginning of 2012 we have seen a continuous increase in the size of spam. By mid-June of 2012, the size of spams exceeded 10 kilobytes. Spammers added legitimate content from randomly chosen websites to its spams, in order to confuse and pass spam filters.

In the example on the right, the left side shows an email as a user would see it when opening it (whether images are downloaded at that point depends on the configuration). Note the huge space in the scroll bar. The same email is shown on the right side, but when the user pressed [Ctrl]+A,[6] the hidden text became visible. This text was copied from a legitimate website: http://ezinearticles.com/?Computer-Booster---Tips-to-Fix-Common-PC-Errors



Spam Sample with hidden text chosen from a legitimate website - Seen in June, 2012

6 [Ctrl]+A: for Microsoft OS, for Mac OS use [Command]+A

Content-based spam filters (e.g. Bayesian classifiers or text signature based approaches) could have problems detecting these types of spam because of the large amount of legitimate text. In a worst-case scenario, users would have to turn off their spam filter because of too many false positive matches, if these spams were added to their spam filters.

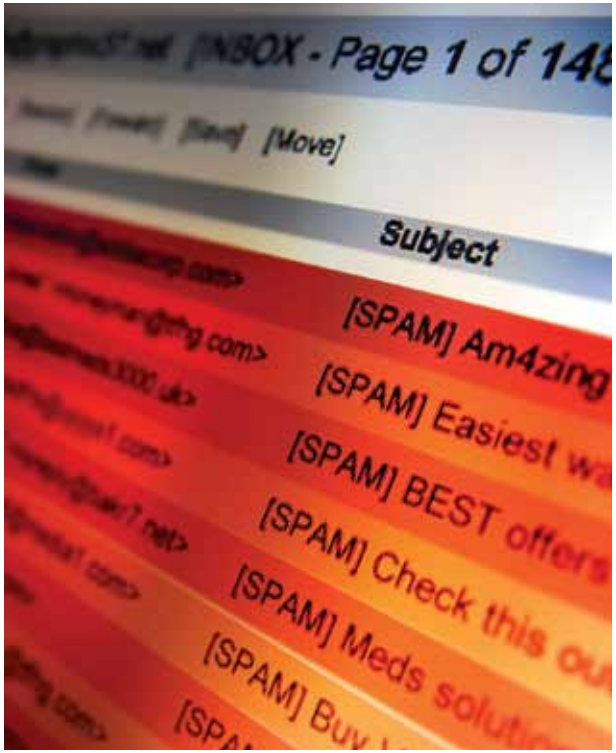
At the beginning of July, we started to see another thread with very large spam messages around 700 Kb in size. Most people would guess or assume that this massive size was due to embedded images or malware attachments. However, the truth is quite different.

These spams had an HTML part with a very large header. The header was filled with CSS commands, which were copied from multiple content management systems (such as Joomla, Wordpress, Typo3, etc.) and was completely useless for the output of that email.

It's interesting to see spammers wasting their bandwidth like this. Not very long ago they tried to keep spam small, in order to send out as much as possible. Even this example is extreme. It represents the general trend that spams are getting bigger.

What could be the reason for that?

- The recent botnet take downs hit spammers who were more focused on smaller spams. Since they have disappeared, bigger spams are more visible.
- Spammers don't care as much about bandwidth anymore, because many personal computers and mobile devices have fast Internet connections.
- It is becoming more important for spammers to not be flagged by ISPs, law enforcement groups and IT companies. Therefore, it is a reasonable approach for spammers to send fewer spams, but to make sure that those spams pass the spam filter. Some filters may have thresholds that detect a mail bigger than "size-X" which would indicate that it is not a spam.



Another view on the latest happenings are the most used spam subject lines.

In summarizing the table on the right, we get:

- January 2012: spammers used innocuous subject lines, such as "RE:" and "Fw:" as an answer or as forwarded email with an empty subject.
- February and March 2012: Employment scams[7] were the most used subject lines.
- April 2012: The month of the Romance scams.[8]
- May and June 2012: Medical products and fashion accessories were the most used subject lines.

Every one to two months we see other topics dominating the top-used spam subject lines. This demonstrates that, despite the decrease of the overall spam volume, spammers did not lose their ability to change the types of spams quickly.

Top Three Subject Lines	
January 2012	**%**
RE:	0.83%
Fw:	0.83%
Fw: Re:	0.58%
February 2012	**%**
Employment Opportunity	1.76%
Virtual Assistant Position	1.33%
Administrative Assistant Position	1.32%
March 2012	**%**
Employment Opportunity	1.04%
Vacancy - apply online	0.76%
Job ad - see details! Sent through Search engine	0.76%
April 2012	**%**
they all want you	0.37%
real beauty	0.37%
real dating	0.37%
May 2012	**%**
Replica Chanel Watches, Replica Shoes,Bags,Replica Handbags ...we specialize in Replica watches, Replica handbags, Replica shoes	0.66%
Buy Cialis Online Safely and at amazingly low prices. Bonus pills, discounts and FREE SHIPPING applied. Order Cheap Cialis Onlin	0.51%
very pretty girl	0.50%
June 2012	**%**
Buy Ciails and Viarga online!	2.01%
Re: viagra_sale	0.96%
The low prices and highest quality pills approved by FDA.Over 75.000 customers trust us. We accept Visa, Mastercard, AmEx & ACH	0.60%

Table 3: Top Three Spam Subject Lines per Month - 2012 H1

7 See http://en.wikipedia.org/wiki/Employment_scams
8 See http://en.wikipedia.org/wiki/Romance_scam

Common top-level domains in URL spam

Spammers have clear preferences over the top-level domains that they register.

- In the last two years, the two most preferred top-level domains were .com and .ru (the top-level domain of Russia).
- Well-established, second-tier top-level domains are .info and .net.
- About one year ago the newcomers .ua (Ukraine) and .pф (the internationalized top-level domain of Russia) were encountered in many spams.

In previous years there were other country code top-level domains in between the top most used domains, such as United Kingdom, the Netherlands, Chile, or Austria. Today this is very rarely the case, and in this context a shakeout seems to have taken place. This might be comparable to **the top-level domains of anonymous proxies** in which a market adjustment, concerning the number of different top-level domains used for anonymous proxies, has also taken place.

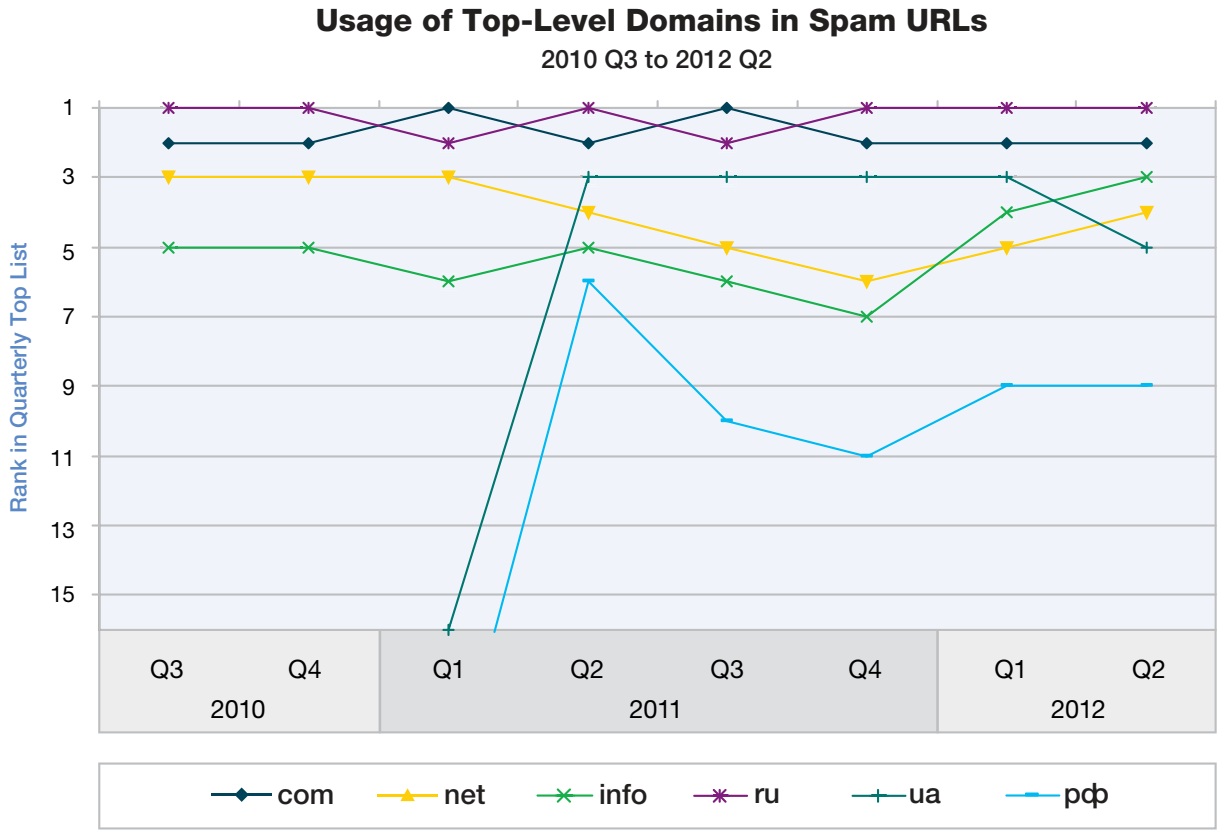


Figure 25: Usage of Top-Level Domains in Spam URLs - 2010 Q3 to 2012 Q2

Spam country of origin trends

When we look at which countries sent the most spam over the last three years, some interesting long-term trends become visible.

- India has shown nearly continuous growth (with one major decline in the first quarter of 2012) and now dominates the scene by a large margin, sending out nearly 16% of all spam. This might be the result of a 25% growth in Indian Internet users over the past 12 months.[9] It is the first time that a country accounts for about 16% of all spams. The previous record holder was the United States, which accounted for 15% in 2007.
- Vietnam varies between 4% and 10%, but seems to be established amongst the top most spam sending countries.
- The United States owned the top position in 2010, and then fell below 3% in the spring of 2011. The U.S. has recovered since the spring of 2012, and currently accounts for more than 8%.
- Brazil fell below 6% for the first time.
- Australia reached more than 6% for the first time.

There is an interesting decline of India and Vietnam in the second quarter of 2012. While both countries together totaled nearly 25% of worldwide spam in the fourth quarter of 2011 and the second quarter of 2012, at the beginning of this year they accounted for less than 14%. During this time spammers obviously found their victims in other countries such as Argentina, Italy, and Romania. These three countries had a strong first quarter in 2012, sending more than 10% of all spam.

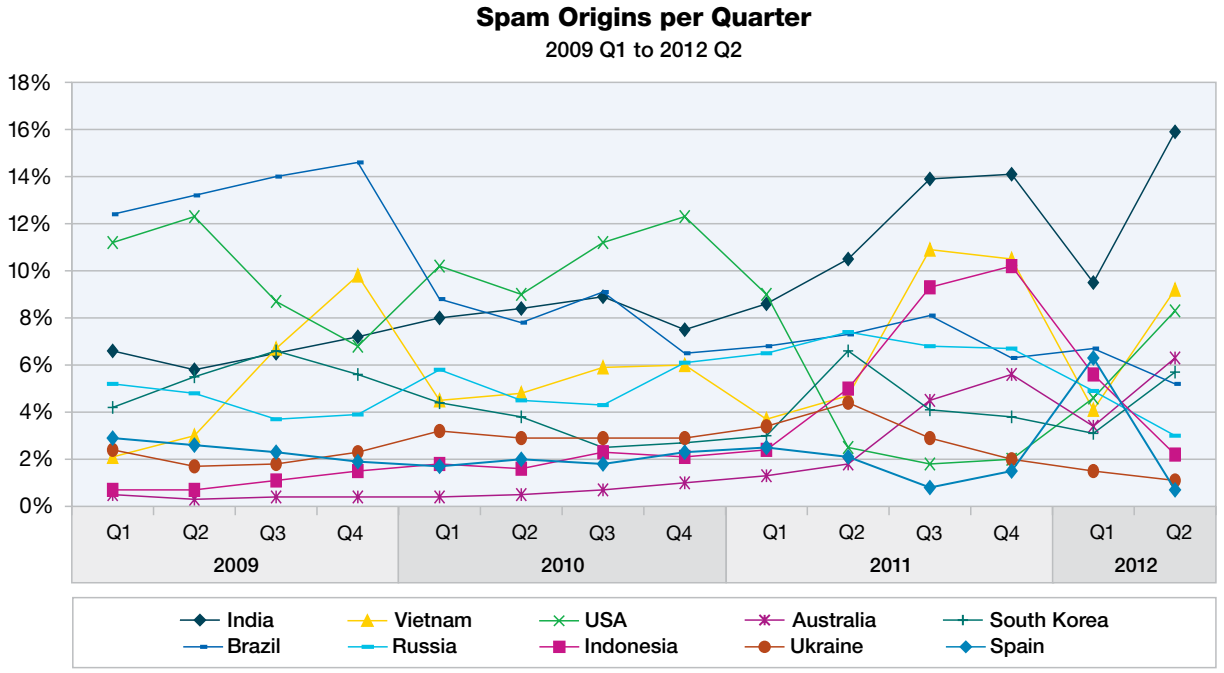


Figure 26: Spam Origins per Quarter - 2009 Q1 to 2012 Q2

9 See http://www.bbc.co.uk/news/business-16354076

Spammers' weekend activities

If spammers sent their spams equably from Monday to Sunday, they would send 14.3% of the weekly volume each day; thus, 28.6% on weekends (Saturday and Sunday). In the IBM X-Force 2010 Trend and Risk Report we saw that the volume of spam in the Russian language on weekends was significantly below the activities on weekdays, because only about 10% of the Russian spam was sent on Saturday or Sunday. In 2012 we recognize a significant change. In the first quarter of 2012, more than 14% of the Russian spams were sent on the weekend. At the same time, the volume of non-Russian spam declined to about 22% on the weekend.

The question is, why? The answers might be as follows:

- Russian spammers increasingly automate the process for sending spam (which of course was already completely automated through botnets in the last years) and they continue to automate new threats.
- Russian spammers might assume that the opportunity to bypass spam filters is better than on business days as anti-spam vendor employees enjoy their weekends too.
- At the same time, non-Russian spammers might conclude that spam threats work best on business days as many users clear their mailboxes first thing on Monday morning and quickly disregard spam sent on the weekend.
- There could be a consolidation and shakeout because spammers now use fewer methods to send spam. This is consistent with the decrease in spam volume over the last two years.

It will be interesting to see whether the weekend activities of the Russian and the non-Russian spammers continue to converge in the coming months and years.

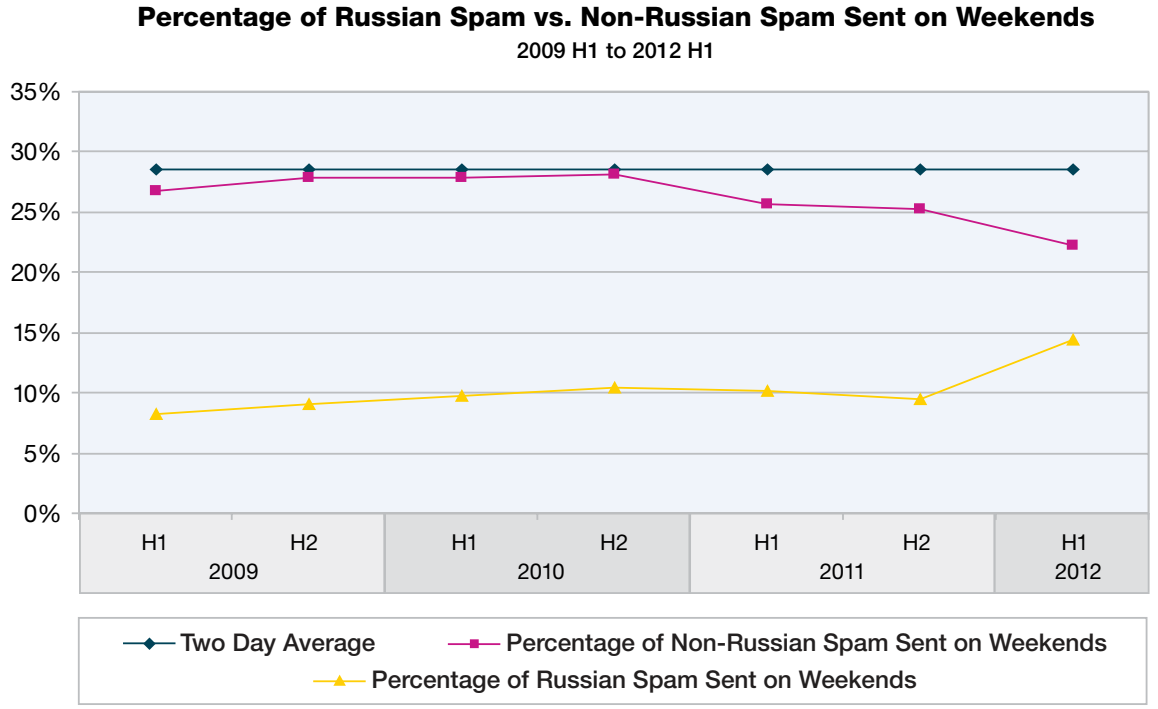


Figure 27: Percentage of Russian Spam vs. Non-Russian Spam sent on Weekends - 2009 H1 to 2012 H1

Grum botnet take down in July 2012

On July 18th, 2012, we witnessed the take down of the Grum botnet.[10] This resulted in an annual low of spam volume.

In the week of the Grum botnet take down, we saw less than 60% of the spam levels that were measured in the first quarter of 2012.

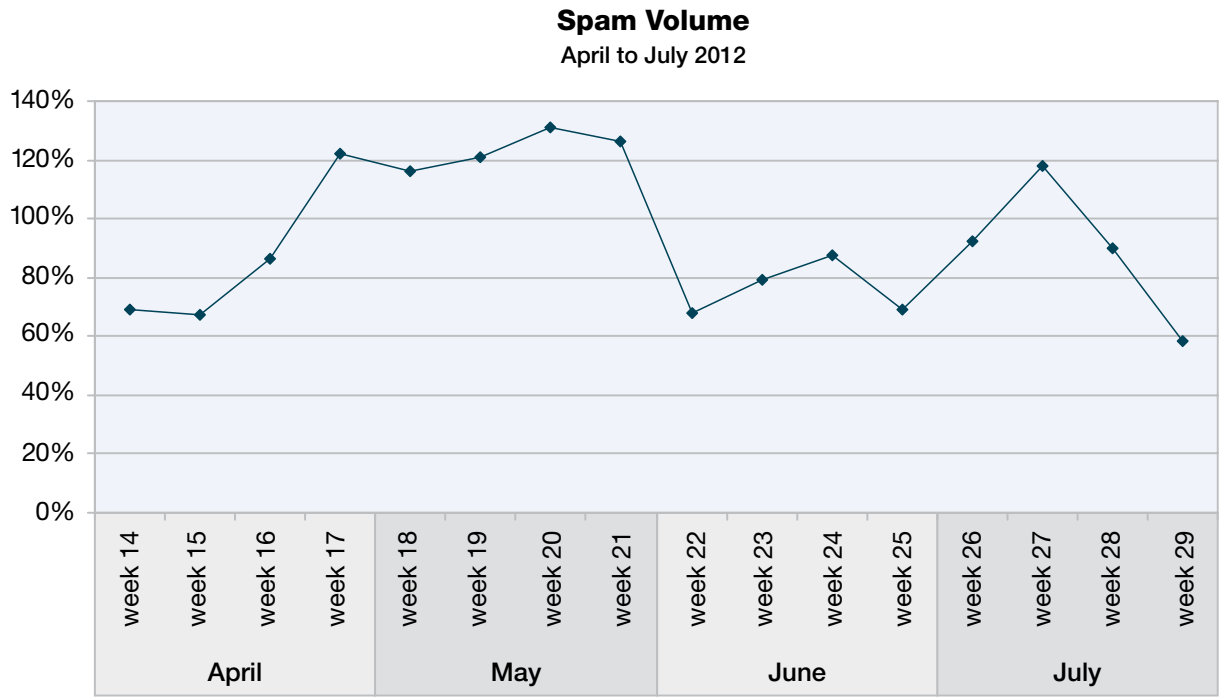


Figure 28: Spam Volume - April to July 2012

10 See http://en.wikipedia.org/wiki/Grum_botnet and http://blog.fireeye.com/research/2012/07/grum-botnet-no-longer-safe-havens.html

When we look at the spam origins before and after the take down, other interesting trends become visible.

- The Grum botnet seemed to evade infecting computers in the countries of India, Saudi Arabia, Turkey and the UK. We can assume this since before the take down of Grum botnet, these four countries gathered 36.5% of the worldwide spam volume, and after the take down they garnered 49.6%.
- Grum targeted many of its infections to computers based in the USA, Vietnam, Australia, Germany, and Brazil. Improvements in the data demonstrate that before the take down these countries were sending out 29.9% of the worldwide spam, but only 22.5% afterwards.

This is not the first time that India has been affected by a botnet deactivation. When the Rustock[11] botnet had its first shutdown during the Christmas holiday season of 2010, India increased its percentage of the worldwide spam volume from 7.1% to 11.4%.[12]

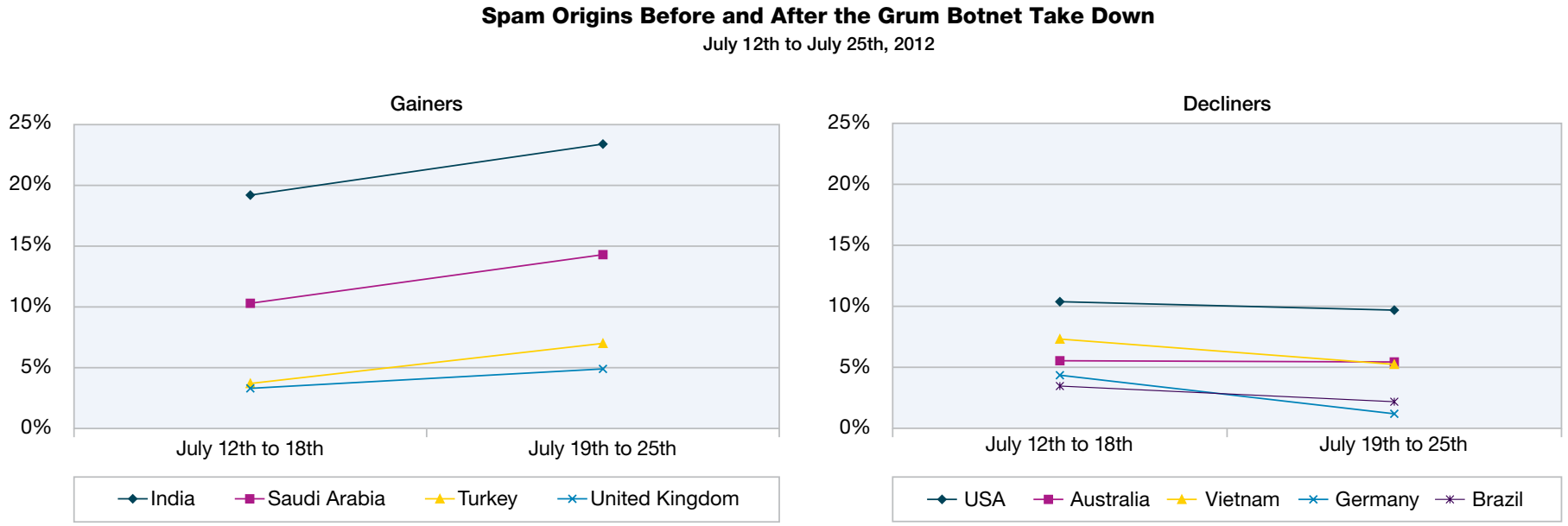


Figure 29: Spam origins before and after the Grum botnet take down - July 12th to July 25th, 2012

11 See http://en.wikipedia.org/wiki/Rustock
12 See http://blogs.iss.net/archive/2011spambotdecline.html

Email scam and phishing

Methodology

To determine the latest trends in scams and phishing:

- The statistics are based exclusively on scams and phishing deployed via email.
- The statistics include all emails that use the name of well-known brands to make users click an attachment or a link, even if the attachment or link is not phishing-related. Hence, some of the included emails are only "phishing-like" emails.
- The statistics do not include any non-email related phishing attempts, such as malware that was provided via drive by downloads and records keystrokes.

Detailed information about the methodology of the provided scam and phishing statistics is provided in the correspondent section of the annual IBM X-Force 2011 Trend and Risk Report.

Latest trends in email scams and phishing

When we take the aforementioned methodology into account, we see some significant differences between the volume of spam and the volume of email scams and phishing from the first half of 2008 to the first half of 2012 (first half of 2008 = 100% basis for both spam and scam/phishing).

- From 2008 to 2010, the spam volume nearly doubled.
- From 2008 to 2010, the email scam/phishing volume decreased significantly to less than 20% of the 2008 levels.
- From 2010 to 2012, the spam volume decreased to about one third of the 2010 levels.
- From 2010 to 2012, the email scam/phishing volume nearly quadrupled, reaching more than 83% of the 2008 levels in spring, 2012.

To conclude, the volume of spam and the volume of scam and phishing behave contrarily.

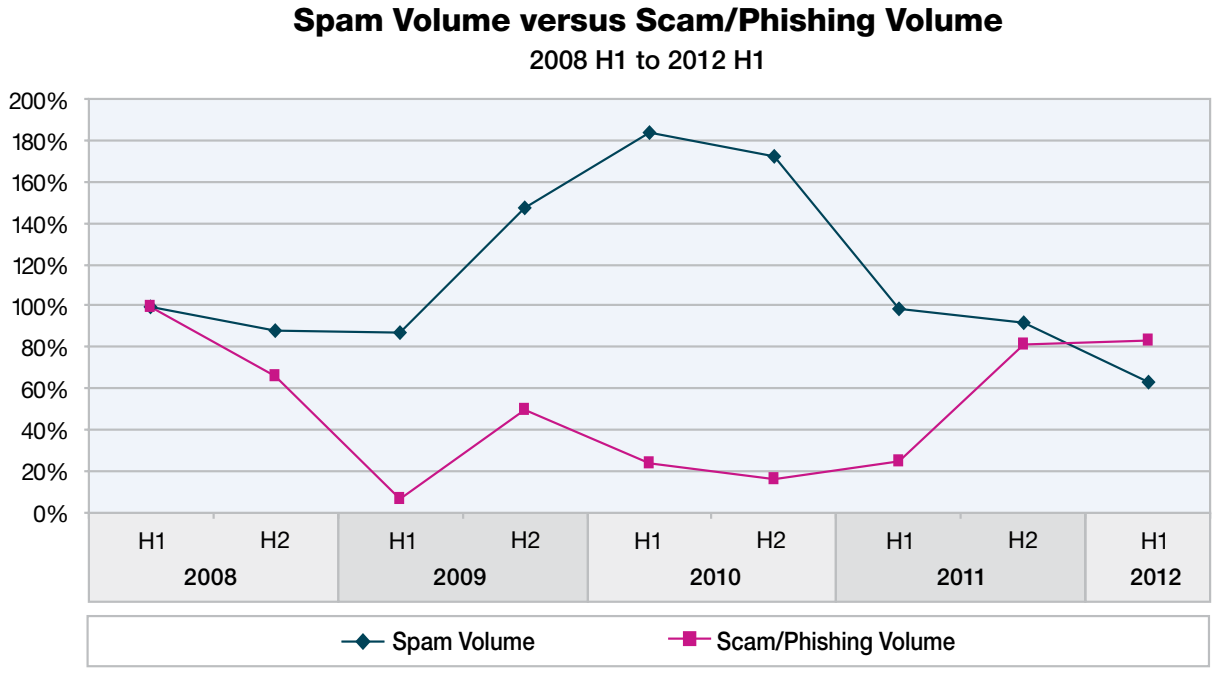


Figure 30: Spam Volume versus Scam/Phishing Volume - 2008 H1 to 2012 H1

When we look at the types of email scam and phishing, some interesting trends become visible.

- Until 2009, traditional email phishing that targeted financial institutions dominated the statistics, and represented more than 50% of all phishing emails. They have not dominated the statistics since the beginning of 2010.
- Since the beginning of 2010—when we started to monitor this class of emails—social networks have dominated the statistics by staying in the top two. At the beginning of 2011, more than 80% of legitimate brand names were used in emails on social networks, stabilizing at 43% during the second half of 2011. After a short break at the beginning of 2012, they now account for more than 31% of all scams and phishing.
- Parcel services were widely used to dupe users during the second half of 2010 when they reached about 20% of all scam/phishing-like emails. In the second quarter of 2011, more than 50% of this type of spam used the good name of parcel services. This type nearly disappeared by the end of 2011 and beginning of 2012, but came back in the second quarter of 2012 reaching more than 27% of the scam/phishing volume.
- At the beginning of 2012, phishers focused on nonprofit organizations, accounting for 66% of all scams and phishing in the first quarter, but then declined to 7% in the second quarter of 2012.
- Scanner scams (such as "Scan from your printer #6269319") that include a malicious attachment made it into the top three in the second quarter for the first time, and accounted for more than 13% of all scams and phishing.

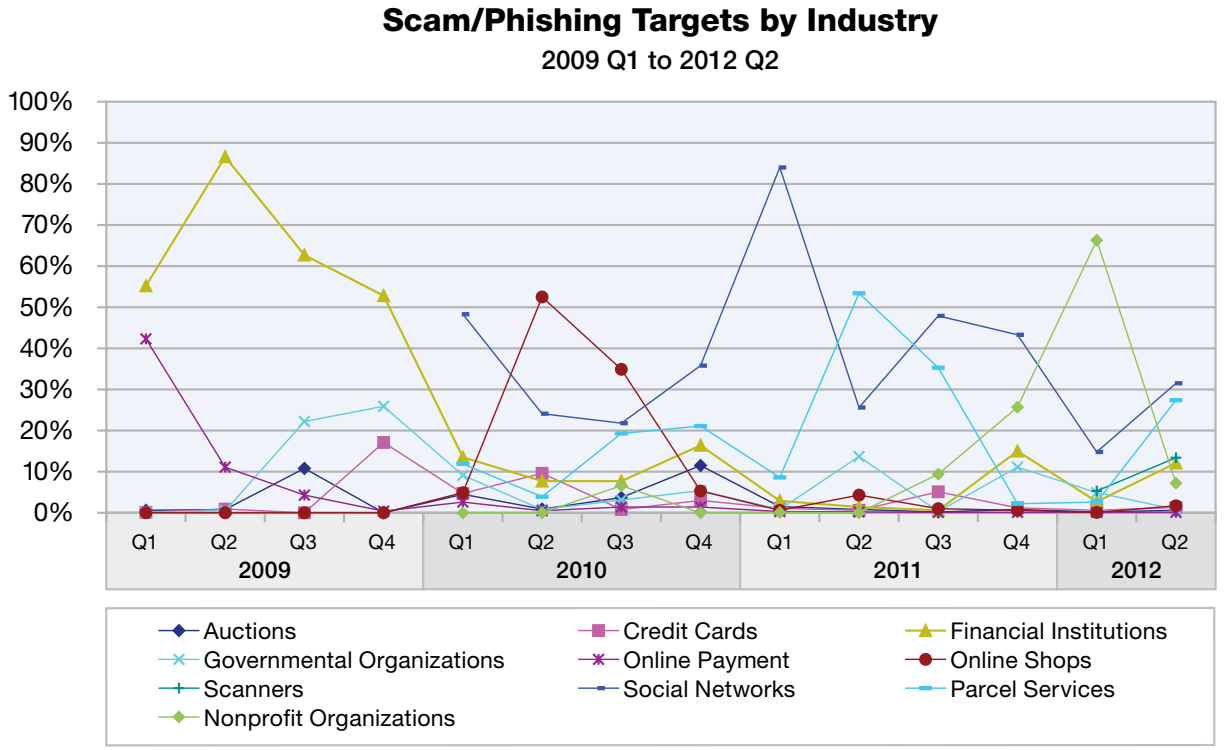


Figure 31: Scam/Phishing Targets by Industry - 2009 Q1 to 2012 Q2[13]

13 The numbers concerning social networks, parcel services, and nonprofit organizations were not recorded before the beginning of 2010.

Reviewing the ups and downs of the previous chart, we can see that phishers are repeating the following targets to get users to click links or attachments. The method is the same, but the target is different.

With each new spam iteration they find new victims (i.e., new Internet users) who fall for their tricks.

It is also interesting to see from which countries the phishing-like emails are being sent.

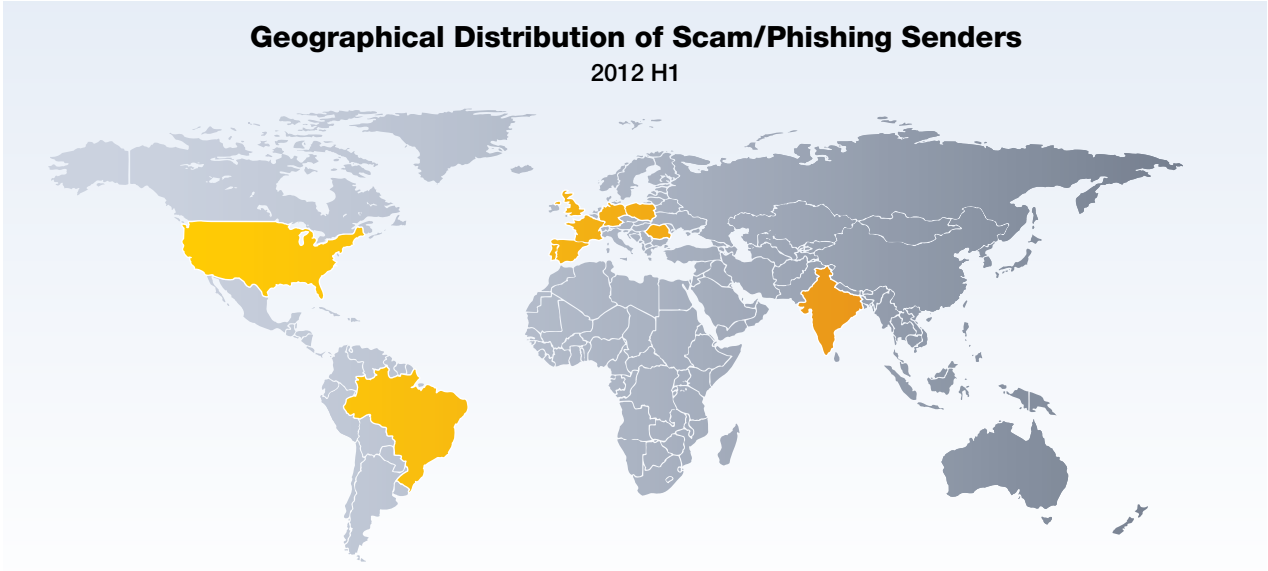


Figure 32: Geographical Distribution of Scam/Phishing Senders - 2012 H1

Country	% of phishing
Spain	7.6%
Romania	7.4%
United Kingdom	6.4%
Germany	5.5%
Brazil	5.0%

Country	% of phishing
India	4.9%
Poland	4.8%
France	4.4%
USA	3.8%
Portugal	2.5%

Table 4: Top 10 Countries of Scam/Phishing Origins - 2012 H1

Social networks have been the dominant targets of email phishing for more than two years, so, in conclusion, let's take a look at the countries this type of email phishing is sent from.

Messages sent from the U.S. account for nearly 15% of all social network scams/phishings. The runner-up is France, which accounts for about 8% of all social network phishings This country distribution is significantly different from the overall scam/phishing country distribution, which seems to indicate that this type of phishing does not come from the same botnets as other types of spam and phishing.

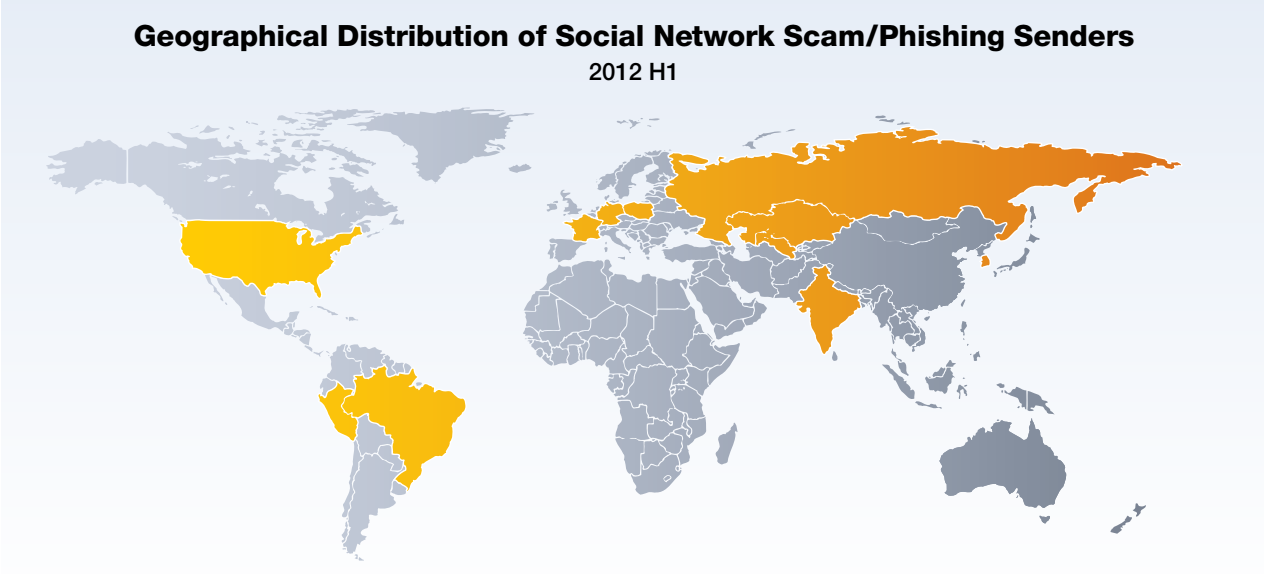


Figure 33: Geographical Distribution of Social Network Scam/Phishing Senders - 2012 H1

Country	% of phishing	Country	% of phishing
USA	14.7%	Russia	4.0%
France	7.9%	Poland	3.5%
Brazil	6.0%	India	3.3%
Germany	5.3%	Peru	3.2%
South Korea	4.5%	Kazakhstan	3.0%

Table 5: Top 10 Countries of Social Network Scam/Phishing Origins - 2012 H1

Section II
Operational security practices

In this section of the Trend Report we explore those topics surrounding weaknesses in process, software, and infrastructure targeted by today's threats. We discuss security compliance best practices, operating-cost reduction ideas, intelligence and 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.

Combating advanced persistent threats (APTs) with security intelligence and anomaly detection

Advanced persistent threats (APTs) have become one of the industry's most discussed topics. Certainly not every security breach is the result of an APT. In fact, the vast majority are not. But it's clear that some breaches *have* resulted from the efforts of well-organized teams pursuing specific objectives via patient, long-running attacks, often using malware and tactics highly customized to the targeted organization and specific employees. In other words, APTs.

The business impact of advanced persistent threats is startlingly large. As we showed in the IBM X-Force 2011 Annual Trend and Risk Report, firms around the world have suffered significant breaches, many due to APTs, leading us to call 2011 the "Year of the Security Breach." In early 2011 and prior, major firms such as RSA, Google and others experienced widespread compromises that exposed customer and user data as well as sensitive intellectual property.

In a recent Enterprise Strategy Group survey of security professionals in US-based enterprise-class organizations, 59% of respondents stated they believe it's "highly likely" or "likely" that their organizations have been APT targets. Moreover, 30% believe their organizations are "very vulnerable" or "vulnerable" to a future APT attack. Even among those organizations considered "most prepared for APTs," 46% of respondents believe they are "very vulnerable" or "vulnerable" to a future APT attack.[14]

The question is how can organizations defend against doggedly determined, patient and creative adversaries who know a great deal about your employees and are often well funded? You can't rely purely on prevention approaches, as advanced attackers may eventually breach your defenses. Nor can you rely solely on signature-based detection technologies, because such attacks can also elude them. While prevention and signature-based detection are both necessary for enterprise security, more protection is required and new strategies must be adopted. Security Intelligence approaches that incorporate anomaly detection have emerged to complement traditional solutions and help defend against advanced persistent threats.

Understanding advanced persistent threats

While the term has no single consensus definition, common views of APTs include the notions of a directed and targeted effort, a persistent and potentially long-running attack, and "advanced" techniques (in terms of technical and/or operational sophistication). Although many executives and information security leaders acknowledge the existence of APTs, quite a few doubt their own organizations would be targeted. *If I'm not in a government agency or Fortune 500 corporation, they reason, would someone actually invest the effort to attack my organization in this way?*

14 "U.S. Advanced Persistent Threat Analysis," Enterprise Strategy Group, October 2011

The answer, unfortunately, appears to be "yes." In a recent paper on one very pervasive APT, it was reported that the attackers had compromised 72 different parties, including construction and heavy industry firms, real estate firms, and national Olympic committees—not what one would think of as the most likely APT targets.[15] **Therefore, many organizations are simply assuming the worst—that they are already under reconnaissance or attack.**

In a true APT scenario, you should also assume that the attacker can penetrate your defenses eventually. That is due to:

1. The inherent difficulty of keeping all points of entry secure. This includes patching and protecting public-facing resources every time a vulnerability is discovered, ensuring secure configurations, and so on.
2. The challenge of protecting against social-engineering based attacks that can lead to account compromise or can neuter prevention capabilities.

Or, as one analyst recently observed:

"Most large enterprise security administrators and CISOs understand that **it is not a matter of if, but when their organization will experience a breach—**one that could potentially be very painful for the whole organization. … One thing is clear: the longer a stealthy attacker sits undetected in the enterprise network and its endpoints, the more damage they can do." (Emphasis added by IBM.)[16]

To understand the tactics employed and vulnerabilities exploited in advanced attacks, let's look at some examples. Not only are some of these actions creative and technically advanced, but the way they are orchestrated makes their effectiveness greater than the sum of their parts. A combination of tactics (such as those listed below) carefully choreographed and sometimes reflecting months of research and customization, makes APTs difficult to combat. Examples taken from a variety of attacks include:

- **Infiltrating a trusted partner.** In one case, attackers compromised their target's trusted third-party software provider, inserted trojan code into the software update server, and waited for the software provider to auto-update the trojan onto the target's network.
- **Creating custom malware.** Typical of an APT, the trojan in the previous example was tailored to only infect the target organization and none of the software provider's other customers, thus preventing the malware from spreading widely and being identified by antivirus vendors before it had accomplished its mission.
- **Using research and social engineering to compromise user accounts.** Patient and committed attackers perform extensive reconnaissance of spear-phishing targets and then contact them with highly believable communications (email, IM, or social networking message) which may reflect knowledge of the individuals' work activities, colleagues, friends, and family. The phishing message can include a link or attachment leading to infection of the target's system, often with custom malware.

15 "Revealed: Operation Shady Rat," McAfee, 2011
16 Blog post: "Okay, Breaches Are Inevitable: So Now What Do We Do?" by Paula Musich, Current Analysis, July 20, 2012, http://itcblogs.currentanalysis.com/2012/07/20/okay-breaches-are-inevitable-so-now-what-do-we-do/

- **Exploiting zero-day vulnerabilities.** An alternate approach to social engineering for the purpose of breaching the target's perimeter—as well as a tactic used to extend the compromise's reach—is the use of zero-day exploits to gain access to user and administrator accounts. With thriving ordinary and black markets in zero-day exploits and advanced deployment technologies, attackers don't even have to discover zero-day vulnerabilities or craft the exploit themselves. In sum, it is not just pragmatic but also economical to utilize these markets.
- **Communicating over covert channels.** Adversaries often use malware to co-opt a machine into joining a botnet, and later communicate with the botnet command and control server over a covert channel, such as port 80 or 8080. This approach might also be used to exfiltrate data from the targeted organization.

Because legitimate APTs are likely to breach their target's perimeter sooner or later, effective detection and forensic capabilities are essential. While protection and prevention efforts should not be neglected, the true measure of an organization's APT defenses is its ability to quickly detect breaches and thoroughly research the extent and impact of those breaches.

Security intelligence: Uniquely equipped to defend against APTs

As we introduced in the IBM X-Force 2011 Annual Trend and Risk Report, Security Intelligence is a new class of solutions that provide unified visibility and real-time analytics across the spectrum of security operations.

> Security Intelligence (SI) is the real-time collection, normalization, and analysis of the data generated by users, applications, and infrastructure that impacts the IT security and risk posture of an enterprise.
>
> Data collected and warehoused by Security Intelligence solutions includes logs, events, network flows, user identities and activity, asset profiles and locations, vulnerabilities, asset configurations, and external threat data.

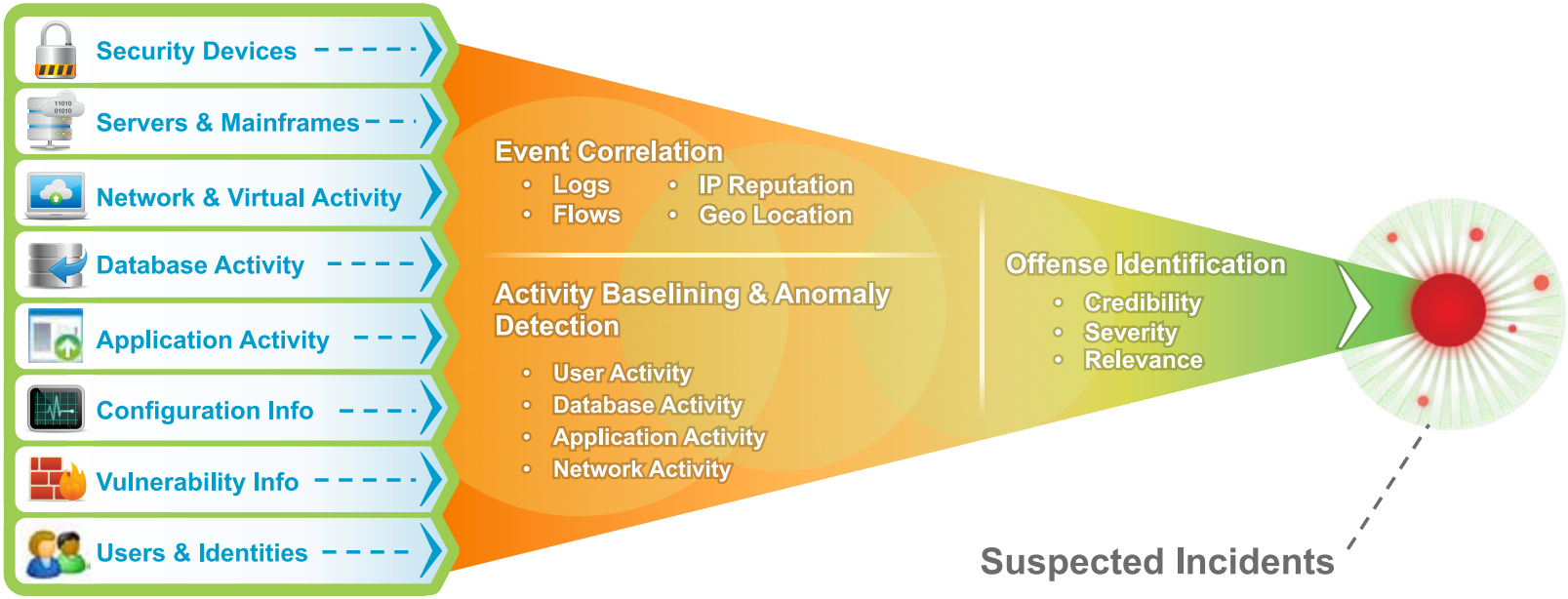
Several elements make SI an ideal approach to help combat advanced persistent threats:

- **Consolidation of data silos for 360-degree view.** Because Security Intelligence analyzes a diverse set of data, it can connect the dots between seemingly unconnected or benign activity and ultimately deliver better insight for APT detection.
- **Pre- and post-exploit insights.** Organizations use SI to gather and prioritize information about existing security gaps that should be addressed (helping prevent breaches), as well as suspicious behavior already taking place within the network (helping detect breaches).
- **Anomaly detection capabilities.** Baselining current activity, identifying deviations from normal behavior, and then determining which deviations are meaningful is a core aspect of Security Intelligence. This can be vital to detecting APTs in progress.
- **Real-time correlation and analysis.** SI solutions can correlate massive sets of data in real time, using advanced analytical methods and purpose-built databases. This allows for earlier and more accurate detection of APTs, helping to distinguish the signal from the noise.
- **Helping reduce false positives.** Through combining all of these analytical approaches, SI can not only help detect compromises faster, but also de-prioritize unusual yet benign activity. Reducing the time spent investigating anomalous but harmless activity can make a huge difference in the

organization's ability to focus on its top objectives.

- **Forensic capabilities.** After a breach has been discovered, the next crucial step is to exhaustively research the impact of the breach. Security Intelligence can provide a single-console view of log data, network traffic, and other security telemetry across thousands of systems and resources, easing the burden on the security and network staff who have to rapidly assess the breach.
- **Flexibility.** Because the internal IT environment and the external threat landscape can change quickly, APT defense approaches should support frequent change. Modern SI solutions typically make it easy to add data sources, create and tune analytics, create new user views and reports, and expand and evolve the overall deployment architecture.
- **Unified approach.** APTs are usually complex, multi-pronged attacks involving dozens if not hundreds of target systems. Because some Security Intelligence solutions are delivered via a unified and modular platform, they can help organizations intelligently wade through masses of data and perform a broader set of analytics and ad hoc querying than other approaches.

Security Intelligence correlates and analyzes a diverse set of security-relevant data



Anomaly detection: The security intelligence lynchpin of APT defense efforts

Perhaps the greatest weapon provided by Security Intelligence in combating APTs is anomaly detection. Since advanced adversaries use creative and targeted attack strategies, often in combination with zero-day exploits, traditional signature-based defenses are often insufficient. What organizations require is the ability to detect activity that's just a little unusual and then enrich it with as much context as possible to distinguish the benign anomalies from the real threats.

APT attacks don't come with bells or blinking lights; they blend into your environment as much as possible. It takes rigorous, automated, and continuous monitoring—and maximum use of data—to have a chance at finding them before major damage is done.

The anomaly detection technologies found in today's Security Intelligence solutions have their roots in the network behavior anomaly detection (NBAD) space. However, they have expanded their capabilities beyond traditional NBAD to support not only network flow (network traffic) analysis, but also log data analysis. With the unified approach of Security Intelligence, security teams can perform real-time analytics on a combination of network flow and log data simultaneously, to gain better insight into potential threats and enhance situational awareness.

Anomaly detection works by monitoring for activity that falls outside of "normal" behavior. It determines baseline levels of activity along dimensions of interest and then triggers alerts as appropriate. Ideally the learning time period and the trigger time period can both be adjusted easily, and seasonality and growth trends can be accounted for.

Examples of the wide array of anomalies that can be detected with Security Intelligence include:

- Outbound traffic is sent to a country in which the company does not do business and to which no traffic should be sent.
- A known application (such as IRC chat) is using a non-standard port (such as port 80).
- FTP traffic is observed in the Finance department when Finance has never had FTP traffic before.
- A self-propagating worm outbreak occurs.
- A new service is initiated on a known host, potentially signaling a breach.
- A host system changes roles—for example, an external-facing DNS server is changed to also be the SMTP relay.
- Network traffic volume changes; the volume of traffic to a particular host is 200% higher during the last 24 hours than its historical average level over the past 3 months, with no clear seasonal explanation for the increase.

In sum, anomaly detection can provide an intelligent foundation for discovering APT breaches. It doesn't require advance knowledge of what the attack might look like, but can automatically monitor network-wide activity for notable deviations.

Best practices for anomaly detection

When deploying anomaly detection capabilities to protect against APTs, we recommend these best practices:

- **Monitor user activity, especially for privileged users.** One of the primary tactics used in most advanced attacks is takeover of employee accounts, especially employees with privileged access. After an account is compromised, the adversary might attempt to access applications or systems not previously used by that employee or to access resources during unusual hours. The more intelligence your solution can develop around employees' normal activities, the more effective it can be at spotting meaningfully abnormal behavior.
- **Monitor access to sensitive data.** Similarly, focus on protecting the data that would be of greatest value to an attacker—customer data, financial data, intellectual property, and so on. Develop intelligence around the typical rhythms of activity involving sensitive databases and other data stores, so you can detect irregularities that might be meaningful. Database security solutions can also provide valuable security telemetry for anomaly detection. Ideally, pair data access monitoring with user activity monitoring for even more accurate threat detection.
- **Monitor outbound traffic to prevent data exfiltration.** Enhance your monitoring of outbound traffic so that you can detect and stop exfiltration of sensitive data. Would you know if traffic were initiated to an unusual country you don't do business with, or sent through an unusual port? Could you detect traffic being sent through a covert channel? Would you know if an internal host initiated communication to a dynamic-IP ranged address?
- **Monitor geographic access and traffic.** Even if you operate in a global environment and do business in and with many countries around the world, there is probably a finite set of countries you expect to see network traffic to and from. When traffic takes place with other geographies, it might be worthy of investigation, particularly if other suspicious behavior is observed with users or systems related to that activity.
- **Leverage threat intelligence with anomaly detection.** Many commercial and community threat intelligence services, including those provided by IBM X-Force, provide rich insight into threat activity and bad actors, which can further enrich anomaly detection. For example, you should know if users or systems are interacting with sites known to host malware, botnet command and control servers, or other threats.
- **Collect network flows for greater insight.** Network flow data—especially layer seven data with content visibility—can be a highly useful data source for anomaly detection. It can also provide invaluable information for confirming or disproving the existence of a breach, and determining the extent and impact of any breaches.

Conclusion

With recognition that breaches are virtually inevitable, the focus in many organizations has turned to detection. Security Intelligence has emerged as a leading candidate to combat APTs, leveraging the ability to collect, normalize, and analyze massive and varied sets of data. Anomaly detection lies at the center of Security Intelligence, enabling information security teams to identify meaningful deviations from the normal rhythms of activity. Through the use of Security Intelligence solutions and best practices, organizations can achieve a more proactive security stance.

Vulnerability disclosures in the first half of 2012

Since 1997, IBM X-Force has been tracking public disclosures of vulnerabilities in software products. IBM X-Force collects software advisories from vendors, reads security-related mailing lists, and analyzes hundreds of vulnerability web pages where remedy data, exploits, and vulnerabilities are disclosed.

In the first half of 2012, we reported just over 4,400 new security vulnerabilities. If this trend continues throughout the rest of the year, the total projected vulnerabilities would be slightly more than the record we saw in 2010 approaching 9,000 total vulnerabilities.

Since 2006, and our first decline in vulnerability disclosures in 2007, we have seen the total number of vulnerabilities go up and down every other year. There is not a defining reason behind the fluctuation year over year, but 2012 could very well be a record setting year for security vulnerability disclosures.

Web applications

The continuing trend of the total number of security vulnerability disclosures can also be found within the category of web application vulnerabilities. In 2011, we saw a decrease in web application vulnerabilities from 49% to 41%. However, in the first half of 2012, we saw a resurgence of web application vulnerabilities. The projected percentage of web application vulnerabilities for 2012 now stands at 47%, with over 2,000 reported so far this year.

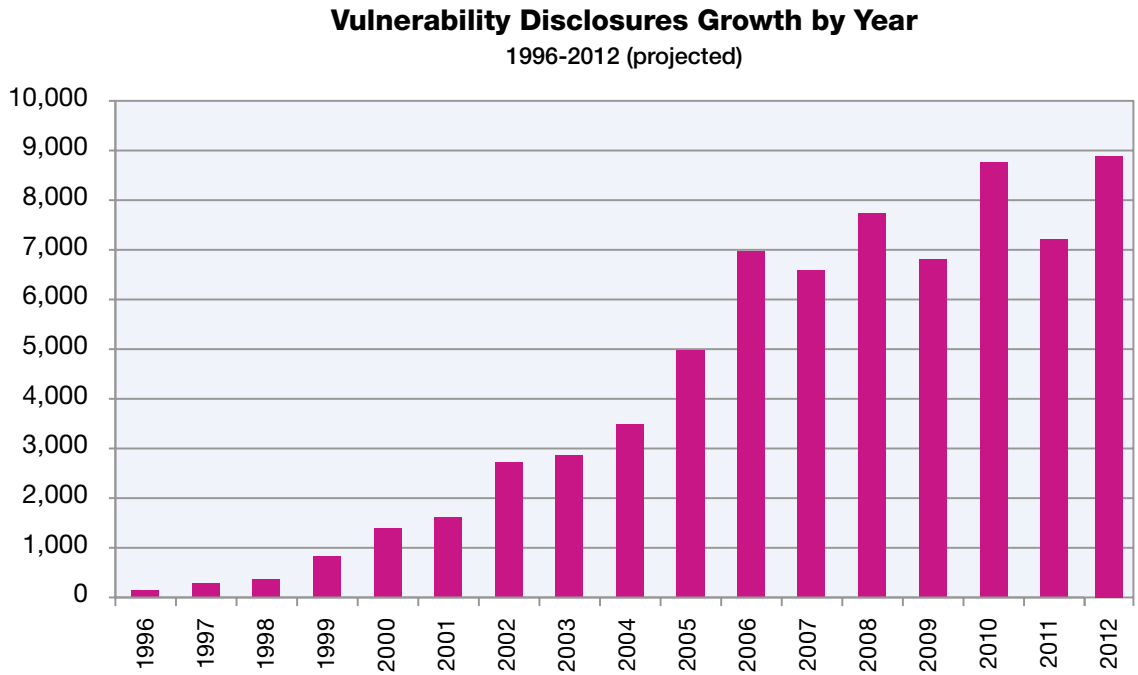


Figure 34: Vulnerability Disclosures Growth by Year - 1996-2012 (projected)

The decline of reported SQL injection vulnerabilities continued in 2012 but cross-site scripting vulnerabilities increased again to a projected all-time high. Cross-site scripting is a term used to describe web application vulnerabilities that allow attackers to inject client-side script into web pages that are viewed by other users. Over 51% of all web application vulnerabilities reported so far in 2012 are now categorized as cross-site scripting. This is a disturbing fact as cross-site scripting is a well-known and researched security issue. Our in-house data from IBM AppScan® OnDemand results from on-demand web application vulnerability scans indicated a greater than 40% likelihood of finding a cross-site scripting vulnerability in these on-demand scans over the course of 2011.

Over 51% of all web application vulnerabilities reported so far in 2012 are now categorized as cross-site scripting.

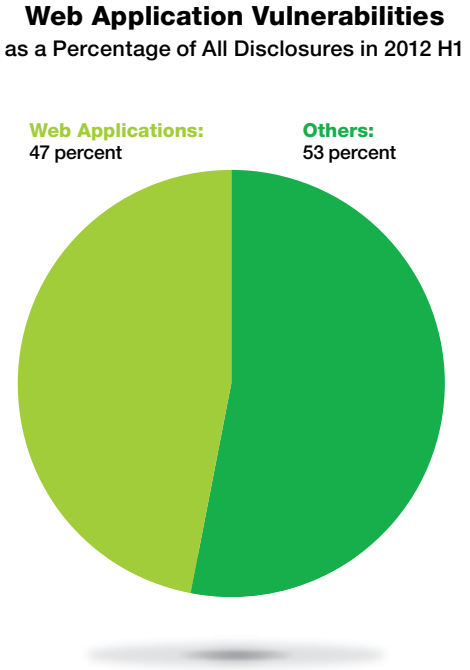


Figure 35: Web Application Vulnerabilities as a Percentage of All Disclosures in 2012 H1

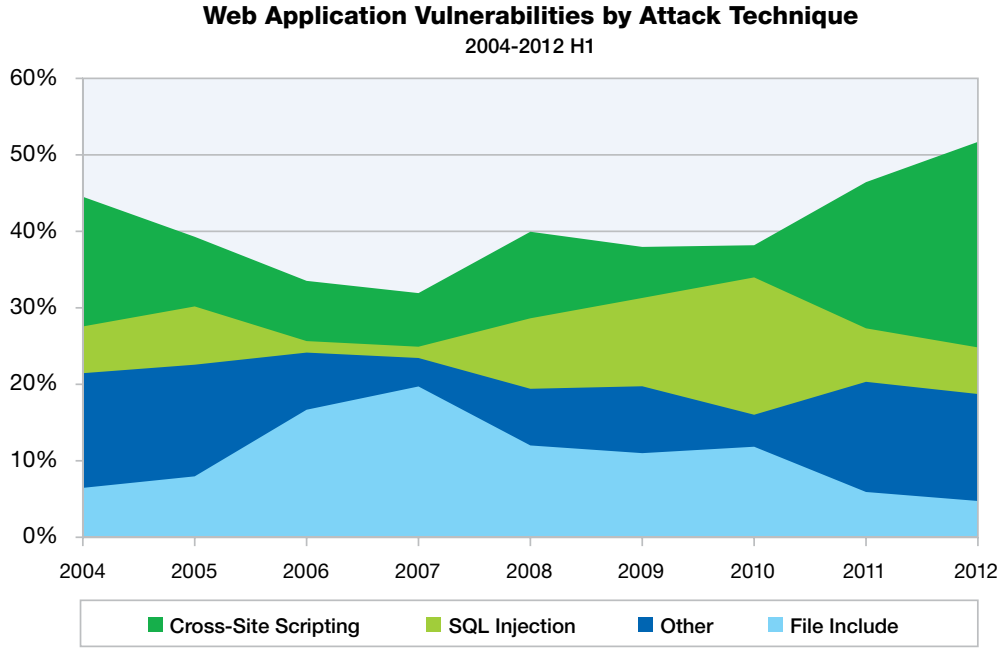


Figure 36: Web Application Vulnerabilities by Attack Technique - 2004-2012 H1

IBM X-Force has seen that a large amount of the web application vulnerabilities are disclosed on public exploit websites. Of these web applications, many can be attributed to plug-ins contained in the in-house designed content management systems (CMS) that are developed by website design companies. Often times these plug-ins are not available for purchase separately. However, once the website is up and running, it is hosted by the consumer on their own hardware and networks. A multitude of vulnerabilities can be found in these small company websites.

There are also widely used content management systems across the Internet. These major web-based CMS programs have become better at notifying the public when vulnerabilities are found in plug-ins written by third parties. We classify vulnerabilities in these CMS programs as core issues and plug-ins. Core issues are patched by the producing company that provides these systems at a much higher rate than the plug-ins written by third parties.

Figure 37demonstrates the percentages of vulnerabilities that are classified as core or plug-in issues.

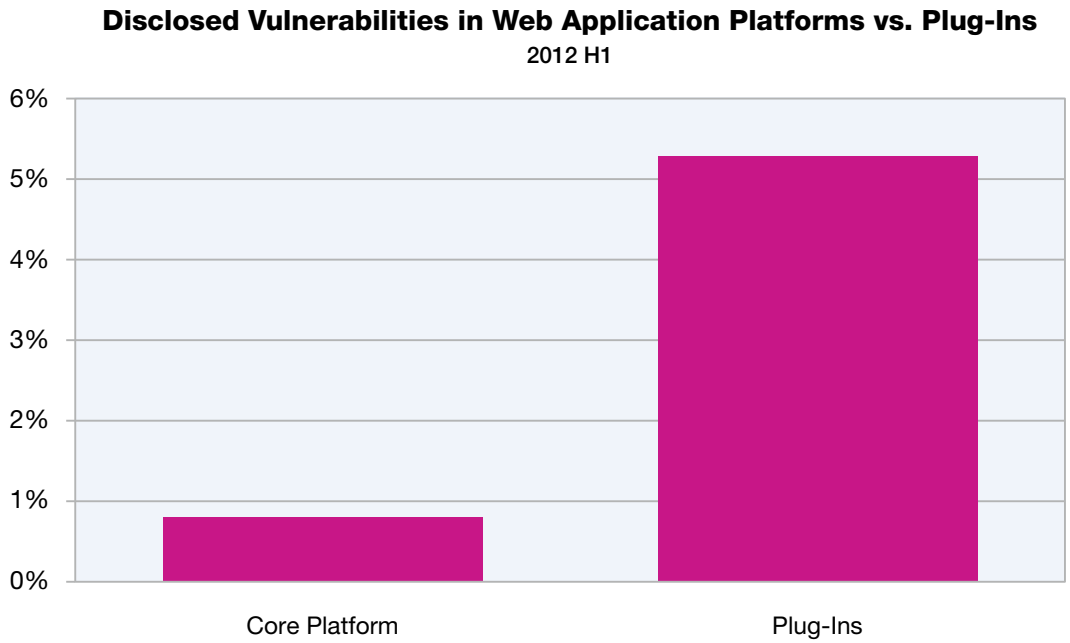


Figure 37: Disclosed Vulnerabilities in Web Application Platforms vs. Plug-Ins - 2012

As you can see, less than 1% of all CMS vulnerabilities are disclosed against the major market leaders of CMS producers. Of those leaders, slightly higher than 5% of the vulnerabilities exist in third-party plug-ins.

Patch rates are also higher for core vulnerabilities versus plug-ins. Many of the leading CMS programs have begun hosting vulnerable third-party extension lists to notify users and developers of those plug-ins that there may be an issue in a plug-in that they have deployed.

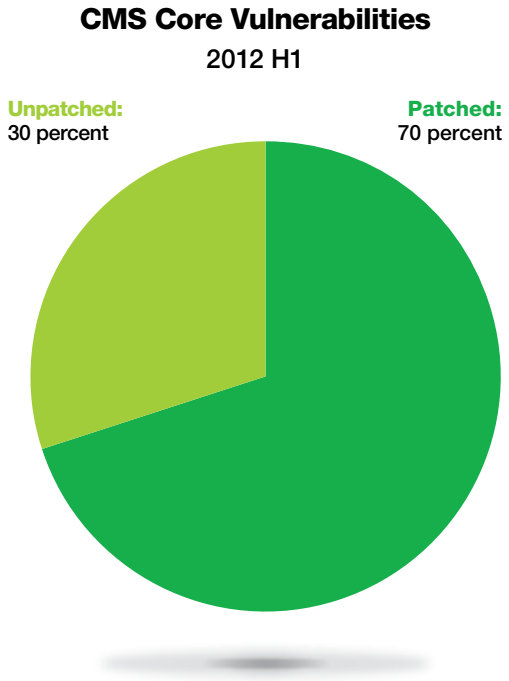


Figure 38: Disclosed Vulnerabilities in core content management systems - unpatched vs. patched - 2012 H1

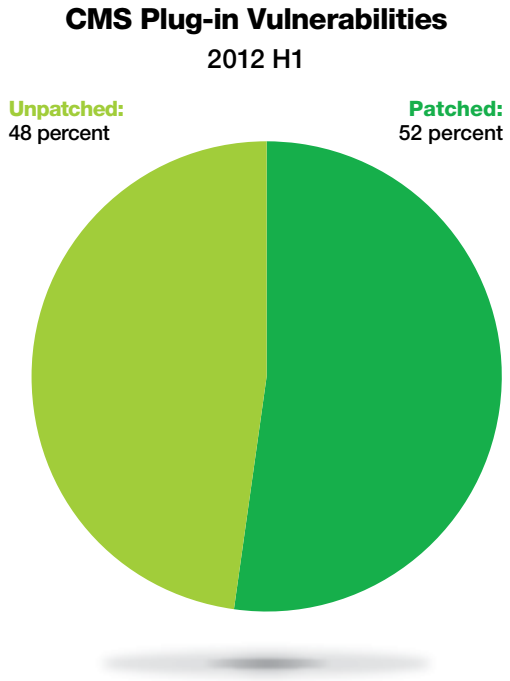


Figure 39: Disclosed vulnerabilities in plug-in content management systems - unpatched vs. patched - 2012 H1

Continuing decline in exploit count

In 2011, IBM X-Force saw a significant decline in publically released exploits. We catalog two categories of exploits. Simple snippets with proof-of-concept code are counted as exploits, but fully functional programs that can attack a computer are categorized separately as "true exploits." When comparing the number of true exploits against the total percentage of vulnerabilities logged in the database, interesting trends appear.

In 2009, the percentage of true exploits peaked at nearly 16% of all publicly disclosed vulnerabilities. Since then, we have observed a decline in overall vulnerabilities that had true exploit code available drop to almost 11% by 2011.

The trend continues into 2012, where based on data from the first six months, we project that only 9.7% of all publicly disclosed vulnerabilities will contain exploits. These percentages do not include many web application vulnerabilities that can be exploited through the use of the address bar in a standard web browser.

Looking closer (figure 40 to the right), we discover that the total number of true exploits is much lower than the high of 2010, though slightly higher than the total for 2011. However, when looking at true exploits as a percentage of the total overall number of vulnerabilities, as show in Table 6, we see it trending downward to a projected 9.7%. IBM X-Force believes that the decline in publicly available exploits is a direct result of architectural changes that have been made in software over the past few years that make exploiting these vulnerabilities more challenging.

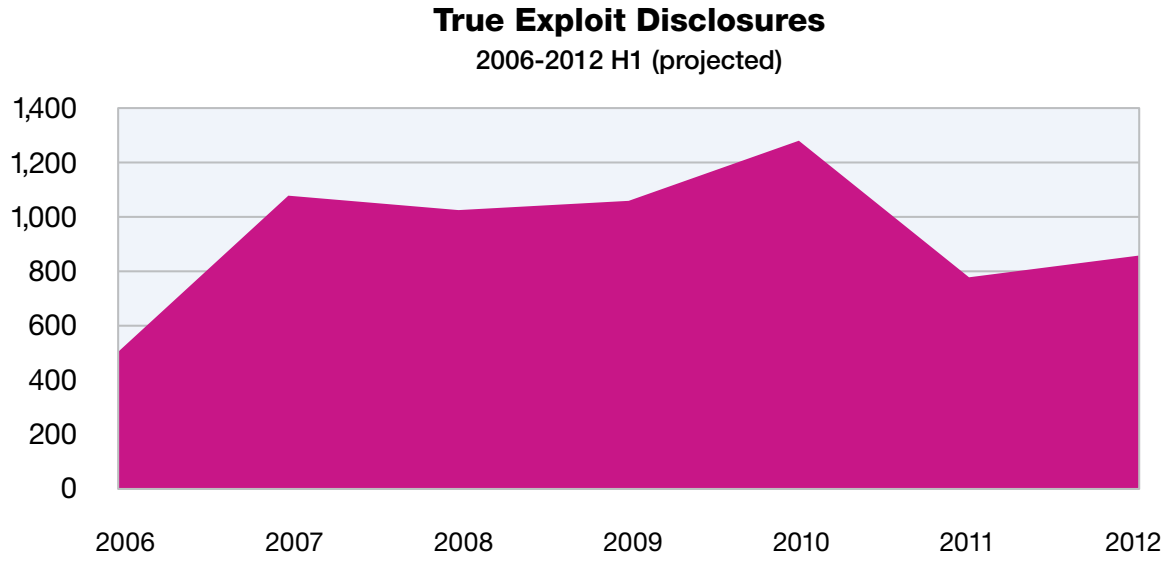


Figure 40: True Exploit Disclosures - 2006-2012 H1 (projected)

	2006	2007	2008	2009	2010	2011	2012
True Exploits	504	1078	1025	1059	1280	778	858
Percent of Total	7.3%	16.5%	13.3%	15.7%	14.7%	10.9%	9.7%

Table 6: True exploit disclosures - 2006-2012 H1 (projected)

IBM X-Force also saw that the number of Multi-Media based exploits remained the same as previous years.

One area in which we observed a significant decrease in publically available exploits is in the area of mobile operating systems. Mobile devices are becoming more and more a part of our daily lives. An increasing concern among mobile device users is the security of these devices. IBM X-Force has found that, in the first half of 2012, reported mobile vulnerabilities and exploits are down to the lowest levels since 2008. We think there are multiple things going on. First, mobile operating system developers are continuing to invest in both in-house discoveries of vulnerabilities as well as enhancements to their security models to prevent vulnerabilities from being successful. Next, as is typically the case with a new area like mobile, we tend to observe an initial spike in discoveries, but then as the easier bugs disappear, and hard to exploit ones are left, there is a lag between when researchers and attackers discover techniques to overcome previously perceived limitations. For example, the application of the "heap spray" technique to the browser vulnerability landscape around 2005 permitted memory corruption vulnerabilities to achieve reliable client-side exploitation as the spray would typically guarantee exploit code to reach locations in memory that previously were not controllable by non-programmatic methods. Though, we will note that heap spraying was not an entirely new concept.

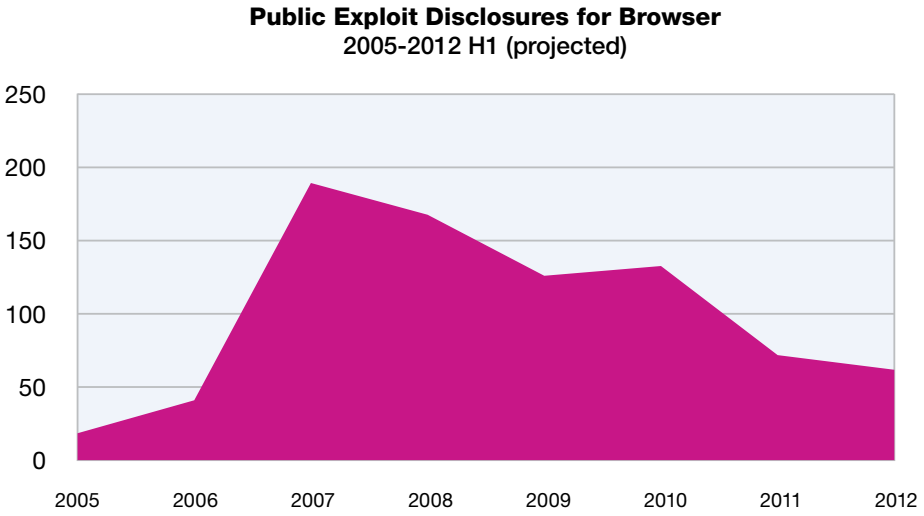


Figure 41: Public Exploit Disclosures for Browser - 2005-2012 H1 (projected)

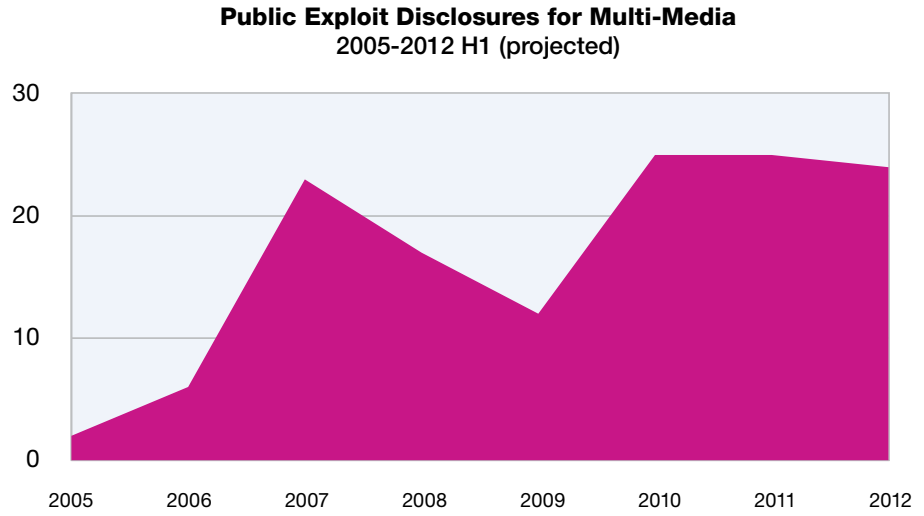


Figure 42: Public Exploit Disclosures for Multi-Media - 2005-2012 H1 (projected)

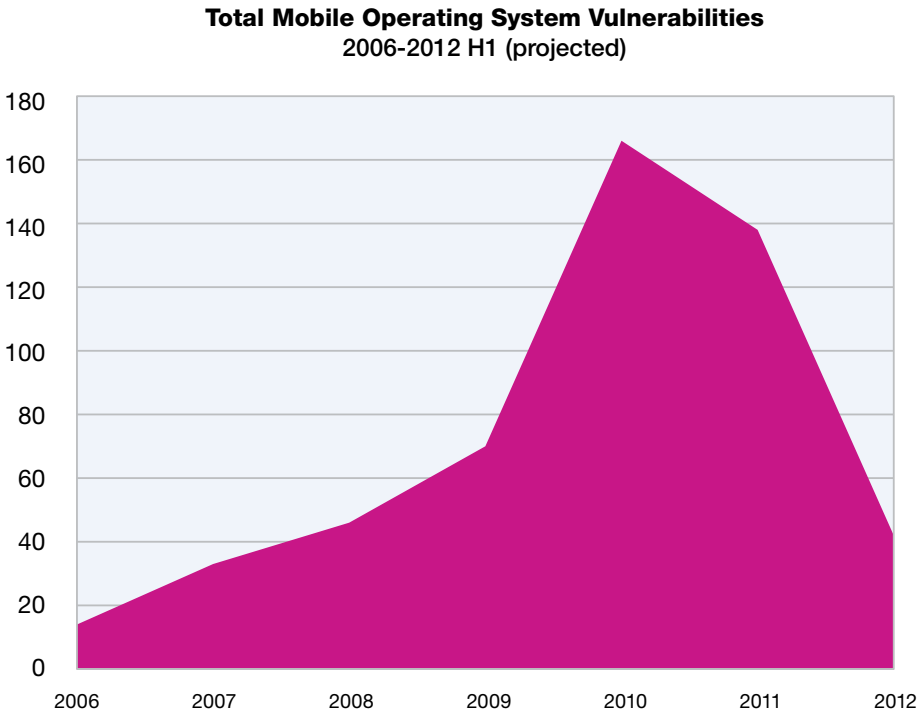


Figure 43: Total Mobile Operating System Vulnerabilities - 2006-2012 H1 (projected)

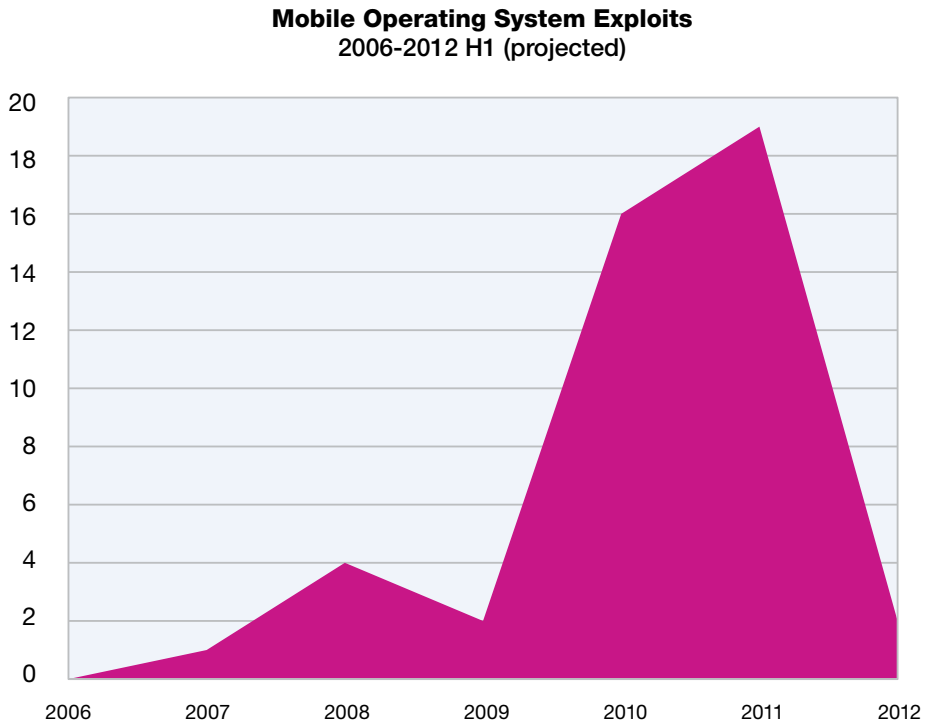


Figure 44: Mobile Operating System Exploits - 2006-2012 H1 (projected)

CVSS scoring

IBM X-Force scores almost every vulnerability that we research using the Common Vulnerability Scoring System (CVSS) based on severity. We score vulnerabilities from three different perspectives: as a vulnerability database that tracks third-party vulnerability disclosures, as a security research organization that discovers new vulnerabilities, and as a large software vendor that needs to help customers accurately assess the severity of vulnerabilities within its products. IBM X-Force is currently working alongside other organizations on developing the new CVSS version 3 standard. In the scoring of vulnerabilities for the first part of 2012, we found that the majority of issues fall into the medium range, with 27% of all vulnerabilities rated as critical or high severity.

CVSS Score	Severity Level
10	Critical
7.0-9.9	High
4.0-6.9	Medium
0.0-3.9	Low

Table 7: CVSS Score and Corresponding Severity Level

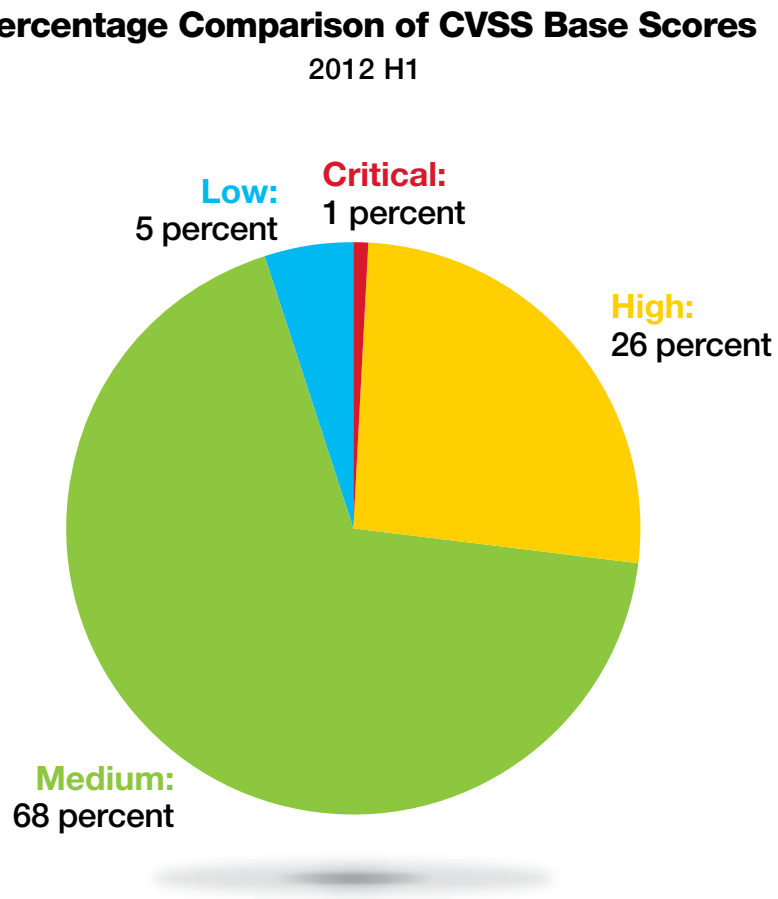


Figure 45: Percentage Comparison of CVSS Base Scores - 2012 H1

Vulnerabilities in enterprise software

When looking at trends in enterprise software, IBM X-Force looks at major software vendors who create the widest variety of enterprise software. We have observed that out of thousands of vendors, these companies consistently disclose a significant number of security vulnerabilities. We categorize these vendors in a top ten group, leaving out the CMS vulnerabilities since the majority of those are in third-party plug-ins and add-ons and not widely used as enterprise-level software. Since 2007, we have seen that the top ten have been increasing as a percentage of the overall disclosed vulnerabilities, with as much as 30% of all disclosures in 2011 coming from the large enterprise software vendors. However, in the first half of 2012, we have seen a decrease to 22% in the overall percentage of vulnerabilities disclosed by these companies.

It will be an interesting trend to track through the end of the year as the number of disclosed vulnerabilities in the second half of 2012 will determine whether or not we are seeing a notable downward trend or they remain relatively unchanged.

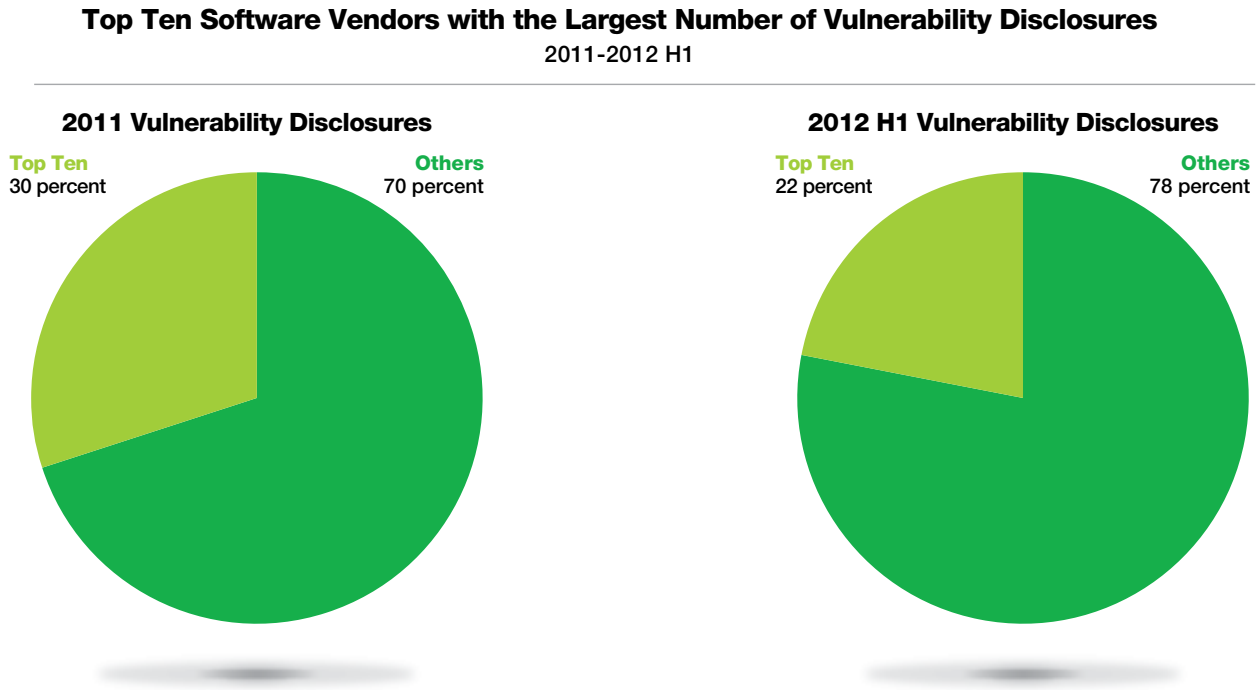


Figure 46: Top Ten Software Vendors with the Largest Number of Vulnerability Disclosures - 2011-2012 H1

One significant finding that IBM X-Force observed in the first half of 2012, was that vulnerabilities in Office and Portable Document Formats (PDF) declined sharply. We are confident that there is a strong relationship between the decline of PDF disclosures and the Adobe Acrobat Reader X sandbox. First, the sandbox should dramatically increase the complexity of creating a reliable exploit; we will return to that in a moment. Given a much higher bar of creating a reliable exploit, mere PDF vulnerabilities are less interesting for attackers to devote time to finding new ones. Sandboxes can provide this kind of benefit to the security ecosystem because they are designed to lessen the permissions that attackers and researchers are able to achieve on those affected systems. Consequently, IBM X-Force predicts a continued adoption of software sandboxes to help discourage attackers and mitigate many if not most existing attacks.

Web browser vulnerabilities declined slightly over the first part of 2012, but not at a rate as high as document format issues. IBM X-Force expects the number web browser based vulnerabilities to remain largely the same over the course of 2012.

IBM X-Force has seen great strides in the rate of patched vulnerabilities of the top ten vendors, which can be attributed to secure development practices and the continued implementation and improvement of Product Security Incident Response Team (PSIRT) programs. The top ten vendors have a patch remedy rate of just over 94% of all vulnerabilities disclosed.

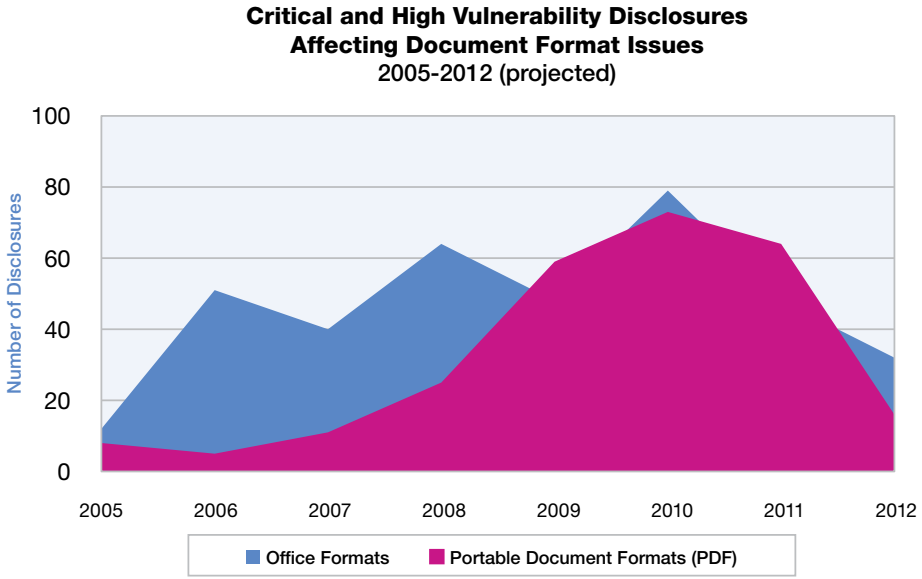


Figure 47: Critical and High Vulnerability Disclosures Affecting Document Format Issues - 2005-2012 (projected)

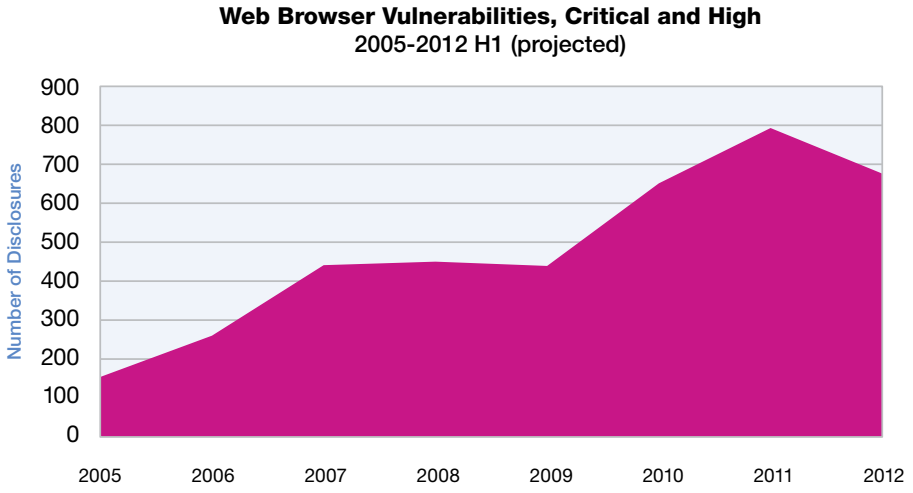


Figure 48: Web Browser Vulnerabilities, Critical and High - 2005-2012 H1 (projected)

This is good news for the top ten software vendors, however the same cannot be said for the rest of the vulnerability world. The rate of unpatched vulnerabilities for the first half of 2012 were the highest IBM X-Force has seen since 2008. 47% of all vulnerabilities disclosed this year remain without a remedy.

IBM X-Force does not necessarily believe the rise in unpatched vulnerabilities is a bad omen. Major enterprise software vendors are doing a much better job today than they were five years ago. We think that the increase in vulnerabilities in small web apps—and obscure software written by individuals or tiny companies—are responsible for the 2012 increase. Many of these vulnerabilities likely will go unpatched or unsupported for the lifetime of the product.

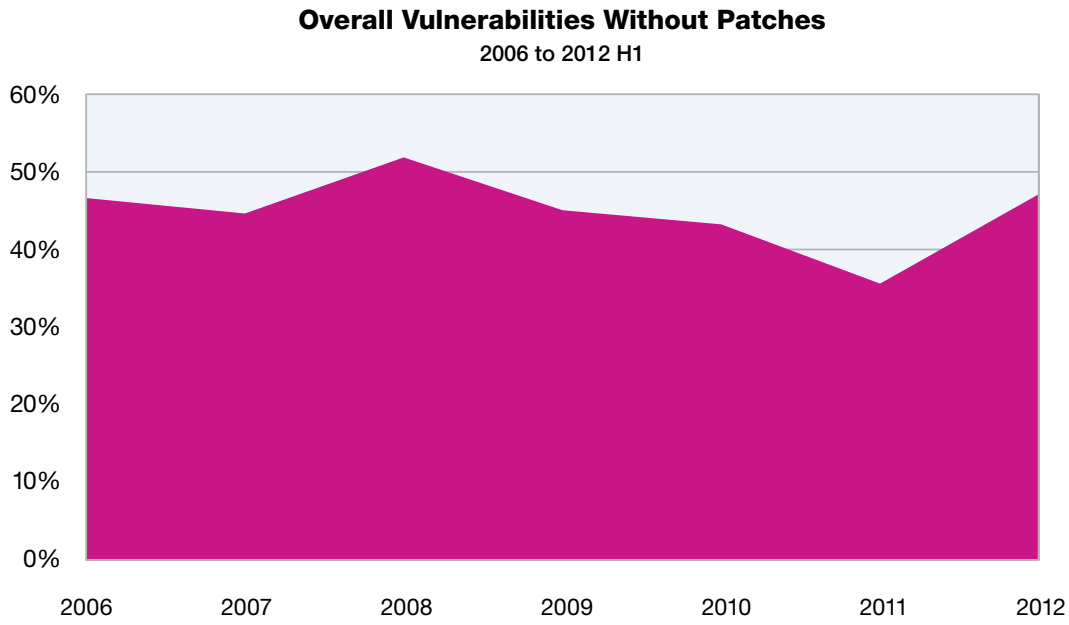


Figure 49: Overall Vulnerabilities Without Patches - 2006 to 2012 H1

Wrap-up

Compared to our last report, there are surprising findings in only a few key areas. As we just discussed, the percentage of unpatched vulnerabilities has increased though with the caveat that they are more obscure pieces of software that are not typically found in an enterprise. Second, the dramatic decrease in vulnerabilities and exploits targeting the mobile platform. Again, IBM X-Force sees a variety of different reasons for this, but recommends that readers remain vigilant on their mobile devices—whether their own, or managed by their employer. We recommend this because with all of the distractions of working-on-the-go, we can easily forget to apply to same security-centric thought processes. For example, one may not apply the same rigor to phishing emails received on their mobile device or pay close attention to security permissions a mobile application requires. Therefore, the simple attacks will be the ones most likely to happen in 2012 and we will explore this further in our mobile security section of this report.

The third interesting trending area relates to the effectiveness of software sandboxes in both mitigating attacks and discouraging researchers and attackers from finding and disclosing vulnerabilities that cannot make it past the lowered permissions of the sandboxed environment. For such attacks to be successful, multiple vulnerabilities need to be implemented—typically disabling the sandbox or finding a vulnerability in the sandbox such that the exploitation scenario becomes a two-part process of exploiting a vulnerability exposed by software and then leveraging another vulnerability against the sandbox to raise one's privileges enough to compromise the system. Some vulnerabilities against software sandboxes have been reported by researchers, such as IBM X-Force Researchers in 2011 and 2012, and likely used by advanced, targeted attacks as well. Software sandboxes are an exciting area for software vendors, security researchers, and security practitioners. We have discussed the decline in PDF vulnerabilities and exploitation based on countermeasures provided by Adobe. Now lets take a deeper dive into understanding sandboxing technology.

Sandboxes: Another line of defense

What is a sandbox?

Imagine receiving an alert that a burglar had successfully entered your house or office. Naturally one of your first concerns will be how to respond. But an equally important question is "what will be stolen and how much damage will be done?" The burglar has full access to all your possessions—jewelry, electronics, important business documents or intellectual property. They also have free reign to do whatever they want in your house or office, including destroy property. What if the burglar was hired by a competitor? Might they install concealed surveillance equipment in your office?

Now imagine this same scenario, but instead of your house or office, the burglar is a remote attacker and has just broken into your computer. The main job of a sandbox is to limit what this remote attacker can do or access once your system has been infiltrated.

How sandboxes work

Sandboxes work by isolating an application from the rest of the system so that when the application is compromised, the attacker code running within the application is limited to what it can do or what it can access.

There are multiple ways that sandboxing can operate. Some of the usual methods for isolating an application from the rest of the system are as follows:

1. **Resource virtualization—**Involves providing an application (or an entire operating system) with a set of virtual resources, such as virtual disks, so that changes to these virtual resources do not affect the actual resources. An example is the resource virtualization provided by virtualization software such as Xen and VirtualBox.

2. **Privilege reduction—**Involves reducing the privileges and capabilities of an application by using existing mechanisms that are provided by the operating system. Examples include the Google Chrome sandbox, the Adobe Reader X sandbox, and the different Adobe Flash Player sandbox implementations.

3. **Controlled execution—**The application is executed in a controlled environment where there is no direct access to the operating system. Specific interfaces must be used in order to perform privileged actions. An example is the Java sandbox.

Sandboxed applications use the services exposed by a higher-privileged application (usually called a broker) to perform privileged actions. The broker, on the other hand, consults a set of policies to determine if the privileged action will be allowed or denied.

Sandboxes can help you

Depending on how the sandbox is implemented and what policies are in place, a sandbox can offer the following protections:

- Helps prevent the installation of persistent malware on your system because write access to important resources is disallowed. The attacker should not be able modify critical parts of your system and should not be able install a malware that can survive a system reboot.
- Helps prevent information disclosure because read access to important resources and network access is disallowed. The attacker should not be able to access your personal files or send them to a remote location.
- Helps prevent damage to your system because modification to critical parts of the system and changes to the system configuration are disallowed.

Not all sandbox implementations are the same so it is very important to understand the capabilities and limitations of the sandbox implementation you will be using. You can consult the publications provided by the vendor and the research done by security researchers who looked at and evaluated the sandbox, such as the Adobe Reader X sandbox[17] research or the Adobe Flash Player sandbox research.[18]

What you can do now

One relatively unobtrusive way to reap the benefits of a sandbox is to determine whether the applications your organization is using have newer sandboxed versions, and if so, test them, deploy them, and use them.

You can start by looking at the applications that consume content from the Internet, such as document readers, media viewers, browsers and browser plugins. Fortunately, some vendors now provide sandboxed versions of their products. Some examples of sandboxed applications for the Windows platform are:

- For Web Content
  - Google Chrome
  - Internet Explorer 7 and later versions on Windows Vista and later operating systems
- For PDF Content
  - Adobe Reader X (also known as Adobe Reader 10) and later versions
  - Built-in PDF viewer in Google Chrome
- For Flash Content
  - Adobe Flash Player 11.3 and later versions (currently sandboxed in Firefox on Windows Vista and later operating systems only)
  - Built-in Flash viewer in Google Chrome (also known as Pepper Flash)
- For Documents
  - Microsoft Office 2010 (in Protected View mode)

Keep in mind that there are opportunistic attacks that target older, un-sandboxed versions of applications and a sandbox serves as another line of defense against these attacks.

17 https://media.blackhat.com/bh-us-11/Sabanal/BH_US_11_SabanalYason_Readerx_WP.pdf
18 https://media.blackhat.com/bh-us-12/Briefings/Sabanal/BH_US_12_Sabanal_Digging_Deep_WP.pdf

What we can expect

Implementing a custom sandbox is costly; the costs include research, development, testing, and maintenance costs. We believe that sandboxing capabilities for most of the off-the-shelf applications that need it will be provided by the operating system itself. This is currently being done as part of the AppContainer feature in Windows 8 and the App Sandbox feature in OS X. It is expected that there will be cases where the sandbox offered by the operating system does not offer enough granular control for some applications, and in those cases, custom sandboxes should still have their place.

In any case, operating systems likely will continually be updated to include additional mechanisms that restrict the privileges and capabilities of an application, and most of these restrictions will be applied by default, because if you think about it, every application you run doesn't really need access to your personal documents.

Attackers will adapt

As vendors continue to integrate sandboxing capabilities with their products, attackers will need a separate vulnerability to fully compromise a system. From an attacker standpoint, this means increased costs in the development of a complete attack, and it should also mean that sandbox escape vulnerabilities will become more valuable. Attackers will likely adapt by allocating additional investments into finding and/or acquiring sandbox escape vulnerabilities.

Final thoughts

Of course, sandboxing technology is not fool-proof and a motivated attacker with enough resources can find ways to break out of a sandbox, so we still need to be vigilant. Complacency can cause us trouble—even if someone gives us a helmet and a bulletproof vest, it doesn't mean that we can run around in a firing line, feeling invincible. Reducing the attack surface by uninstalling or disabling unused applications, unused features and unused browser plugins, and keeping your software up-to-date. Also, educating your users about the dangers of opening unsolicited content, always makes good sense.

Auditing made easier with UNIX shell history time stamping

For computer forensic analysts, discerning the time at which events occurred is a major challenge when investigating a security incident. UNIX provides a useful auditing system in the form of the shell history file. By default, this file is not always set up to log a time stamp along with the command, making it difficult to correlate events to a timeline.

IBM Emergency Response Services (ERS) analysts who have performed detailed computer postmortem analysis of dozens of UNIX/Linux cases have noticed that only a mere handful of those appear to have had the HISTIMEFORMAT values set. IBM X-Force believes that this setting aids in understanding the time at which commands were issued by Unix/Linux system users.

For example, the shell history file might record an instance where a user typed **`ping 192.168.100.10`**, but unless there was a packet sniffer or a firewall log entry, it is not apparent precisely when they typed the command.

ERS analysts who work on Unix—specifically Linux for the purpose of this article—support the value of adding time stamps to the history file. We hope to raise awareness of implementing this technique on production servers for security analysts and system administrators.

Unix command shells like C shell (csh), Korn shell (ksh), and Borne again Shell (bash) provide a 'history facility' that is kept as an account of individual activities. Basically, it keeps a record of each command typed (and mis-typed in the case of mistakes) into the command-line environment by a logged-in user.

Computer forensic analysts can use the content in this history file (.bash_history) to retrace activities whenever an computer account intrusion is believed to have occurred or similar suspicious events are under investigation.

However, there are several possible problems. The data in the history files that belong to a user account is not immutable and can be altered or destroyed.

Also, a feature to put the activity records into a precise timeline is a seldom used feature.

Consider a hypothetical user Joe Black, who types in the following commands while working at fictional company Acme.

```
telnet fs1.acme.com
```

Upon access to the fs1.acme.com resource, Joe issued these commands according to his history file:

```
mail bigcheese SUBJ: Resignation
rm -rf *
```

The presence of these commands suggests that Joe Black logged into FileServer1, communicated to his boss something pertaining to a resignation, and issued a command to destroy data. But without time stamps, you cannot know when these activities occurred which is a critical factor in explaining to staff and executives when analysis is performed and put into meaningful terms. Let's examine a possible solution.

For the purpose of this discussion, we will examine a Linux host computer system and command-line interaction with the Borne Again Shell (bash). When we examine the configuration of the users' profile, we can introduce a feature to institute time stamping in a way that is useful for the examiner, as well as the managers and system administrators of the Linux host.

The following line of code, which can be added to the /etc/profile file, puts the change into effect:

```
export HISTTIMEFORMAT="%s %T%z
%d/%b/%y "
```

Essentially, this places a time stamp for each command entered while a user is logged on and interacting with the shell. Furthermore, the date set up allows for times to be inserted as Unix Epoch Time, which can simplify parsing these history files, and it puts the time in human readable terms. Unix Epoch time is the number of seconds that have elapsed since 00:00 (UTC) on January 1, 1970. The spaces in the above command line improve the readability of the output.

The output also reports the time zone in effect when the entry was made, which helps confirm the setting or detect any misconfiguration. This is important as IT professionals should understand what time zone is active when examining the records.

There is one important caveat to consider when setting this up. You have to archive existing history files if you are setting this on a system that already has an established history. It turns out that if this is not done, then all events prior to the moment the 'export=HISTTIMEFORMAT' command was entered into the profile will possess the incorrect date. Clearly, this can be a serious problem.

Before setting the HISTTIMEFORMAT, be sure to back up and archive any existing .bash_history files. One approach is to use a 'for-do'-loop that seeks out and finds past .bash_history files and then archive them in a tar-gz file (using bash shell on Linux).

```
$ sudo tar -czvf `date "+%d%e%Y"-
history.tar.gz ` $find( -f /home -type
f -name '*history')
```

The result is the creation of a 'tar.gz' file that has the date of when the history files were backed up. After running this command, it is possible to clear out the existing .bash_history file (by renaming it, which is always reversible,) and then implementing a time-stamp by invoking the following:

```
$ mv ~/.bash_history ~/.OLD_bash_
history

$ echo export HISTTIMEFORMAT="%s %T%z
%d/%b/%y " >> ~/.bash_profile

$ history -c && exit
```

When calling up the history (by invoking the 'history' command), you would see entries similar to those shown here:

```
$ history
1 1341870050 14:40:50-0700 09Jul2012
history
```

This means the .bash_history file maintains time-stamped records in the actual file. The file itself would maintain a record of the time stamp along with the command as seen here:

```
#1341870056
exit

#1341870112
tcpdump -i eth0 host 192.168.100.12
-s 0 -w ./PacketCapture.pcap

#1341870112
tcpdump -n -r PacketCapture.pcap

#1341870452
history
```

If you call up the history record (by entering in the command 'history' into the shell) you are presented with the following sort of display because the Bash shell uses the format of the HISTTIMEFORMAT variable to present the data in a way useful to the user as shown here:

```
1 1341870056 14:40:56-0700 09Jul2012
exit

2 1341870112 14:41:52-0700 09Jul2012
tcpdump -i eth0 host 192.168.100.12 -s
0 -w ./PacketCapture.pcap

3 1341870274 14:44:34-0700 09Jul2012
tcpdump -n -r PacketCapture.pcap

4 1341870452 14:47:32-0700 09Jul2012
history
```

Not only does this provide a quick understanding of what commands were entered and the order in which they were entered, but it is also clear what time zone the host is in.

The reason one sees just the Unix Epoch time entry in the 'raw' history file itself versus seeing both the Unix Epoch time and the human readable entry (14:47:32-0700 09Jul2012) is worth noting. Once the shell sees that the HISTIMEFORMAT value has been set, the records are stored in the file in the most-machine precise manner (Unix Epoch time) and the display renders a simple conversion which makes it meaningful to those reviewing the history records. It is also helpful to members of the ERS team as they can easily search a file system for any deleted entries and recover those that appear to have the time-stamp data structure once they learn a system had enabled the HISTIMEFORMAT values.

Consider correlating event logs, file time stamps, or network packet captures that record when the log entry was made, the time a file was accessed or modified, or the time when packets traversed the wire. This allows you to get closer to attributing actions and observations with the shell command history.

To illustrate the simplicity of this time-stamped history in action, below is a listing of events that occurred when correlating logs and network packet capture files, that were run through an open-source tool known for artifact timeline analysis.

In the following listings we took the root-user .history files from a server designated as 'VICTIMSRV' and integrated them into the syslog event log files, file activity time-stamp attributes (such as modified, accessed, created, and entry-updated), and packet captures. The values of these four disparate sources of data clarify not only the order of events, but also the commands issued by the root user.

```
Tue Jul 10 15:02:17 2012 Z
PCAP   192.168.100.10  - -  ICMP
packet 192.168.100.10 ->
192.168.100.12|PST8PDT|File: KSServer.
pcap.pcap inode:1872361
HISTORY VICTIMSRV     root - ping
192.168.100.10

Tue Jul 10 15:01:58 2012 Z
LOG   VICTIMSRV   - - (Linux Syslog
Log File) [Entry written] [passwd]
log event on [victimsrv] by [pam_
unix(passwd:chauthtok)] : "password
changed for tmillar "|PST8PDT|File:
secure inode:11334
FILE   VICTIMSRV   - MA.E     /etc/
shadow

Tue Jul 10 15:01:32 2012 Z
HISTORY VICTIMSRV     root - passwd
tmillar
```

It is clear when the user ID root changed the password for the user ID tmillar. There is not only a log entry which backs up that assertion, but also the /etc/shadow file reflects that a modification was made at that time. Also, the command issued from within the history file shows that the command was issued in the moments prior to the changes. One can easily spot that the user typed in ping 192.168.100.10 and this entry record correlates nicely with that of the time-stamped entry within a packet capture performed on the network using another host.

Adding a time stamp to the shell history can tie together many disparate pieces of information into a succinct timeline of events.

There is a chance this is already enabled on UNIX/Linux hosts, specifically critical servers. If not, consider our recommendations.

As forensic analysts, it is important for you to be aware of how to enable time stamps in history files. Optimally, it would be beneficial if this is implemented before one attempts to investigate an incident. By establishing the HISTTIMEFORMAT value, it should be easier to quickly recall commands used to maintain the system or even spot out-of-the-ordinary occurrences when they occur.

Evaluating the cyber terrain with OCOKA

With discussions of cyber warfare, attackers, and defenders, one needs the ability to evaluate a network as a terrain on which a battle will be fought. This is a battle between the attacker who is attempting to gain access, steal data, destroy information, or commit crime and the defenders who seek to protect their networks from the attackers. Networks can be thought of as a terrain when you consider the similarities: perimeters, access points (gateways), challenge and password (username and password authentication), key terrain (accounts, servers, and sensitive data), observation posts (IDS/IPS), and those that occupy and defend the terrain (users, security).

The military has a process for almost everything they do and one of these processes is to evaluate the terrain a unit is going to defend or move through. This same evaluation process, remembered using the acronym OCOKA—Observation, Concealment, Obstacles, Key Terrain, and Avenues of Approach—can be used to evaluate the terrain of your network environment from the perspective of both the defender and the attacker.

O Observation

Observation is the ability of the network defenders to observe the activities of the attacker and the attacker's ability to view and obtain data about and from the network. Observation methods frequently include:

Defender

- Network logs—firewall, Intrusion Detection/ Prevention System (IDS/IPS), VPN, and Proxy
- Server log files—DNS, Domain Controller, and Anti-virus network console
- Host log files—Windows event logs, AV scan logs, firewall logs, and Linux access logs
- Application logs—web, email, SharePoint, and FTP
- User awareness training that creates a "call 911" culture for suspected security events

Attacker

- Reconnaissance to identify system and data exposures. Information recovered from this process can range from the discovery of remote access and application portals requiring no authentication to exposed vulnerability scan reports.
- Network packet captures and samplings using tcpdump, sn.exe or similar programs are used to attempt to capture data or identify network segments that have credit card or other sensitive data.
- Nmap and other scans of the external and internal networks can be used in an effort to identify key areas of the network to attack.
- Physical access to the facility can be used to gain information about the network.
- Monitor and compromise email accounts of executives and incident responders. This can be done with something as simple as an account forwarding rule for emails.
- Use of local administrator account to conceal account use from network observation efforts.

Recommendations

Validate and monitor defensive observations systems. For defense observation mechanisms to be effective, they must be functioning properly, monitored, and the alerts must be responded to with an appropriate, planned response. It is not uncommon for incident responders such as the IBM Emergency Response Service (ERS) to discover during an incident investigation that the logging mechanisms weren't functioning properly (inadequate log size resulting in frequent rollover of logs or logging only success events). Or extensive indications of malicious activity are present in the logs but, since no monitoring of the logs was being conducted, the intrusion went unnoticed. With sufficient monitored observation capabilities, the odds of detecting an attack increase. With little or no observation capabilities or capabilities that aren't monitored and responded to, the likelihood of a successful, undetected attack increases.

Obtain a situational awareness of the threats. While not directly an observation method, defenders should participate in organizations such as the IBM X-Force Threat Analysis Service (XFTAS), FIRST, and Infraguard to gain an understanding of the current threats and attack trends. This provides a situational awareness of the current trends in attacks and helps to better recognize attack indicators when they observe them.

Train and provide. Train security staff to examine logs, evaluate contents for indicators of malicious activity, and respond to events and incidents. Provide them with observation capabilities suitable for security observation, not just performance observation. Provide security staff to monitor logs for security events within the network.

C Concealment

Concealment refers to the ability of the network defenders to conceal the network architecture and data, especially high-risk portions of their network or data from attackers. It also includes the ability of the attacker to hide their malicious actions from the defenders. Several concealment techniques include:

Defender

- Encryption to conceal data from unauthorized access by requiring a key to gain access to data, whether at rest on a drive or in motion on a network.
- Implement "security through obscurity" by using non-predictable naming conventions for host names and user account names.
- Use network address translation (NAT) to make it difficult to identify hosts within a network from the Internet.
- Limit the amount of data that is publically available on corporate and social networking sites that can be used for exploitation by Open Source Intelligence (OSI) efforts.

Attacker

- Compromise and use legitimate user accounts to co-mingle legitimate and malicious activity.
- Tunnel malicious traffic through encrypted tunnels, frequently to common destination ports such as port 80 to give it an appearance of legitimate traffic.
- Exfiltrate data using compressed files uploaded to public Internet file-sharing sites.
- Access the targeted network from multiple source IP addresses to conceal the actual source of the attack.
- Use local administrator accounts to conceal account use observation at the network level.
- Disable antivirus software during attacks and malicious activities.

Recommendations

Conduct your own OSI data gathering. Search social networking and other sites for data related to your organization. Items to look for include: postings by employees on technical forums that provide information about internal network structure and configuration; information on sites where data related to vulnerabilities, intelligence information, and compromised passwords and accounts are posted; and employees posting about company activities which may provide information useful for phishing attacks.

Develop the capability to identify unauthorized communications. Attempt to identify encrypted connections with a destination port that is not typically associated with encrypted communications, such as port 80, or SSH protocol destined for ports typically associated with SSL.

Develop the capability to monitor use of local administrator accounts: Because attackers prefer to hide their activities from observation by using local administrator accounts, develop methods to collect and monitor that information from the hosts. This could be done via security information and event management (SIEM), a syslog, or by having a script to collect that information from systems. Attempt to identify patterns of local administrator account use that deviates from the normal quantity and duration of account use.

Develop the capability to identify normal account use that is occurring outside of normal work hours. Many attackers are from a different time zone than the systems being attacked and use of stolen account credentials during the attacker's normal "work" time, due to time zone changes, may occur during the defender's normal "off work" time. Attempt to establish a pattern of normal activity and then watch for significant deviations from that pattern.

O Obstacles

Network defenders and attackers frequently place **obstacles** in each other's way in order to deter or obstruct the ability to successfully defend or attack the network. Some of these obstacles include:

Defender

- Complicated passwords or two-factor authentication.
- Network Access Control Lists.
- Encryption of data at rest and data in motion.
- User awareness training.
- Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS).
- Antivirus and other malware scanners.
- File integrity and monitoring systems.

Attacker

- Log file deletion.
- Cleanup routines within their attack—batch files that clean registry keys, delete pre-fetch files, delete and overwrite malware and exfiltration files to hinder efforts to determine malware capabilities and the content of exfiltrated data.
- Theft and use of legitimate network credentials to co-mingle attacker activity with legitimate activity and hinder detection and investigation efforts.
- Use of local administrator accounts to conduct activity that is not likely to be visible at the network level.
- Disable antivirus software during the execution of the attack and during the installation of malware.
- Deletion of entire file systems.

Recommendations

Implement defensive obstacles. The coverage of defensive obstacles should overlap, creating several layers of obstacles that an attacker must overcome before gaining access. Attackers rely on the failure of the defensive obstacles to provide them access—frequently a user runs the email-attached malware on a system where the passwords are stored with a weak hash, allowing for easy determination of passwords. This provides access to a network with no internal segmenting or access control lists and to sensitive data that is not encrypted.

Anticipate and plan for obstacles from attackers. Within your computer security incident response team (CSIRT), practice war-game scenarios where you ask yourself the question "What would we do if the attacker…?" Focus not just on the "what" but also on the "how" of the response. If your action is "obtain these logs"—do you know who to call to get them, who to call if they are on vacation, and does that person have the access and skill to get what you need? If the attacker deleted local host logs, does your organization have logging at a central location? Does your CSIRT have the access and skills needed to obtain those logs? If an attacker commits their attack through extensive use of local administrator accounts, is there a way to identify a spike in the number of local administrator accounts within portions of the network? Does your CSIRT have visibility when a user account disables AV software? If so, what follow-up is conducted to determine whether it was legitimate activity or the action of an attacker? These are examples of some of the issues to recognize and deal with during an attack.

K Key Terrain

Key terrain refers to areas within the network which contain high profile, high value, or high payoff targets. Key terrain can include servers, accounts, and individuals. An example of a high profile target would be a public-facing web server for a hacktivist who wants to embarrass the organization or use it as a platform to make a public statement. High value targets may include compromising accounts related to senior executives or systems used for payroll and other banking or financial transactions. Areas within the network that could be considered high payoff targets include networks containing credit card databases, personal information useful for committing identity theft, or medical information useful for committing fraud.

Recommendations

Identify your key terrain and make sure it is properly protected and well monitored. Develop an inventory list of all key terrain for your organization. This can be the typical high-value targets such as a domain controller but should also include high-payoff targets such as the organizational management, payroll, human resources, corporate legal, and locations of confidential intellectual property.

Develop a damage assessment process. If key terrain is compromised, you should identify the content that was exposed or compromised, the nature of the data, a risk assessment resulting from its exposure, and a list of mitigating actions to take to reduce the risk. Mitigation strategies should be assigned to and owned by individuals to ensure the actions are completed. This damage assessment could take the form of the examples in the following table:

File Name	Nature of Content	Content	Risk	Mitigation Strategy	Residual Risk
Vuln_scan.txt	Network Security	Vulnerability scans from 2010	Medium	Verify vulnerabilities were remediated	Low
Payroll.xls	HR	List of employees and bank accounts	High	Notify employees	High
Vacations.doc	HR	Employee vacation schedule	Low	None	Low
Passwords.xls	Network Operation	List of passwords for network devices	High	Change passwords within 8 hours; increase monitoring	Med

A Avenues of Approach

Frequently referred to as the attack vector, **avenues of approach** identify the mechanism by which an attack can be executed. Among others, these avenues of approach frequently include:

- Social engineering email containing malware or links to malicious websites.
- Dictionary and brute force attacks against Internet accessible webmail or other remote logins.
- Application vulnerability attacks (misconfigurations, buffer overflows, etc.).
- Physical access to the network such as a cleaning crew using password-cracking boot CDs.
- Corporate wireless signals accessible from neighboring businesses or the parking lot.
- Installed rogue wireless access points.
- Distributed-denial-of-service (DDOS) attack.

Recommendations

Implement technical solutions. Even though determined attackers frequently attack from several different avenues of approach simultaneously or separate, unrelated attackers may be attacking from several avenues of approach at the same time. You can address these attacks by vigorous security awareness training, complicated passwords, and vulnerability assessments. Avenues of approach can be interdicted with sufficient technical solutions (update patching, AV software, and robust passwords stored using secure methods).

Implement individual user solutions. Technical solutions can be circumvented by the user who runs the malware attached to the email, providing remote access to the attacker. This can be addressed by the implementation of a good security awareness training program. The goal of your awareness may be to build a “call 911” culture in a manner similar to calling the fire department when you see smoke, where users are encouraged to call security when they suspect a security incident. Your organization may also track users who are “frequent flyers” for AV alerts or security issues and chose to take enforcement action against them for their unsafe actions. This action can range from transitioning them to a different operating system less prone to malware up to and including disciplinary actions for their frequent exposure of the network to attackers due to their unsafe computing activities.

Network defenders should identify defensive strategies within each of the OCOKA categories and anticipate and prepare for attacker actions within each OCOKA category. All aspects of OCOKA are impacted by user’s actions and a recommendation common among several of the OCOKA areas is user awareness training to develop and foster a risk aware culture and management system, training users to recognize, report and properly respond to security threats. For a network defender, OCOKA can be a valuable tool to help assess the terrain of the network. Based on the results of an assessment of the network terrain using OCOKA, the security administrators can obtain a broader situational awareness of the defensive capabilities of their networks helping them better prepare for, defend against, respond to, and recover from an attack.

Using perimeter security to take the risk out of file transfers

More than ever, the security of data is at the forefront of the greater public's radar. Over the past 18 months there have been many high-profile data breaches from many sectors including government, healthcare, and financial services. Through June 2012, there have been 214 documented breaches with over 8.5 million records exposed.[19] These breaches are widely known because of their direct impact to consumers, but this data fails to show the complete data breach picture.

The 2011 Cost of Data Breach Study conducted by the Ponemon Institute and sponsored by Symantec reports that the cost of a data security breach in the U.S. was $5.5 million which represents a 24% decline from the cost in 2010 which was $7.2 million per breach.[20] Organizations are aware of overall security, but what about their files? How secure are they?

These are the questions companies face on a daily basis. How will I protect my enterprise? What part of my enterprise needs to be protected? Is a firewall enough? Are virus scans on our machines enough?

There are many questions about security, and only recently have CIOs and corporate security types become aware of business to business (B2B) and file transfer security.

Every day billions of files are sent over the Internet with little thought about security. People send emails with confidential information daily. Corporations send sensitive data over their internal networks and outside the enterprise without thinking about the potential issues that could face them.

19 Identity Theft Resource Center, 2012 Data Breach Stats, July 3, 2012, http://www.idtheftcenter.org/ITRC%20Breach%20Stats%20Report%202012.pdf
20 Ponemon Institute, 2011 Cost of Breach Study United States, March 2012, sponsored by Symantec, http://www.symantec.com/content/en/us/about/media/pdfs/b-ponemon-2011-cost-of-data-breach-us.en-us.pdf

For the last 40+ years, some form of File Transfer Protocol (FTP) has been the standard bearer for sending files from person to person or business to business. It was created even before TCP/IP and is still used in its base form largely unchanged today. FTP gave people a way to transfer files from point to point or machine to machine, but 40 years ago we were not as concerned about security as we are now. It is well known that the protocol is insecure because it sends passwords and data in plain text over the network.

In addition to the issues with the protocol itself, many file servers sit unprotected and vulnerable to attack. Over the last 40 years there have been major improvements in the File Transfer space with the addition of SFTP (FTP using SSL), FTP/S (FTP over SSH), HTTP, HTTPS (HTTP using SSL), and many other messaging and file transfer protocols both proprietary and open. Even with the advent of these more secure protocols, security issues exist.

By 2015, analysts forecast the B2B market to be $2.22 billion[21] and Managed File Transfer (MFT) $2.48 billion[22] in annual revenue. Companies are adapting to the changing times, but so is the world. As more and more corporations look to lock down their file transfers, the need for a strong file transfer security strategy is vital. With millions of files coming in and millions of files leaving these enterprises yearly, it is important to have a strategy in place to make sure that data is secure.

With the growth of the MFT market comes new ways to transfer files and data in an ad hoc fashion. One prominent space for ad hoc file transfer today is cloud file storage providers like Dropbox. Public cloud services provide a way for people to share files by simply uploading a file into a virtual folder housed in a large data center. While convenience may the biggest driver to these services, security tends to fall short of enterprise-level standards.

Many vendors have started taking the ad hoc problem to task by providing enterprise-level ad hoc solutions with security on par with more traditional managed file transfer solutions. New products are being released that integrate with other enterprise security and managed file transfer applications to provide well-defined perimeter security.

Securing your perimeter

Perimeter security is not a new concept, but is still not widely implemented. The results of the Ponemon Institute's Best Practices in Data Protection survey showed that 55% of the 718 IT and IT security practitioners responded that they lack a formal strategy governing the security of moving data.[23] That is a staggering statistic given the likely thousands of trading partners that send and receive files from a large organization. It is likely that the majority of the other 45% of respondents are financial services organizations that are faced with very stringent security requirements, but they are not the only industry dealing with confidential information.

How secure is your organization's perimeter when dealing with files and confidential information? Do you have a strategy? What can you do about this?

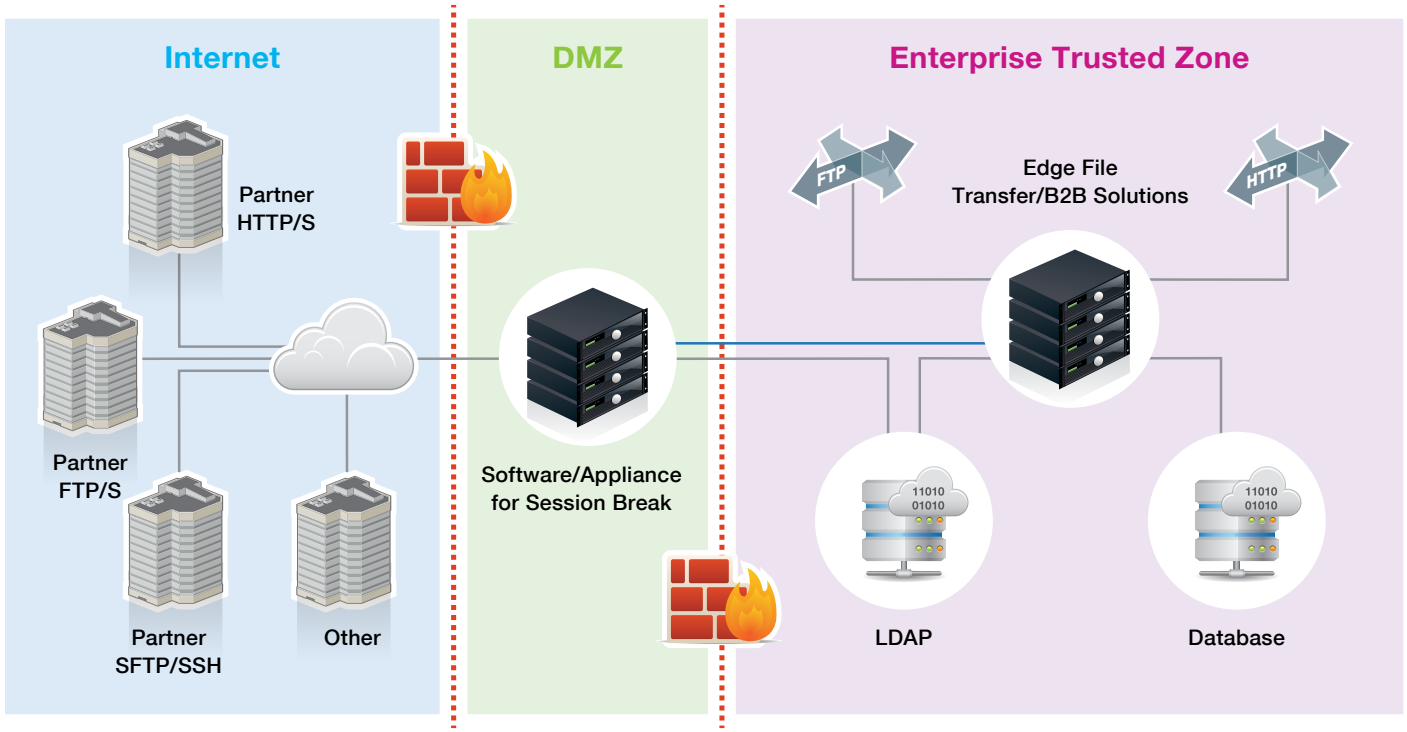
Does your enterprise have a strategy around perimeter security? If not, why? What are the next steps? These are the questions that should be answered to determine what will work for your company and your industry. What works for one organization may not work with yours. Does your

21 IDC, Worldwide Horizontal Business-to-Business Middleware 2011-2015 Forecast, August 2011
22 Ken Vollmer, Forrester Research, Market Overview: Managed File Transfer Solutions, July 2011
23 Ponemon Institute, Best Practices in Data Protection: Survey of U.S. IT & IT Security Practitioners, October 2011, sponsored by McAfee, http://www.mcafee.com/us/resources/reports/rp-ponemon-data-protection-full.pdf

traffic consist of mainly large files or do you receive large quantities of smaller files that are expected to be transmitted in real time? Working with your security and IT-governance teams to determine your file transfer requirements is the first step to determining what capabilities you require and what type of deployment you should be investigating.

There are many different definitions of what type of security should be deployed when dealing with file transfers, but most vendors agree that simply sending files, regardless of protocol, is no longer enough. Ideas differ on perimeter security and Demilitarized Zone (DMZ) best practices. Some vendors offer IP session breaks and authentication/authorization in the trusted zone, while others offer hardened appliances with databases that allow virus scanning in the DMZ. Regardless of the deployment mechanism, it is vital to have DMZ-based proxies to cut down the number of open ports to the trusted zone of the enterprise.

Example of Best Practice Implementation



Best practices

No single solution is perfect for every enterprise, but there are many capabilities that should be examined while researching DMZ-based proxy solutions. Some solutions are optimized for high speed and low latency transfers while others are optimized for large file transfers. Whatever your file transfer requirements and use cases, consider these best practices:

Data protection

- Use secure sockets layer (SSL) and transport layer security (TLS) protocols.
- Do not store data in the Demilitarized Zone (DMZ).
- Understand industry and legal guidelines and requirements regarding cryptography and encryption and adhere to those guidelines and requirements.
- Use Hardware Security Modules (HSMs) for cryptographic key storage.

Perimeter security

- Use a DMZ-based proxy to terminate both IP and SSL sessions in the DMZ, thus blocking direct port access from the public Internet to the trusted zone.
- Minimize inbound and outbound firewall port access.
- Deploy a Data Loss Prevention (DLP) solution.
- Deploy in a multitier DMZ structure.
- Provide in-line virus scan or Internet Content Adaptation Protocol (ICAP).

Authentication

- Authenticate in the DMZ rather than the trusted zone.
- Use multifactor authentication.
- Provide role-based access.

Each of the aforementioned best practices are necessary pieces to providing complete perimeter security.

Vendors are actively developing new deployment methods, improving their breadth of supported protocols, and providing their customers with the ability to connect to trading partners in a safe and secure manner. Vendors are doing more to come up with the best solutions to secure the perimeter. But firewalls are no longer enough, and a virus scan on your machine is clearly not the end all for enterprise protection.

No single perimeter proxy solution can provide all of the capabilities of the best practices list that we provided. As the marketplace continues to demand more security, vendors are improving their offerings but there are still gaps. File security should be your first priority because this is where you are most vulnerable due to the nature of your connections with multiple vendors and with the public Internet. It is up to you to determine the weaknesses in your enterprise security plan to determine what solutions can best secure your perimeter security.

Section III
Software development security practices

In this section, we present processes and techniques for addressing security during software development. 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.

Email password—the keys to your personal online identity

How important is your email password?

For anyone using the web today, your email address is a crucial part of your online identity. This means your inbox is much more than a treasure trove of personal emails, photos, and information you wouldn't want shared with the world. It is a gateway into your online identity. When you sign up for a website, your email address is one critical piece of data, and your password is the other. If a malicious person gets their hands on both pieces, he or she can wreak havoc on an unsuspecting user. The vast majority of users on the web today simply don't realize this danger and fail to take simple steps to protect themselves. Furthermore, webmail and other online portals use dated techniques to permit password recovery which attackers have taken advantage of recently and on an on-going basis.

Once more into the breach

So how does your password make it out to the Internet for the world to see? It is a direct result of all of the security breaches we hear about in the news each day. It has become a sport now for attackers to steal as many user names and passwords from a website as they can, and post them publicly. In the past six months alone, millions of email addresses and passwords have made their way to public sites.

Once leaked, even passwords that are encrypted using a hashing function, can often be recovered into plain text, either through dictionary based brute force methods, or lookups in preexisting tables of common passwords and hash values.

Why does this matter?

Data from recent breaches has shown that a high number of users on the Internet reuse passwords across multiple websites. Thus, when a random website is compromised, the attackers often dump a list of all of the email addresses and passwords they can find. This is bad enough when the email address ends in Gmail.com, yahoo.com, or hotmail.com. What happens when that email address belongs to a .gov domain, or your own business? How comfortable are you knowing that if an end user's email and password were leaked, there is a real possibility that it is the same password they use for your corporate resources? You likely reuse passwords for different types of corporate and personal resources too. Having multiple passwords is an advisable approach, but it can still cause grief if the password is not sufficiently complex or if it is stored in a non-encrypted format.

What happens next?

Once your email address and password are publically posted, it is open for any determined individual to begin to try and log into your email account using the password listed. Many of the most popular sites do little to prevent such brute force attacks. Once someone finds a password that works, what they can do next depends on what is linked to that account. It may be as simple as reading all of your private emails and looking through your photos, or using your account to spam others. On the pricier side, they can end up gaining control of your online banking, shopping accounts, or credit cards. They can learn where you live, who you bank with, and what you buy online. There is more than enough information for someone to commit identify fraud.

Forgot your password? Click here to reset

Most user based websites have some kind of password recovery mechanism in place. In general, a common technique is to email you a link on which you can click to initiate a password change. If an attacker has access to your email account already, this is a huge security risk. Think about all of the services associated with that email address: eCommerce, financial services, and social networks. The list is long. Any one of these sites will gladly email you a link to change your password. Some sites have put in additional steps now to make changing your password more difficult, but most just require that you click on a link. Now that some malicious person has access to your account, they have the chance of costing you real money. Most of us have our credit card details already on file with say an eCommerce site, making it simple to purchase items. With some online services, an attacker could setup an additional bank to transfer funds towards. Sure, the service may email you a warning that says someone has added a new account, but what good is that if your email account is already compromised and someone can just delete that email?

"Don't use the same password on different sites"

We have all heard this advice before, that we should never reuse our passwords. Some people suggest a different password for every site using a password managing tool. Others say that you should use one password for secure sites like online banking, and a different password for the not so secure. These are great suggestions, but difficult to get people to act upon. While financial fraud is highly inconvenient for the end user, do not forget that many people are reusing personal passwords for enterprise systems. Think about the scale of loss as it may impact your business if intellectual property was stolen due merely to credential reuse. Of course, in this scenario, IBM X-Force would recommend a layered security approach to mitigate and/or minimize the potential damage.

Rules and regulations vs. the real world

No matter how strict corporate security policies are regarding passwords, users most often do the bare minimum needed to be compliant. It is common for people to change their password from ending with a 1, to a 2, to a 3 whenever they're forced to change their password. It is human nature: if we don't understand the reason behind something, we are less likely to follow through. Take the time to educate users on how easy it is to have their personal finances spoiled by some attacker and they may develop a much greater sense of caring about their passwords. The end result may be users who take their security a bit more seriously in the future, and that is good for everyone.

What is a secure password?

Ask a dozen different security professionals what constitutes a strong password, and you will get over a dozen different answers. The growing trend we support is using very long passwords, more commonly known as passphrases. A passphrase is simply a combination of words, or an entire sentence. Quite simply, the longer your password is, the more difficult it is to crack. Statistically, a 10-character password, no matter how many special characters are included, is not as secure as a 30-character password made up of random words. Using a passphrase is also much easier to remember than a convoluted mix of letters, numbers, and special characters. How much simpler is "MyPasswordIsNowSuperSecure" to remember when compared to "4K4$!lvabQ!"? A long password

you're sure to remember generally beats out any shorter password. The moment you have to write down your passwords on a sticky note so you can remember them is the moment you need to alter your approach to passwords. Make it a variation of your favorite song title from when you were a kid, combine some random words that make sense to you, or just come up with a random sentence. Spend another few minutes adding in some random characters (place a few symbols in the mix, capitalize a few letters, toss in a number here or there, and you've got a very secure password that is easy to remember.

An example

An example of this process comes from the lyrics "One Eyed One Horned Flying Purple People Eater." This is a rather long phrase, but easy to remember. You can shorten things up if you want to by changing "one" to "1" and get "1eyed1hornedflyingpurplepeopleeater." You can replace "purplepeopleeater" with "PPE" to get "1eyed1hornedflyingPPE." And, as this would be a scary site to see, you can put a "!" at the end for added effect (and security). Your final result, "1eyed1horendflyingPPE!" is 22 characters, and involves mixed uppercase and lowercase letters, numbers, and symbols.

Remembering your passwords

If you do want to use a different password for each online account (as you should) you need a way to keep track of them all, and I don't mean writing them down on a piece of paper. This is where a password management tool comes in handy. There are a variety of these available. Some of them keep your passwords encrypted in a local file that only your master password can unlock. Other services take this to the cloud where browser plug-ins can help make this job simple and easy. Whatever method you use, make sure that the tool supports a strong form of encryption (such as AES-256) and that your master password is actually a long 'passphrase.' Use the steps above to create this secure passphrase and you now have a secure method of generating random passwords for every website you log into.

Security questions

There is one other security risk when it comes to your email: security questions. Many websites, in an effort to enhance security, have actually weakened it with security questions, oftentimes a required field. Many of these questions have an answer any attacker could figure out in five minutes of searching. Your high school mascot, the city you were born in, and your birth date are all poor "security" questions. It is best that you answer these with false data if you wish to keep things secure. Rely on your secure password and a password manager to keep track of your passwords, not false security questions.

Two-factor authentication

While the steps above go a long way in the effort of keeping your email safe and secure, if you want more security, find an email provider that offers some form of two-factor authentication. Some services offer a smartphone app that generates a six digit code which is required to finalize the login process. Other offerings may send a SMS code to your phone. In either case, the end result is a second piece of information that exists, on your phone alone, to access your email. They also offer the ability to remember the computer you are on, so you don't have to enter your code every time you login. This means that you are prompted for this code whenever you log in from a new computer, one you've never used before to access your email. Such restrictions help ensure accessing your email from a home computer is safe and easy, while accessing it from an unknown—and potentially malicious computer—require the additional security code.

Putting it all together

We've discussed how valuable your inbox is, how easy it can be for attackers to gain access to it, and the havoc they can wreak once it has been accessed. In an ideal world, everyone would use a random password for every website they log into. There are a wide variety of tools to make this a reality for those willing to do so. For everyone else, take five minutes and come up with a long passphrase. Use this passphrase for your email and nothing else. Or, if you choose to use a password manager, use this password as the master password, and then use the password manager to generate a long and complex password just for your email. For added security, seek out an email provider that supports two-factor authentication. Embrace the idea of not knowing what the password to your email is; use a password manager to keep track of it.

Secure password hashing—when faster is not always better
When slower is better

In the fast-paced cycle of technology, we are led to believe that faster is always better. In many cases, this is true. However, there is one computational use case where being slow is not only preferable, but more secure. This use case is the way passwords are verified and stored in databases.

We continue to see headlines in which company X is breached and thousands (or millions) of user email addresses and passwords are posted publicly for all to see. Thankfully, often these leaked passwords are hashed rather than being saved as plain text. So, instead of seeing a bunch of actual passwords, each one appears as a long, encoded string.

This approach seems much more secure than storing passwords in plain text right?

Not quite.

A hash is a one-way encryption. Pass a string of text like a password to the hash function, and it passes back a new, mostly unique string, which is a mathematically transformed representation of the original text. One-way means that it is not possible to take the end hash and work backwards to the original text.

Web developers are advised to use best security practices such as hashing passwords before storing them in databases and ensuring that these hashes are properly "salted". We will explore password salting a bit later in this article. Hashing passwords in general adds a layer of security so that even the website owners can't easily see their user's passwords in plain text. This is critical for people who use the same password on multiple sites because it means that someone snooping the database can't take a password and email address and use it to gain access to other sites using the same password.

Let's consider an example of a fairly poor password, such as 12345.

If we calculate the hash using the PHP MD5 function (a popular and easy-to-use hashing tool for storing user passwords), we get this MD5 hash:

```
827ccb0eea8a706c4c34a16891f84e7b
```

This looks fairly complex and certainly does not in any way give away the original text. However, as a simple web search illustrates, our original password text is displayed directly in the results.

In the event of a security breach, if a website is using MD5 hashes, and has a user with this password, the text will be discovered in seconds with a simple web search.

This is a simplistic example. However actual hashes from recent public breaches have shown that people are still using passwords like this to secure access to web services.

As a website or service, asking users to use more complex passwords may help and, while a good practice, it is not going to be effective as a single solution. Even complex text and hashes can be quickly recovered; limited only by the speed of the hardware used to guess passwords in relation to the time it takes to run the hashing function.

Consider the options

As web developers, what can be done to help ensure password hashes are stored more securely?

Some possible strategies might be:

- Run the same hashing function multiple times.
- Make the password / source text more complex.
- Use a slower hashing function.

A hash of a hash

Running the encryption multiple times is one way to obfuscate the original password and make it more difficult to reverse.

Going back to the password 12345, we calculate the MD5 hash (827ccb0eea8a706c4c34a16891f84e7b) and then run the MD5 function again to get the hash of this string which is now

`1f32aa4c9a1d2ea010adcf2348166a04.`

This appears to add a new layer of security, but again a quick search leads us right back to 12345.

In theory we could repeat this multiple times instead of just one extra time, and while an effective strategy under the right circumstances, there are still other issues to consider.

The first has to do with the concept of collisions, which means that two different source strings create the same hash. While a possibility (both theoretically and proven with some hash functions), this is outside the scope of this discussion.

Additionally, calculating the hash of a hash string is potentially more mathematically limiting than calculating the hash of a text password. In cryptography, this is referred to as password entropy. When talking about password entropy, we are considering the length of the password as well as the variety of characters, numbers, and symbols that can be used. Therefore a strategy that relies on hashing a hash is limited to a fixed entropy no matter how many iterations are involved.

There are many great resources online which explain how to calculate password entropy.[24] The central idea is that the more bits of entropy in a given source string, the longer it takes to randomly guess every possible combination.

An MD5 hash consisting of 32 hexadecimal characters has 128 bits of entropy. Guessing every combination even with today's computing power would take a long time. However, the hash is always fixed at 128 bits of entropy whereas a source password or passphrase can be made to have higher entropy than a hash. If we assume that hardware will continue to speed up, reducing the time it takes to guess every combination, the ability to increase entropy over time is the best solution.

The next logical conclusion is to require that passwords are very long, ensuring high entropy.

24 Password entropy: http://pthree.org/2011/03/07/strong-passwords-need-entropy/

More complex passwords

A great deal of effort has gone into educating people on how to choose a secure password. See **"How important is your email password?"** for more information. Many companies and websites attempt to enforce a strong password policy. While this is a good practice, especially for preventing common password guessing, it is not a perfect solution.

From a software side, another recommended security practice is to add a salt value to a password before hashing it and storing in the database. A salt is just an additional element, such as a random string of text combined with the password before it is sent to the hashing function.

Adding the salt not only increases the entropy of the password (by making it longer and more random), but also limits the use of pre-calculated lookup databases called Rainbow Tables. Unfortunately there have been breaches in the past year where password salting was not used wholesale for user records.

A web search for a hash is itself a kind of Rainbow Table lookup (more specifically, indexes and links to sites which maintain large Rainbow Tables of common words and phrases). Creating Rainbow Tables is as easy as running the hashing algorithm for millions and millions of combinations of possible passwords and storing the result for later use. For low entropy passwords like six lowercase letters, it is possible to create a lookup table for every possible hash in minutes. Then recovering a hash is just a matter of checking if it exists in the lookup table.

Adding some random text (the salt) to each password, reduces the likelihood that there is an existing Rainbow Table with this value.

Using the previous example of password 12345, we can add a random salt string:

```
'12345'
+'GlpQclJDRqYGeHi5PeRbg0oMHFlhNnBa'
```

which results in an MD5 hash of

```
09f60edb0aa088d50d0482c7ba745059.
```

Search this hash in any freely available Rainbow Table lookup, and it won't likely be found.

Existing pre-calculated Rainbow Tables are unlikely to have a hash stored for this string. However, if a database is breached, and the salt value is stored within the hash or as a separate column, the ability to recover the password with the salt is still only limited by the speed of hardware related to the speed of the hashing function.

In a recent high profile breach this year, 6.5 million hashes were publically posted. These hashes were generated with the SHA-1 algorithm without any additional salt. In just a few weeks, researchers were able to recover 90% of these passwords.

The reason they were able to achieve such a high rate of recovery is based on several factors and illustrates just how fast today's hardware is at recovering passwords.

The first was that the source passwords either did not have very high entropy or were based on common words and phrases that can be easily guessed given a large pool of source words. Even using a longer multi-word passphrase of more than 20 characters was not sufficient when the phrase was a song title, lyric, famous quote or any other "known" phrase. All of these things can be added to the pool of guesses and, given enough time, can be recovered.

The researchers cited[25] that, had the company salted the passwords, it would have slowed things down considerably. In itself, that is not a solution because the largest contributing factor was that the hash function SHA-1 is very quick in comparison to other hashing functions. Using a free tool and a home server, the researchers were able to guess a staggering 15 billion SHA-1 combinations per second.

Imagine any dictionary word or simple common password would likely be recovered instantaneously. Even If the researchers were focused on a single password with a known salt, they are still able to try billions of combinations in a short time. Adding salt is a best practice, but given the speed of today's hardware, is not enough.

Go slowly

Since hardware continues to get faster and hugely parallel computational systems are inexpensive and well suited at guessing passwords, it seems that the next best solution is to slow down the hashing algorithm. If it takes one second to calculate 15 billion SHA-1 hashes, a different function should be an order of magnitude slower.

SHA-1 was not designed to hash passwords. Optimally, there should be hashing functions that are able to keep pace with the increase in computational speed and power, and adjust accordingly.

One technique has to do with the hash of the hash concept where the number of iterations —depending on the function—can scale up into the billions. Running the same function a high number of times would certainly slow down the time it takes to calculate, which also slows down the time it takes to guess combinations.

SHA512crypt is one such password-hashing function that can be configured to iterate thousands of times or more. For the researches recovering the SHA-1 passwords at 15 billion a second, using similar types of hardware were only able to guess 11,405 per second using a SHA512crypt function set for 5,000 iterations.

Password-Based Key Derivation Function 2 (PBKDF2) is another cryptographic function created specifically to address the issue of password recovery speed and can also be configured to run multiple iterations.

Bcrypt is a cryptographic hash function created specifically for passwords and is based on the Blowfish cipher. Bcrypt uses internal salts to randomize the resulting hash making it more difficult to create rainbow tables. Bcrypt also provides configuration support for multiple iterations, although the function itself is slower, meaning less iterations are needed to slow things down in comparison to a function like SHA512crypt.

Scrypt is another dedicated key-derivation function that can be used for password hashing. One of the differentiating advantages of using Scrypt is that each calculation is designed to use a large amount of memory which can make it more resource intensive to do parallel password guessing using GPUs or FPGAs (see sidebar on page 96).

Given these existing "slow" password hashing options, why don't more web developers adopt these in practice?

25 LinkedIn password recovery: http://securitynirvana.blogspot.co.uk/2012/06/final-word-on-linkedin-leak.html

There are several possible reasons. The first is that functions like MD5 and SHA are well documented and easy to use in server-side languages such as PHP and Java. For many years, these seemed like viable solutions to password security, and worked easily and efficiently. Libraries that provide slow hashing functions were not as smoothly implemented or readily available. Today there are many tools available and recommended best practices for implementation.

Another reason likely comes down to education. As the hardware needed to crack billions of passwords a second becomes commonplace, more and more developers are welcoming the idea that something better is required.

Secure web applications are a first line of defense. Databases full of password hashes should not be dumped in the first place. However, as with any security best practice, multiple layers of defense are always recommended. Using a slower hashing algorithm—one designed for secure password storage—is a highly effective way to help ensure the integrity of customer data.

Faster, cheaper and powerfully parallel

A few years ago, multi-core CPUs (Central Processing Units) made it possible to guess password hashes in faster and faster batches per second.

At the same time, the demands of 3D games spawned the need for faster and more powerful dedicated graphics processing cards.

In recent years, graphic card manufacturers have released high level API's which allow programmers to more easily write applications to run in parallel directly on the GPU (Graphical Processing Unit). This is great news for science and medical applications, audio and video processing, and other heavy mathematical uses which can leverage the power of this multi-core platform for the greater good. However, for cryptographic algorithms that are only as strong as the speed at which they can be brute forced, this presents a problem.

Whereas a desktop CPU today may have around 2-16 cores, a consumer GPU card might have anywhere from a few hundred to a few thousand cores. Considering that each core is capable of handling tasks in parallel, a task repeated over and over, like running a hash function for every possible letter and number combination of a password, goes much faster.

It turns out that rendering frames of a first-person-shooter game is not much different from the math required to do advanced cryptographic calculations. Where the CPU is a kind of jack of all trades, responsible for handling a variety of different tasks and computations, the GPU excels at crunching huge batches of numbers repeatedly in quick succession.

Password guessing has even gone "in the cloud." By using a cloud service provider, it is quite inexpensive to rent an array of GPUs to crunch a task for a few hundred dollars an hour. These types of computations scale very well in parallel.

Using the GPU to guess password hashes is still a software operation and is thus limited by the speed of software running on a disk on an operating system.

Another tool in the emerging field of "password recovery systems," is hardware-based Field Programmable Gate Arrays (FPGA). FPGAs come in the form of an appliance that contains several cards, capable of executing a task like calculating a password hash at blinding speeds. At the moment, these are more expensive than using CPUs or GPUs but do provide significant speed increases. According to one FPGA vendor,[26] the appliance is able to guess 1,756,800 WPA-PSK wireless passwords per second vs. 103,800/sec guesses on an AMD GPU vs. 30,000 /sec on an Nvidia GPU vs. 4,000/sec on an Intel I7 CPU.

26 http://vr-zone.com/articles/pico-fpga-hardware-beats-gpgpu-in-password-cracking/16705.html#

HASHES to ASHES

Don't get burned by leaked passwords

How
Do They Do It?

Rainbow tables pre-calculate password hashes and store them efficiently for future look-up. Over time, they can include a huge number of password combinations.

Dictionary attacks guess passwords using a very large file of known words, phrases, quotes, and and other rules used in password creation like substituting a 3 for the letter E or capitalizing first letter.

Brute force tries all possible letters, numbers and symbols. Using modern hardware and a fast hash function, every combinations of a 6 character password can be guessed in seconds.

What
Can you do?

As a User

- Don't reuse passwords on multiple sites
- Don't use established common password tricks
- Don't use dictionary words or known phrases
- Use two-factor authentication where available
- Use a password manager

As a Web Developer

- Use slow hash function made for passwords
- Audit code for XSS and SQLi vulnerabilities
- Use IPS, Web Application Firewall or similar

Once the hashes are leaked it is possible to rapidly recover the password text through several methods using freely available tools.

3D Graphic cards (GPU) can run hash functions very quickly in parallel. In some cases guessing billions of passwords a second. Specialized hardware like FPGA's and cloud services have dramatically increased cracking speeds.

MD5 or SHA-1
BILLIONS OF GUESSES PER SECOND

SHA512CRYPT
A FEW THOUSAND GUESSES PER SECOND

BCRYPT or SCRYPT
A FEW THOUSAND GUESSES PER SECOND

Slow it Down

By design, some hash functions can be calculated quickly. These are not good for storing passwords as attackers can guess many combinations per second.

Better to use a slow hash function which vastly reduces the number of guesses per second, making the recovery process much harder.

After passwords are recovered, attackers will use the leaked email address and plain text passwords to attempt access to webmail, social networks and other common sites.

Users who resuse passwords are often unaware of how a breach on one site can allow access to several others.

Passwords are leaked when an attacker gains access to a database through SQL Injection, XSS, or another vulnerability.

The passwords are often stored as a hash, an encrypted representation of the text.

In a recent study*

59%

of users were found to be using the same password on multiple sites, including their webmail accounts.

*http://www.troyhunt.com/2012/07/what-do-sony-and-yahoo-have-in-common.html

Section IV
Emerging trends in security

This section looks at fast-developing technology that challenges enterprises considering whether it is time to make investments in these future areas. We explain where threats and exploits are being used in these early technology adoptions and how enterprises can stay focused on securing them.

Influences of initial bring your own device (BYOD) in most enterprises

Mobile enablement in most enterprises continues to be a challenge to security. One game-changing transformation is the legitimization of bring your own device (BYOD) programs. Many enterprises have not acknowledged or supported personally owned traditional computing devices previously, so the implementation of a BYOD program for mobile devices such as smartphones and tablets is really a broader transformation that should include formulation of policy and governance to support the use of these devices. This is in addition to required security controls and corresponding technologies.

The importance of appropriate BYOD policies, formulated in a cross-discipline manner that includes input and guidance from human resource, legal, and perhaps input from the employee population is fundamental. For those enterprises with existing BYOD programs that already support traditional computing devices, it may be appropriate to review the existing policy to determine if changes are warranted in expansion to mobile devices (since mobile may drive a significantly higher use of personally owned devices compared to personally owned traditional computing devices).

State of security

The state of mobile device security is in flux. While there are reports of exotic mobile malware, such as TigerBot/Android.Bmaster on Android, and Zeus/ZITMO on multiple mobile platforms, most smartphone users are still the most at risk of premium SMS scams and the like. These scams work by sending SMS messages to premium phone numbers in a variety of different countries automatically from installed applications. There are multiple scam infection approaches for this: 1) an application that looks legitimate in an App Store but only has malicious intent, 2) an application that is a clone of a real application with a different name and some malicious code, 3) a real application that has been wrapped by malicious code and typically presented in an alternative App Store. This brings up an interesting side-point: primary App Stores have strong brand incentives and known security initiatives to identify rogue apps being submitted whereas alternate App Stores may not. While the freedom of choice is beneficial to the ecosystem, it adds a lot of complexity to the security paradigm and subsequently is less beneficial to enterprises and a significant cross-section of end users that will not participate in alternative App Store environments for any of a number of reasons. Adding to the complexity of BYOD and applying best practices to one's own mobile device, is that several popular applications require extensive permissions to the extent that even experienced users may become less vigilant and numbed to permissions for new applications that may be risky or unnecessary.

Why SMS? In actuality, SMS/text is important to mobile malware writers whether for the purposes of a direct SMS scam, or something indirect like Zeus. Text messages can be utilized to direct command-and-control of a botnet (so far on the mobile platform in a centralized fashion), they are used by some banks around the world for two-factor authentication on bank/wire transfers, they can be sent to premium number around the world where bad guys (and dishonest organizations) can rake in the money directly from your phone company.[27] Since the mobile carrier automatically handles the billing, this is unique in the endpoint world as it directly links a device to some automatic level of financial risk or access—depending on one's perspective. And finally, text messages have become so ubiquitous in society that even unsuppressed messages to and from malware may go unnoticed.

The point on two-factor authentication via SMS text is interesting as it exposes what seemed at its inception to be a great approach to security but while it surely has reduced the risk to financial organizations, the number of mobile operating systems that Zeus mobile (ZITMO) supports indicates that as time goes by it can become increasingly ineffective without adding some complexity to the transaction.

Code bombs? There are a variety of for-hire mobile app programmers and outsourcing development firms. While it is easy to test general application quality upon delivery, few will audit the code they will present to an App Store for surreptitious code. While we haven't seen a major brand affected by trojanized software development, there isn't much preventing this from happening at some point. Organizations that outsource their mobile application development should be especially mindful if their applications handle sensitive personal or financial data.

27 http://www.guardian.co.uk/technology/2012/may/25/android-users-angry-birds-malware

Regarding targeted attacks, while there have been many anecdotal stories IBM X-Force has heard from reputable sources, we believe that the cost to deliver targeted exploits to mobile users is high enough that only potential victims that are known to have useful data in a consumable form will be at risk. In other words, targeted attacks on mobile likely exist on all major mobile operating systems, but one's likelihood of being targeted is overall extremely low.

To reiterate, IBM X-Force sees the mobile security threat landscape as in flux. The software security models of the different mobile platforms such as Android and iOS are different than the typical endpoint and have some differences between each other that we shall explore another time. While there are some exotic attacks of some scale, the primary mobile security risks are with fake or rogue applications that cost the end user or business money through premium SMS messages. As criminals find ways to monetize these at scale, we may see more mobile bots like Android(dot)Bmaster. We shall now explore the scope of the BYOD topic and best known practices.

Making BYOD work

To make BYOD work within your company, a thorough and clear policy should be in place before the first employee-owned device is added to the company's infrastructure. This policy should cover all aspects of the relationship between the company and the employee's device, as well as buy-in from all parties. Suggested areas to be covered in such a policy include:

- Identification and authentication
- Access authorization
- Information protection
- Service integrity
- Assurance
- Incident response

Most companies already have policies in place that cover these areas for the protection of company-owned equipment. These policies should be applied to devices with employee ownership used in a BYOD model. In nearly all cases, the required controls should ensure the same level of security expected to protect data.

Identification and authentication

Control requirements for the data classifications being considered for enablement in a BYOD program should remain aligned with existing authentication requirements. In the mobile context, this means helping ensure properly managed and enforced passwords that meet required complexity and syntax requirements. Asset databases should be extended to help ensure identification of the inventory of personally owned assets in use, along with properly managing the licensing of any enterprise-supplied software as part of device lifecycle management. A clearly defined software licensing policy should be in place to help ensure that employees use only properly licensed software when using their devices in the enterprise context.

Access authorization

Enterprises approaching BYOD for the first time likely have existing remote and application access programs and it is suggested that the use of the existing infrastructure, technologies, and processes be extended for BYOD devices. First, this helps ensure that access occurs within existing control requirements. Data associated with this access are unlikely to change in terms of classification and corresponding controls, so this approach helps consistency. Second, this approach is likely more cost effective than rolling out ad hoc, device-specific BYOD access, especially because this is likely not only remote access gateways but also application access controls.

The obvious exception to this recommendation is those enterprises that have elected to provide completely unique BYOD access programs. In some industries, completely virtualizing access by BYOD devices also drive unique access methods that should not be overlooked.

Information protection

The security of enterprise information and data on employee-owned devices is of utmost importance to the enterprise. Typically, information protection requirements are well defined and aligned to specific data classifications. These should be applied consistently in a BYOD program. One option an enterprise may require in their BYOD policy is data encryption. This option should be aligned with existing requirements but clearly should be understood and defined appropriate to the mobile operating systems being used.

Many devices in use today, offer the option of encrypting all the storage available to the device and requiring the user to enter a pass phrase at boot time before the device can be accessed.

If the device is lost or stolen, encrypted storage offers a certain level of protection. However, with the ever increasing power of Graphical Processing Units (GPUs), it is conceivable that the encryption could be broken, given enough time. An emerging area of concern is the upcoming availability of **GPU processing in the cloud**—potentially at a massive cost reduction and boon for attackers. One suggestion for any good policy is a “wipe clause.” In the event of a missing device, this allows the company to send a wipe command that deletes all data on the device once the device is accessible to a network.

Operating system and application integrity

Like company-owned servers and workstations, devices that qualify for BYOD status have operating systems and applications. In the case of smart phones and tablets, the level of maturity of the software is far less than that of traditional servers and workstations. This makes these devices prime targets for attacks. A good BYOD policy should take this into account and require the same (or a higher) level of patch requirements as traditional devices. At present, this may mean clear identification of properly updated version of firmware and using technology to help ensure only devices of appropriate versions are allowed to access

enterprise information. Device fragmentation, most prolific within the Android device community creates additional challenges such as older devices may not receive firmware updates even when the device is relatively young from a corporate refresh policy.

The devices should also be required to run an anti-virus application that has been approved by the company. This offers a level of protection from malware and malicious websites.

Smart phones and tablets offer less than full access to the device and must be "rooted" (Android) or "jail broken" (Apple iOS) to gain a higher level of access. The higher the level of access a user has, the greater the risk becomes, should the device be attacked. Since the practices of jail breaking or rooting essentially circumvent security controls—such as application sandboxing—within the mobile operating system, enterprises should ensure that such devices are not used within BYOD programs.

Companies should also restrict the sites where applications can be downloaded or purchased to the device-specific vendor sites. The vendor sites typically provide some level of quality control over the software they distribute.

Assurance

As in any existing enterprise security program, assurance that required controls are implemented and monitored is a fundamental element. This same level of assurance should be extended to include all devices with access to enterprise information in a BYOD program. Since these devices are employee owned, it is important that the monitored elements are clearly spelled out and understood by employees as they consider voluntary inclusion of their device.

Incident response

While a well-defined incident response process may seem obvious, it is an important and required part of any BYOD program. Since mobile devices, particularly smartphones, get lost and stolen far more often than traditional computing devices, educating employees in how to report a lost or stolen device along with an appropriate process to remotely wipe the device can be vital. In the ideal security program, this is integrated into the existing incident response process to determine the degree of loss, manage potential actions to mitigate, and to help ensure that exposed information is identified.

BYOD program definition and review

A BYOD policy is a voluntary contract between company and employee. As a contract, it should pass certain criteria before it can be presented to the employee.

Naturally, a company's legal department has to sign off on the policy. Human resources may need to be involved to approve certain aspects of the policy. The policy also has to abide by local country laws if the policy is to be implemented worldwide (or properly developed for adherence to local regulations).

Extended user education is also suggested. For a BYOD policy to work well, employees should understand, accept, and abide by all aspects of the policy. It may be necessary to go into more detail as to "why" certain policy elements were put into place. A well-informed employee is less apt to infringe on the policy. Similarly, an overly restrictive policy can lead to policy infringement or worse; the disabling and/or removal of security and access control systems from the device.

With careful planning, both the company and the employee can reap the benefits of a successful BYOD implementation.

Best practices in mobile security

While it would be wonderful to reference well-established best security practices for mobile devices (both tablets and smartphones), this formulation and maturity is still developing. While there have been some published security guidelines specific to governments, there is still varied practice in terms of actual control requirements required in mobile programs for enterprises.

As discussed in previous editions of the **IBM X-Force Trend and Risk Report**, mobile security control programs should be based upon existing data protection and control requirements that correspond to the data and information enabled on mobile devices. While this approach sounds straightforward and perhaps simple, in practice, what we've witnessed working with hundreds of customers varies significantly. Much of this variation is being driven by device ownership. This hasn't existed in many enterprise computing programs and, as a result, we see a segment of enterprises not necessarily reworking existing control requirements, but rather forging new ones for enterprise data on personally owned devices. Given the maturity of security control technology for mobile, we should not be surprised by this fractured approach but this may become a tactical issue as mobile operating systems continue to mature and increase the controls possible via their APIs.

We have observed a definite trend toward aligning access credential strength with existing control requirements. This may occur using certificate-based approaches; controlling device access to enterprise-managed devices via a numeric PIN are quickly diminishing. Many enterprises also recognize the need for malware prevention and/or some form of compromise detection.

We can summarize the acceptance of a degree of consistency in controls as progress toward best practices. But there is significant ground to be covered before best mobile security practices match those in other areas of enterprise computing.

State of mobile security technologies

Security control technologies continue to mature at a rapid pace. Platform vendors have continued to add controls that are accessible to all product vendors via their APIs. Enterprises are being given more and deeper controls with each revision; sometimes not as quickly as desired, but progress is being made nonetheless. Access to these added capabilities via API are fundamental to their inclusion in mobile device management (MDM) solutions in the marketplace. The marketplace in the MDM space has continued to mature, reducing the number of participants as many of the major security vendors acquire MDM start-ups to add to their portfolio. It is commonly believed that this solution market will become a commodity in coming years, causing further maturation of the marketplace. We expect this to occur as it does in any emerging technology that develops and becomes part of what would be considered mainstream and hence supported by all major vendors.

We have observed a shift in emerging mobile technologies as these changes to MDM solutions have occurred. The number of vendors offering "separation" or "isolation" technologies is increasing. These solutions focus on allowing enterprises to separate their applications and associated data from the existing applications and data owned by employees in BYOD scenarios. This approach sounds like a good balance of control in enterprise mobile programs that are heavily leveraged by personally owned devices. It also represents some compromises.

Many of these solutions are built on top of the major operating systems, commonly both iOS and Android, though many solutions exist for only one or other platform at present with roadmaps to address the other. They come with a set of limitations that may diminish their value, depending on enterprise mobile enablement goals. The two primary limitations common with such solutions are the loss of native functionality (because they may replace platform clients for things like mail, calendars, and contacts) and the absence of the ability to easily apply this separation to any and all applications that could run on the device. The inability to address this separation outside of a limited scope has often required enterprises to recompile applications for use within the separation solution. In some cases, where the enterprise has developed their own application, this is possible but often source code is not available to the enterprise, so this approach becomes a limitation.

As "separation" solutions have sprung up to support iOS and Android, it should be noted that this function is already part of the operating system in the current Blackberry release (as "Blackberry Balance technology"). As technology built into the operating system, it addresses the limitations we've seen in third-party solutions on iOS and Android and once again supports the need for separation technology to occur at the operating system level, fully integrated into how it works. Since Research In Motion has consistently led mobile platform vendors in the introduction of security controls needed to address enterprise security requirements, the other vendors may catch up and begin including this capability within their operating systems. We should at least consider that an investment in separation technologies is viewed as a tactical investment. Strategically, vendors in the mobile platform marketplace may include operating systems that offer this balance so that their devices are easily embraced by both consumers and enterprises.

Until inclusion of this separation capability occurs within popular mobile operating systems, expect much debate about this approach since, even with the use of these technologies, most security experts point to the need to trust the device these applications run on as a fundamental requirement to trusting solution security.

Approach trends by industry

While establishing best security practices for mobile security has remained very much a work in progress, we have started to observe some trends that correspond to industry segments. We should note that this is still far from the kind of trends and approaches we see in other computing segments due to the lack of best practice maturity, specific mobile controls, and so on. The use of separation or virtualization technology has identified some distinctions in certain industry segments. We tend to see enterprises in the healthcare, financial, banking, and government industries opting to use some form of virtualization or separation approach. In some observed cases, this is supported by a trend of not allowing sensitive data on personally owned devices at all. In that case, it has meant the use of either virtualization (for traditional computing devices like laptops) or application virtualization (for mobile devices). While virtualization approaches can prevent the presence of enterprise data, we do not recommend use of this technology approach because it relies on the presence of a trusted host to accommodate it and this usually means some level of device management to establish even a fundamental level of trust. To achieve a trusted host, virtualization approaches should be used with some form of device management to ensure device integrity.

Outside of those industries, we have seen the adoption of MDM solutions, using a more traditional security controls-based approach. While there is debate about things such as the strength of device passwords, there isn't usually debate around the need for the use of a password (just the specific password length, make-up, and reuse). In most of these cases, the whole device is being managed and participation in most programs tends to be voluntary. While BYOD is often discussed as a replacement for corporate supplied devices (due to cost savings), in practice, few enterprises have migrated to a completely employee-supplied, involuntary program for multiple reasons. BYOD is largely being offered and supported for those employees who do not qualify or require the constant use of mobile enablement to perform their work, although they may benefit from occasional access to improve efficiency and improve work-life balance.

Mobile platform vulnerability management

While we continue to see significant change in areas like security control technologies, devices, and corresponding features, the one constant we have seen in the mobile security landscape is the compromise of nearly every mobile operating system at every released version. In fact, often new release versions are jail broken or rooted within days or even hours of their release. This is a consistent statement across nearly all mobile operating systems. It is particularly unfortunate for a couple of reasons. Some mobile operating systems were designed with strengthened security models to begin with (application sandboxing for example) so the nature of how easily and quickly they are compromised undermines the addition of things such as sandboxing. Second, many security vulnerabilities remain

unpatched for weeks and even months on most mobile operating systems today. This second item is the problem that should most concern enterprises strategically, especially in previously discussed cases of device fragmentation and their support.

Rapid application of patches to close discovered vulnerabilities is a fundamental practice used to help ensure the integrity of enterprise computing devices. It is one of those foundational requirements that we know is central to a sound security program. In fact, lack of sound practices to patching has been a primary reason for some of the largest, most damaging security problems the Internet has ever experienced. In more specific, less pervasive issues, unpatched vulnerabilities provide a fundamental attack surface in many of the advanced threats we see today. We should expect to see advanced threats migrate to mobile operating systems as they become primary computing devices for many.

It is only a matter of time before patching of mobile devices becomes a primary requirement. Today, for most mobile operating systems, this is not even possible; enterprises have no ability to patch devices and, in practice, the operating system infrastructures are doing a poor job of it. Unfortunately, this problem likely leaves the enterprise with few options. Thus, enterprises may control this concern by locking out devices it considers too significant a risk because of the presence of unpatched vulnerabilities. Unfortunately, the loser in this approach is employees who are stuck with vulnerable devices that the vendor, OEM, or carrier is not ready or willing to patch.

Many mobile operating platforms do not even have the notion of a patch, relying instead on firmware upgrades that deliver a whole new operating system image to devices. This is compounded in some ecosystems by having multiple layers of firmware control between the platform vendor, hardware OEM, and carrier, typically resulting in many months or longer in devices getting needed upgrades. Commonly, the model is one of obsolescence. Rather than update devices to the current firmware level, carriers and OEMs try to sell replacement devices. While an economical solution for the manufacturer to devise fragmentation relative to architecture and implementation, obsolescence is not desirable from a corporate perspective and IBM X-Force predicts many organizations going through such headaches as the mobile ecosystem evolves.

As mobile devices become a primary computing device for many—both in enterprise as well as the Internet at large—we may find that patching of vulnerable devices becomes our primary security concern since this area has had the least progress made in the past year or so.

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

Produced in the United States of America
September 2012

Please Recycle

WGL03014-USEN-02