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

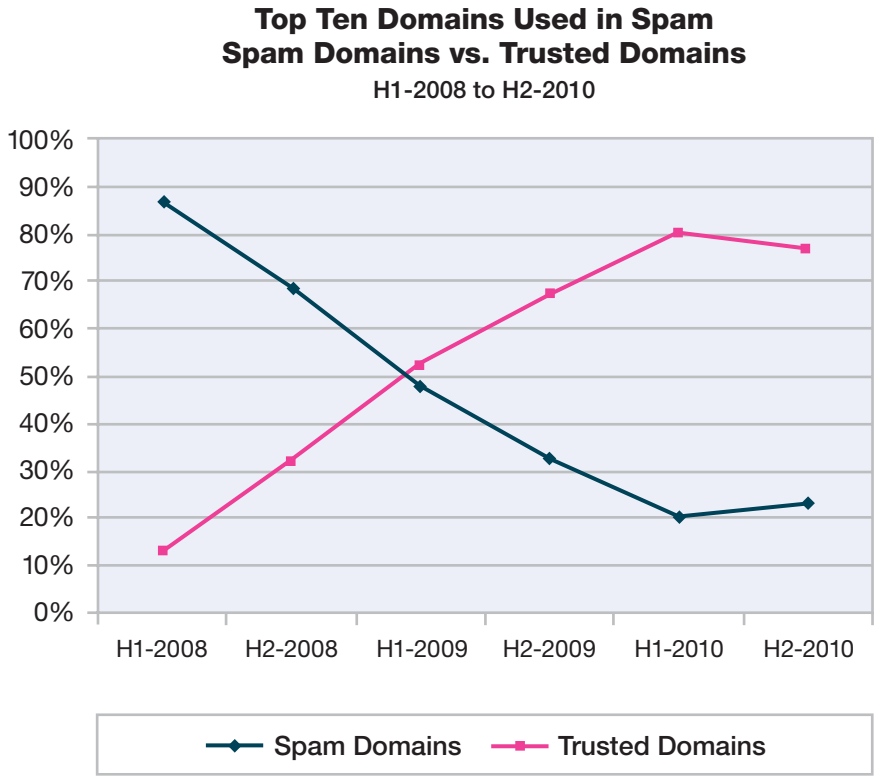


Figure 31: Top Ten Domains Used in Spam: Spam Domains vs. Trusted Domains – H1-2008 to H2-2010

Common top-level domains in URL spam

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

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

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

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

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

Internationalized country code top-level domains

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

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

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

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

Top 5 TLDs used to host Spam Content

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

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

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

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

Spam—country of origin

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

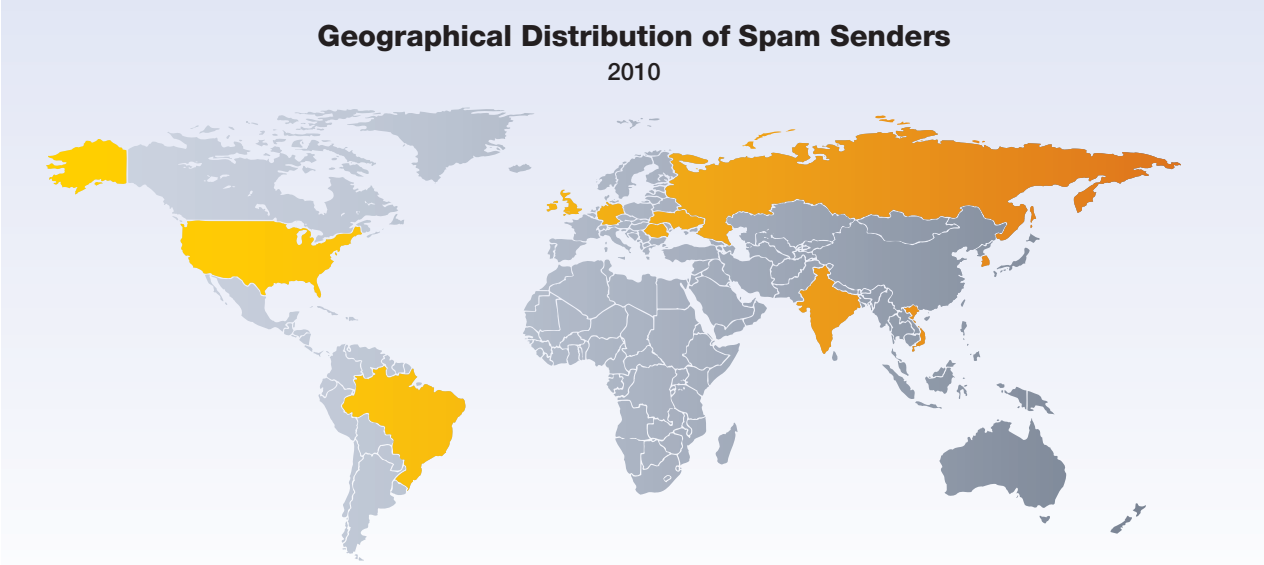


Figure 32: Geographical Distribution of Spam Senders – 2010

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

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

Table 6: Geographical Distribution of Spam Senders – 2010

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

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

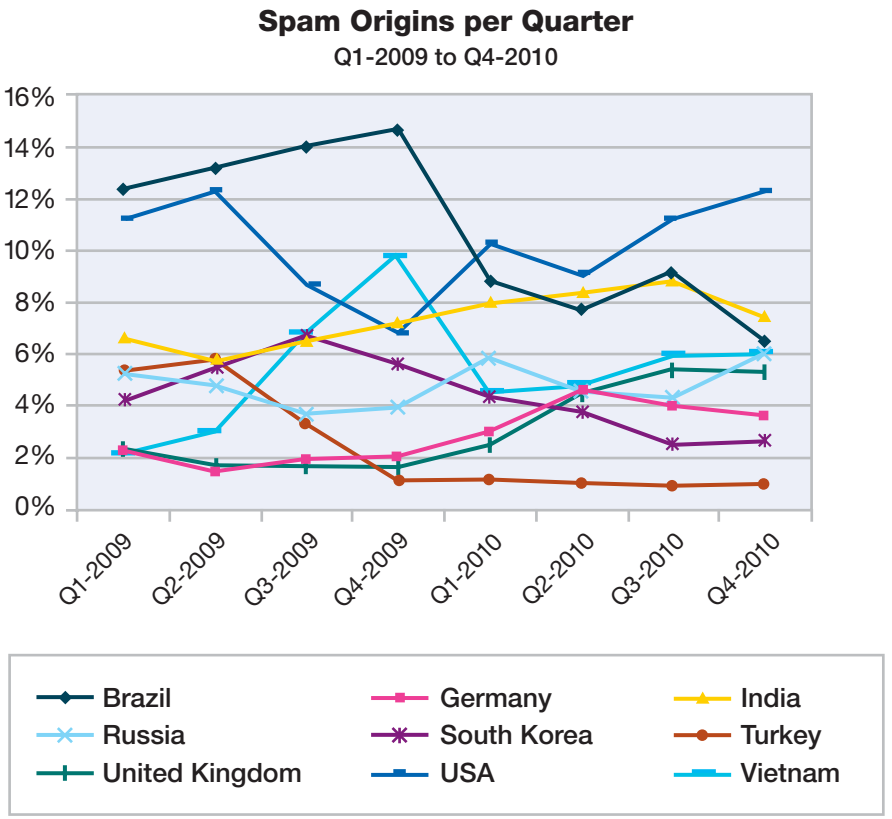


Figure 33: Spam Origins per Quarter – Q1-2009 to Q4-2010

Spam—country of origin trends

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

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

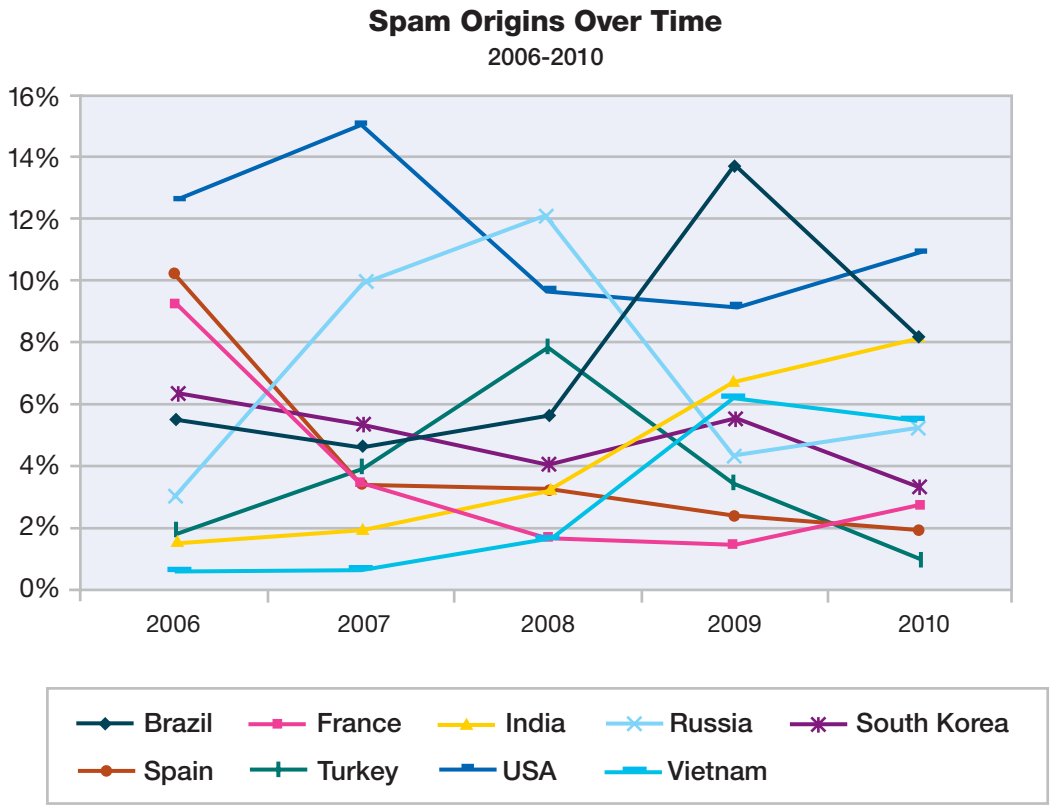


Figure 34: Spam Origins Over Time – 2006-2010

Spam URLs—country of origin trends

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

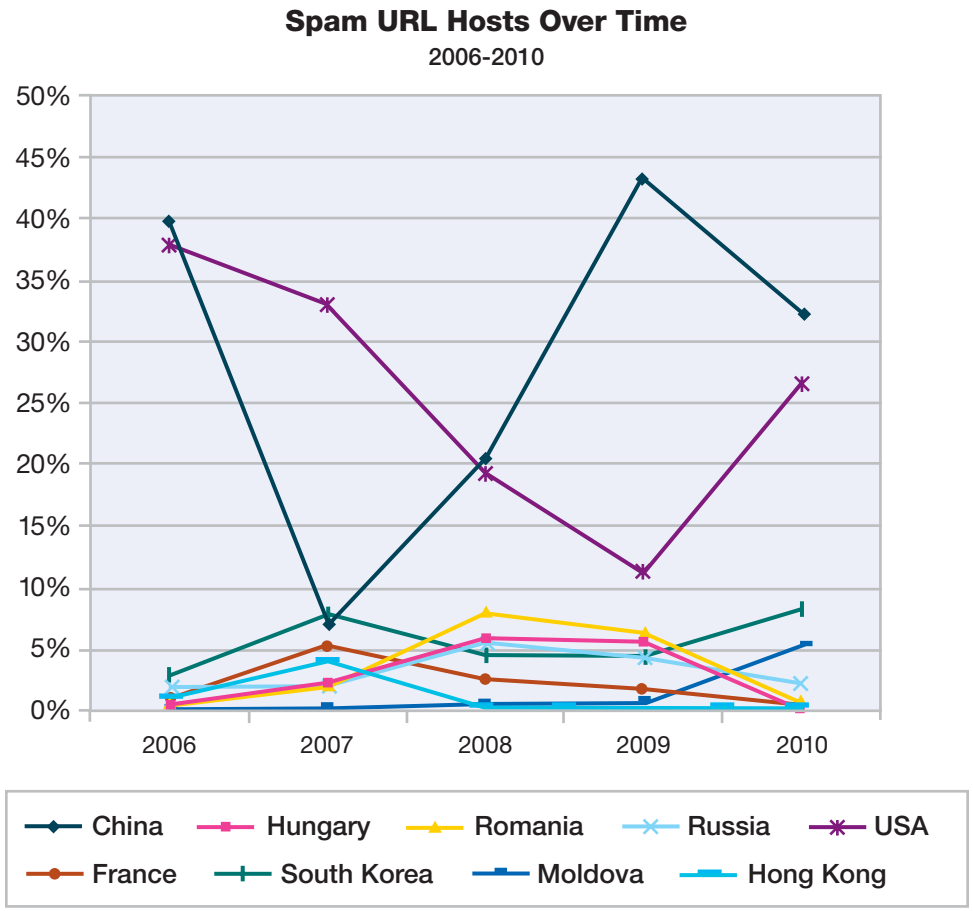


Figure 35: Spam URL Hosts Over Time – 2006-2010

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

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

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

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

Table 7: most popular spam subject lines 2010

Phishing

This section covers the following topics:

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

Phishing volume

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

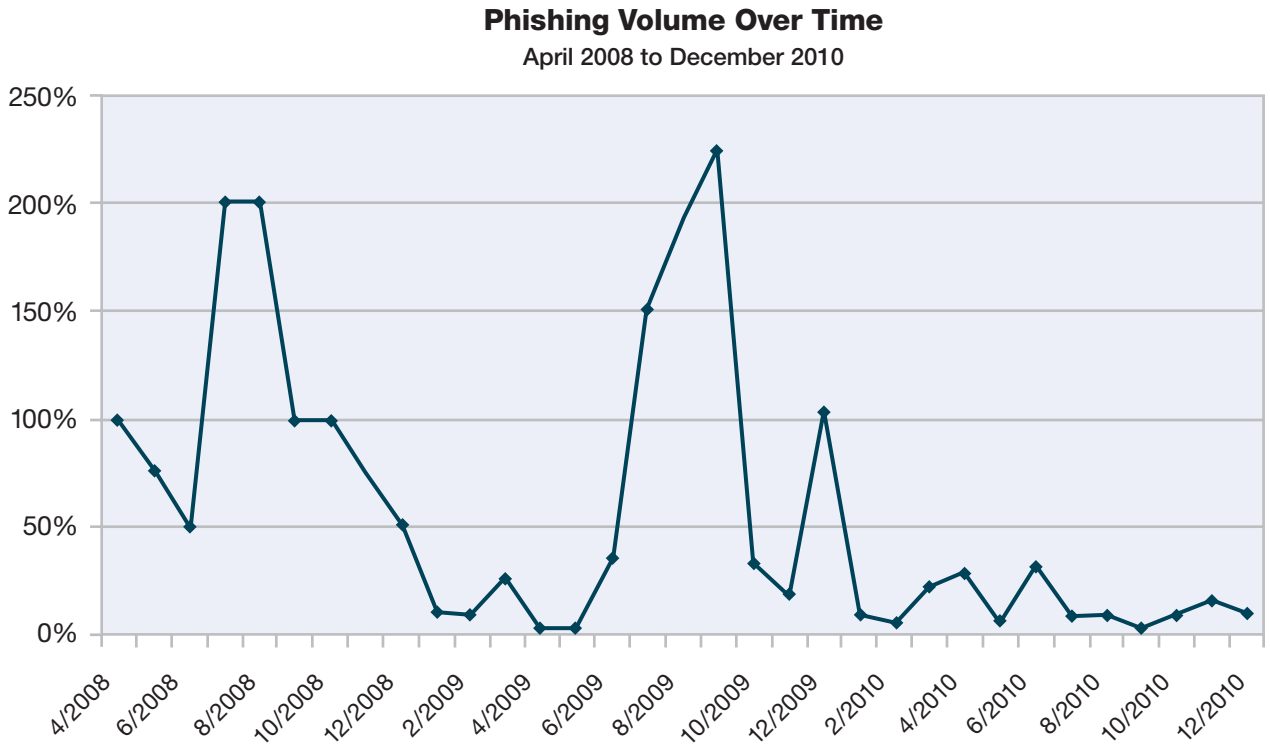


Figure 36: Phishing Volume Over Time – April 2008 to December 2010

Are phishers becoming skimmers?

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

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

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

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

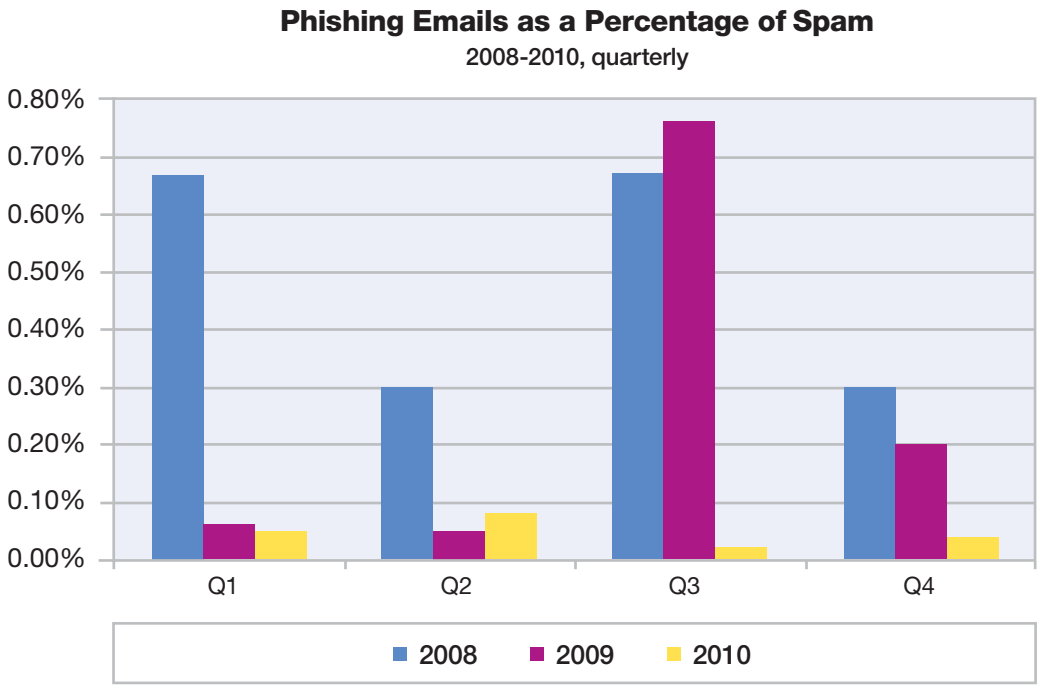


Figure 37: Phishing Emails as a Percentage of Spam – 2008-2010, quarterly

ATM skimming

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

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

Phishing—country of origin

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

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

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

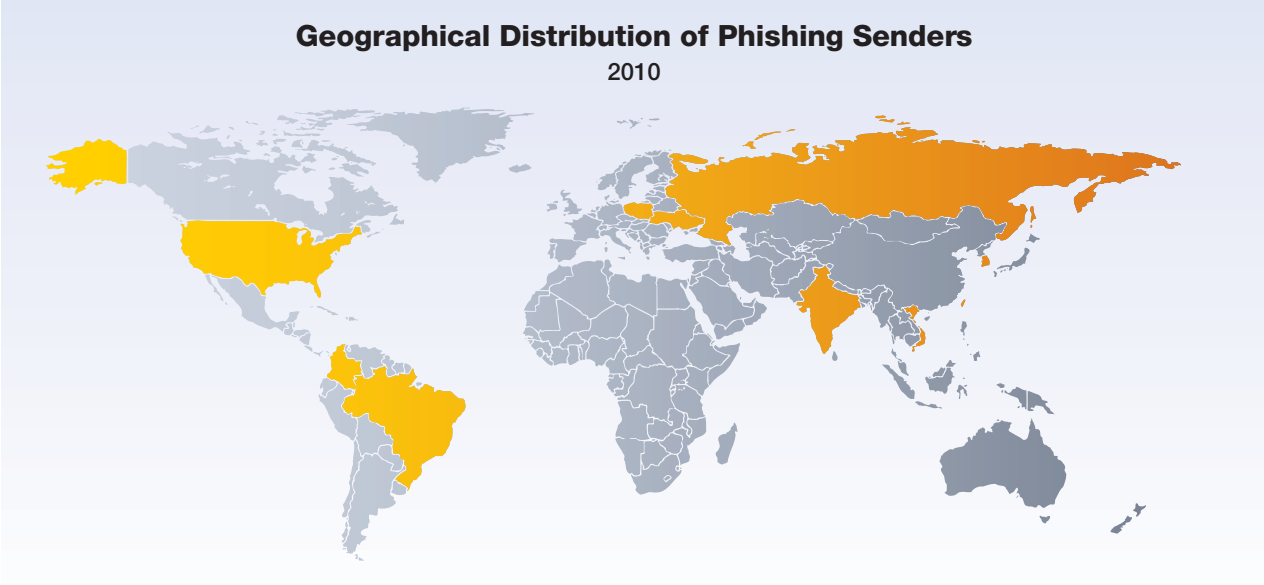


Figure 38: Geographical Distribution of Phishing Senders – 2010

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

Table 8: Geographical Distribution of Phishing Senders – 2010

Phishing—country of origin trends

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

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

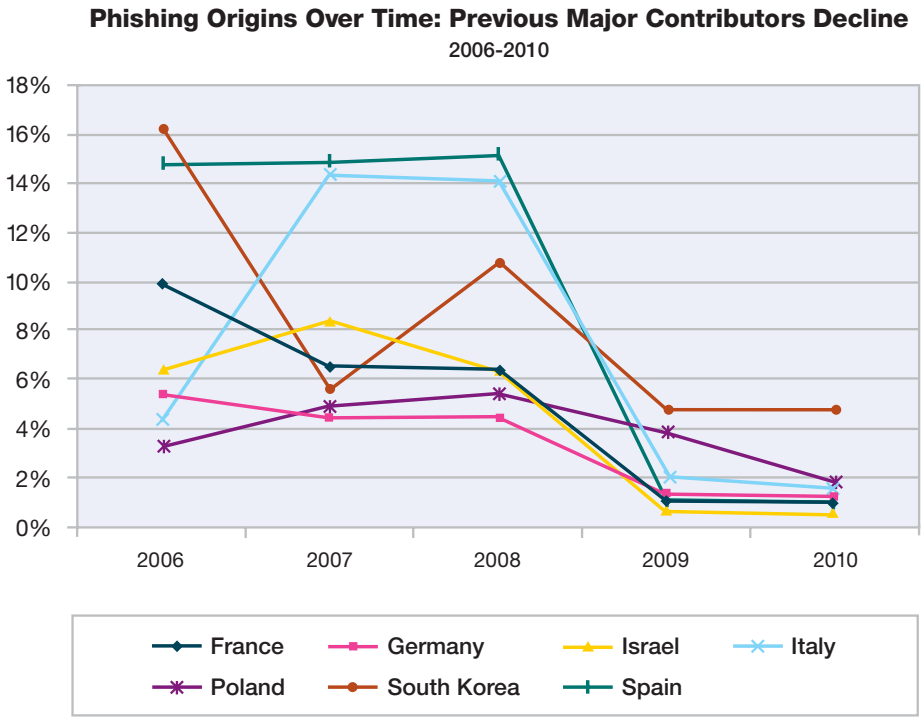


Figure 39: Phishing Origins Over Time: Previous Major Contributors Decline – 2006-2010

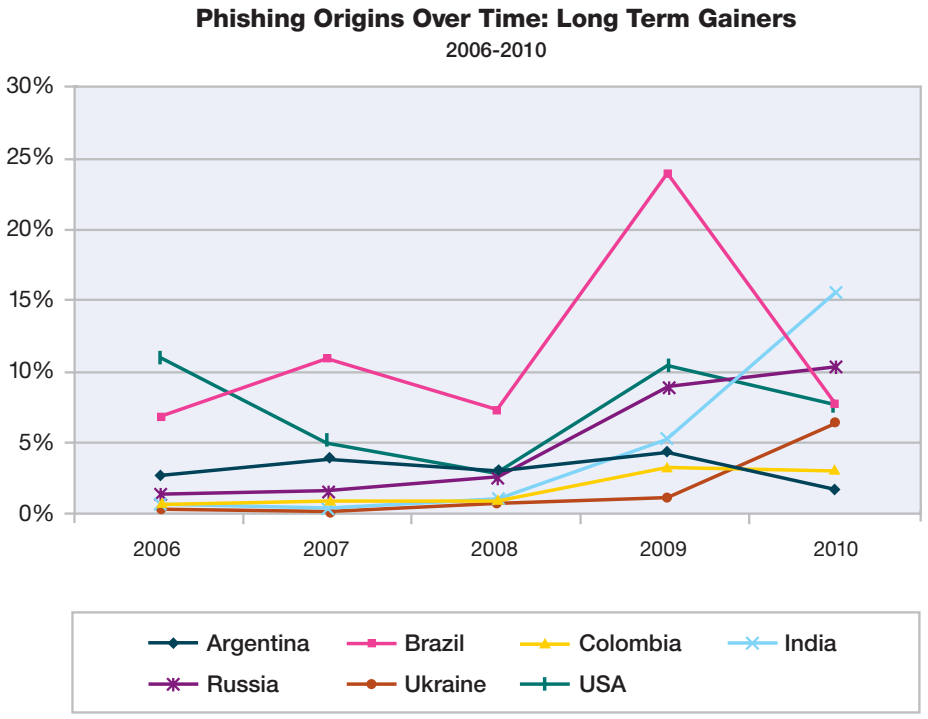


Figure 40: Phishing Origins Over Time: Long Term Gainers – 2006-2010

Phishing URLs—country of origin

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

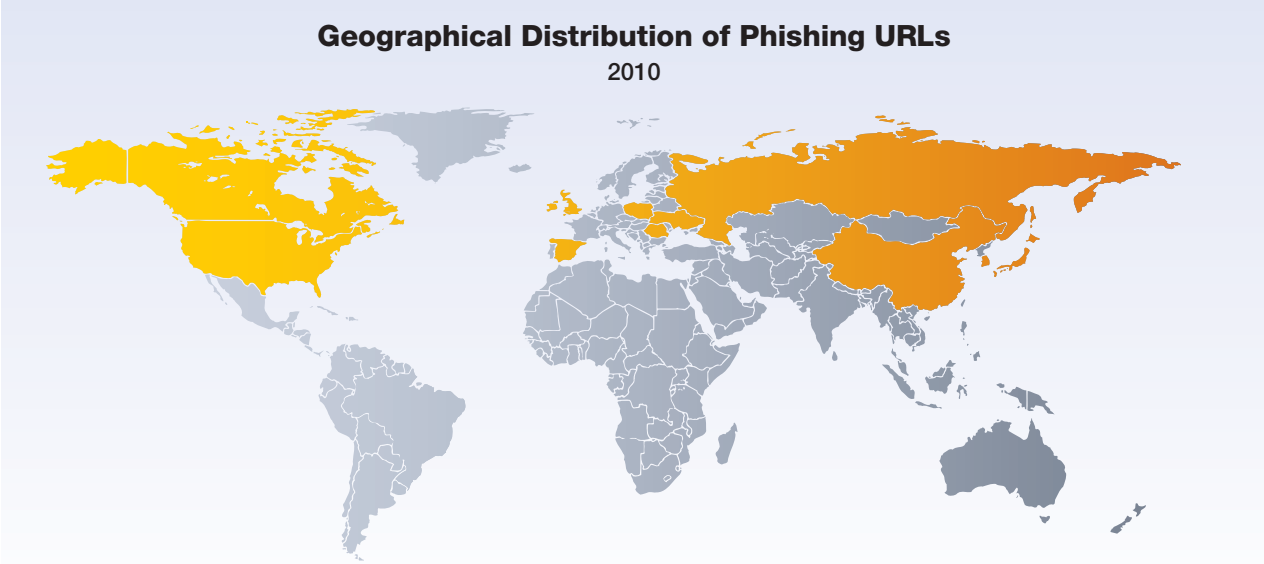


Figure 41: Geographical Distribution of Phishing URLs – 2010

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

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

Table 9: Geographical Distribution of Phishing URLs – 2010

Phishing URLs—country of origin trends

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

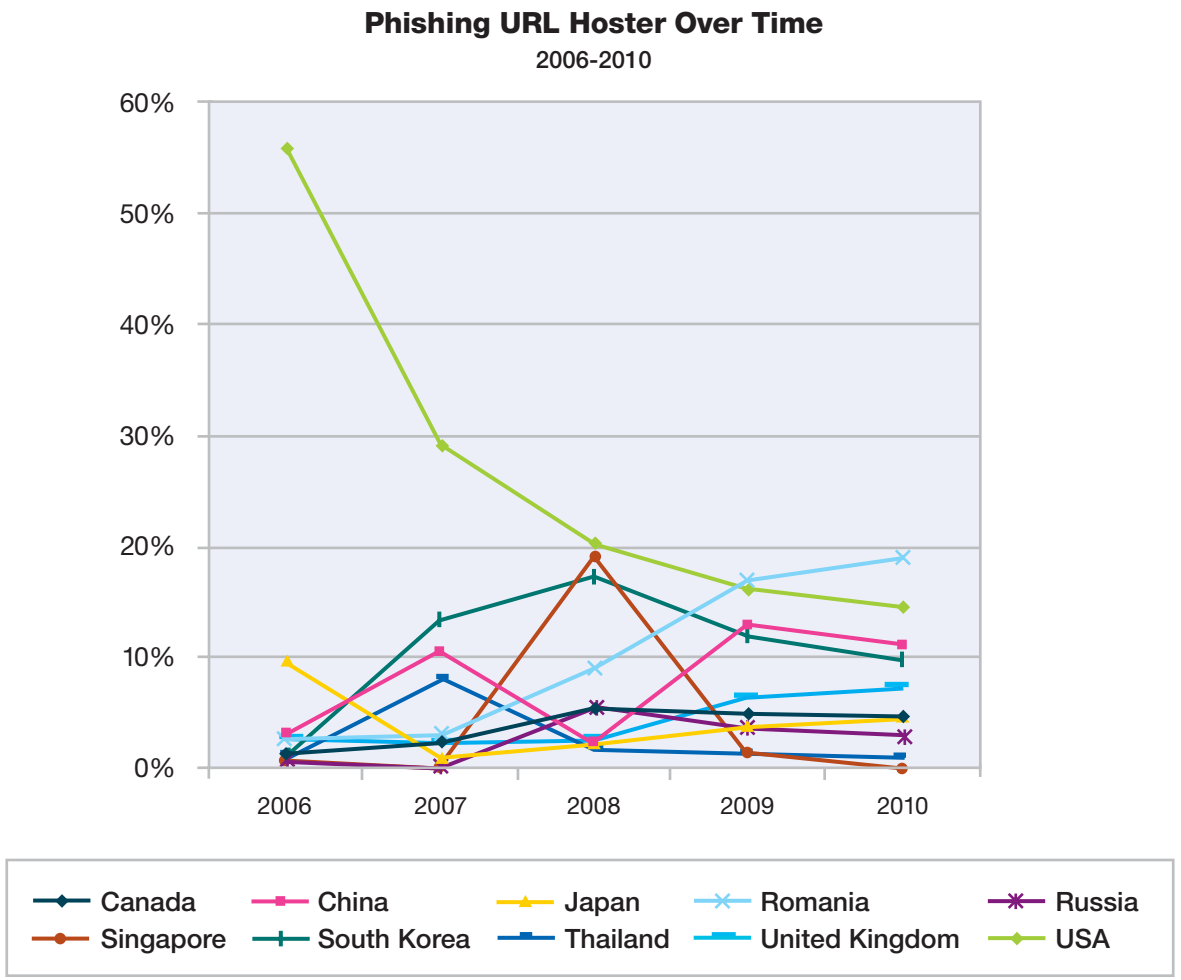


Figure 42: Phishing URL Hoster Over Time – 2006-2010

Phishing—most popular subject lines

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

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

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

Table 10: Most popular phishing subject lines, 2010

Phishing targets

Phishing—targets by industry

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

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

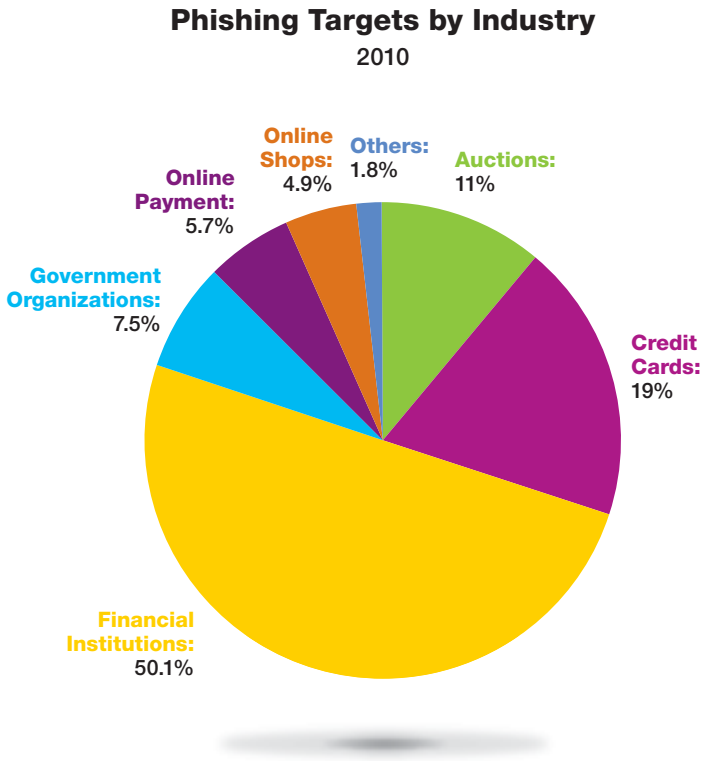


Figure 43: Phishing Targets by Industry – 2010

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

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

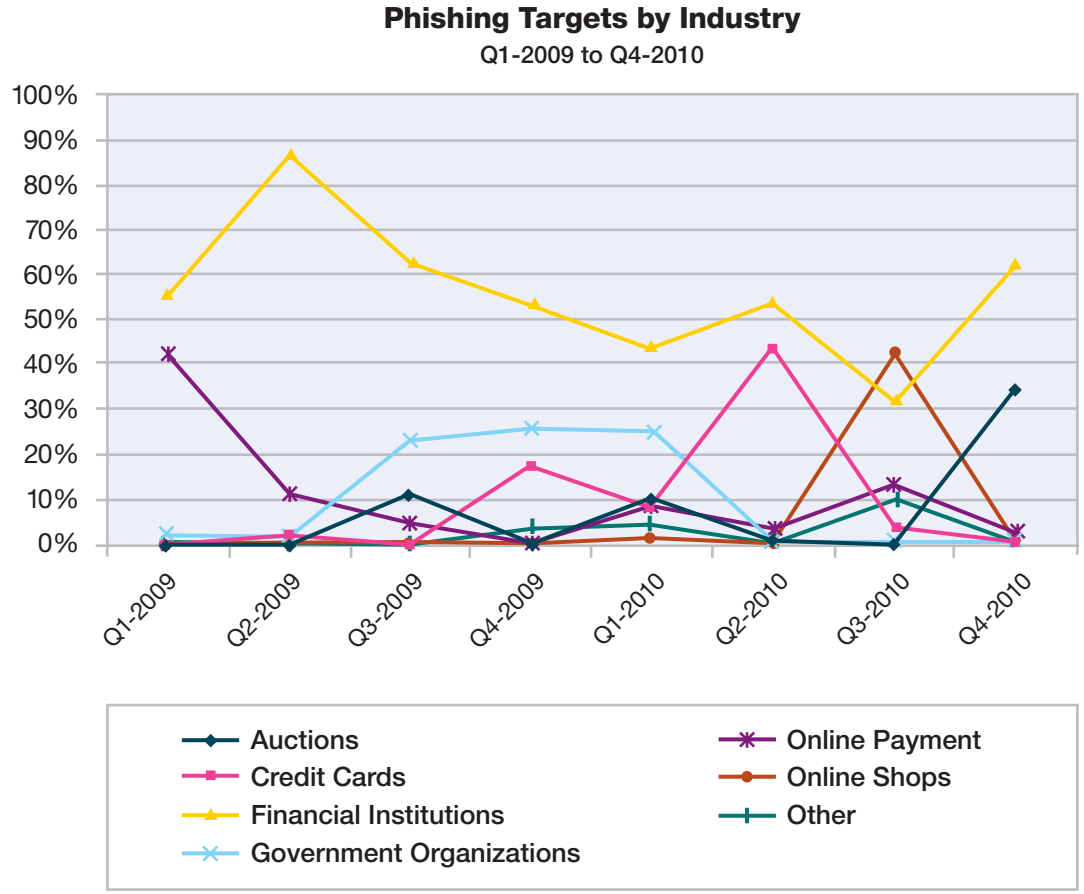


Figure 44: Phishing Targets by Industry – Q1-2009 to Q4-2010

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

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

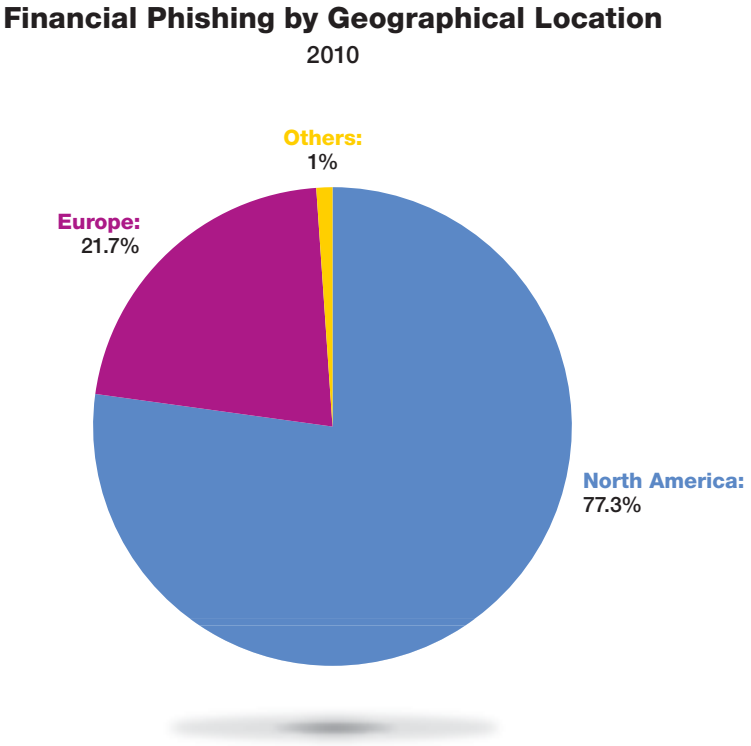


Figure 45: Financial Phishing by Geographical Location – 2010

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

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

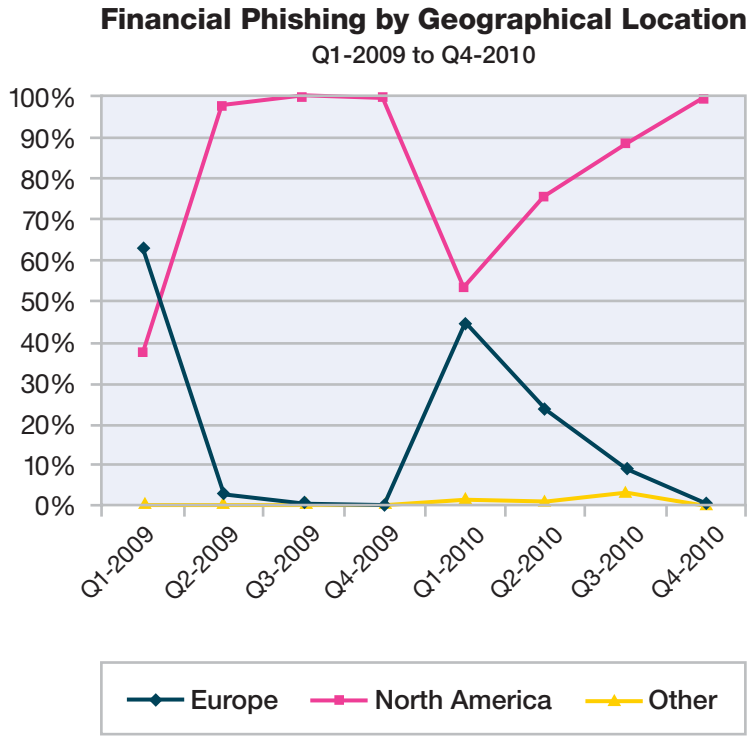


Figure 46: Financial Phishing by Geographical Location – Q1-2009 to Q4-2010

Section II—Operating Secure Infrastructure

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

Advanced persistent threat (APT) and targeted attacks

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

Background and definitions

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

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

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

Response and research

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

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

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

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

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

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

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

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

Conclusions and recommendations

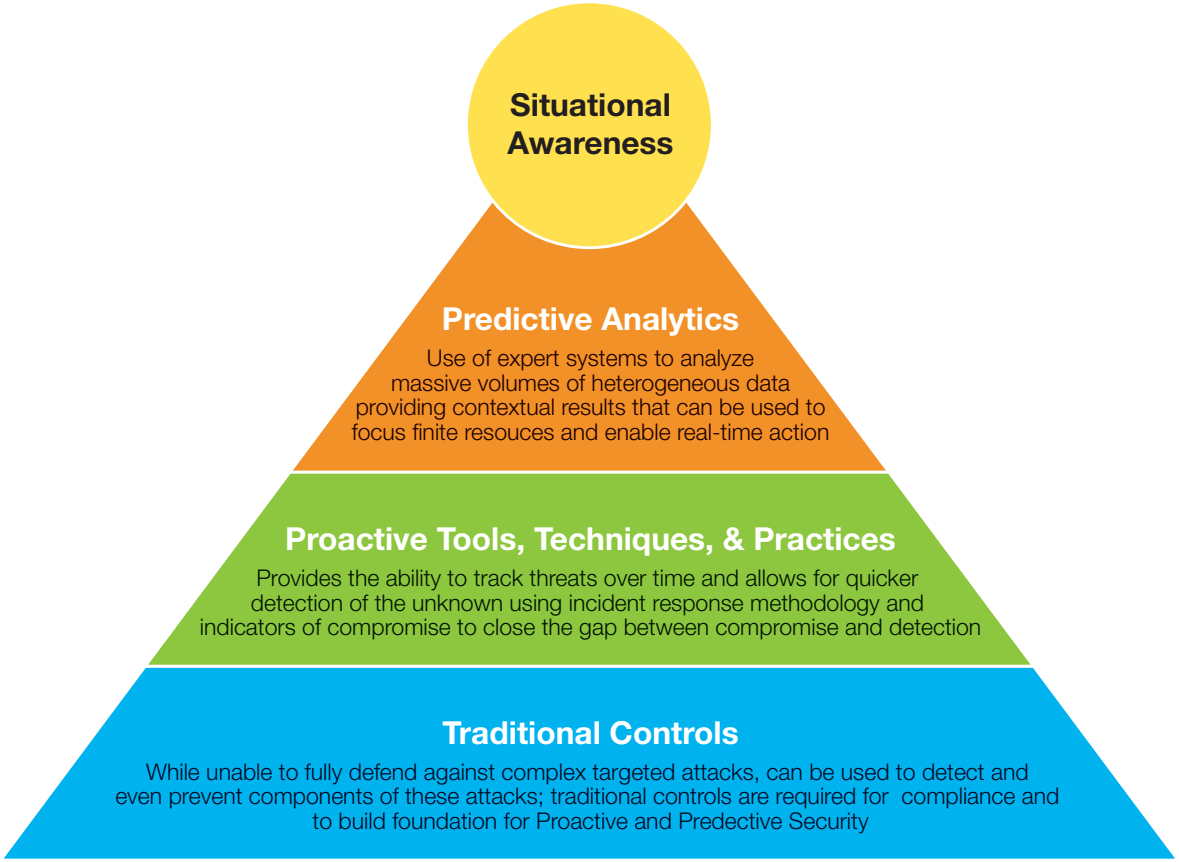
In summary, this is a dynamic and challenging problem; however, it is our strong belief that significant steps can be taken to understand and combat this type of threat through better situational awareness.

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

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

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

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



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

Stuxnet and SCADA

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

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

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

Who is behind Stuxnet?

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

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

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

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

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

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

Works cited

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

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

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

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

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

Public vulnerability disclosures in 2010

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

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

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

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

2010—A record setting year

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

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

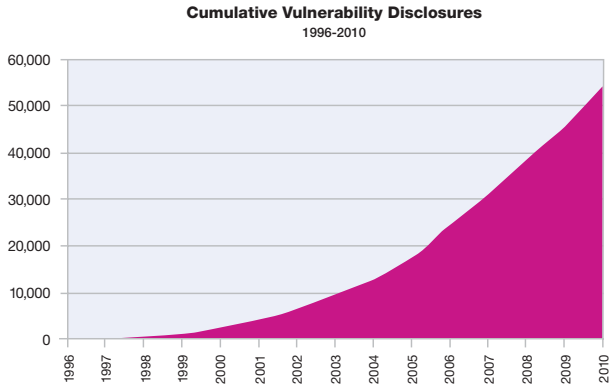


Figure 47: Cumulative Vulnerability Disclosures – 1996-2010

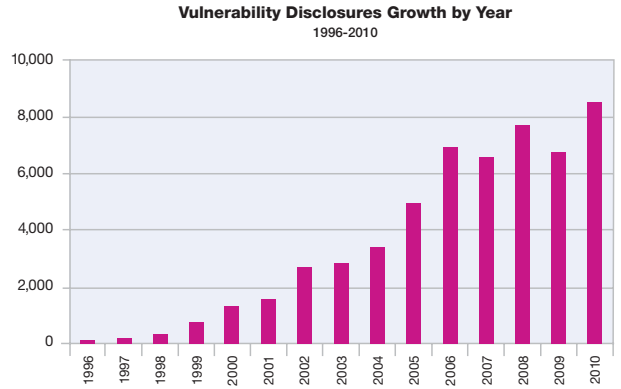


Figure 48: Vulnerability Disclosures Growth by Year – 1996-2010

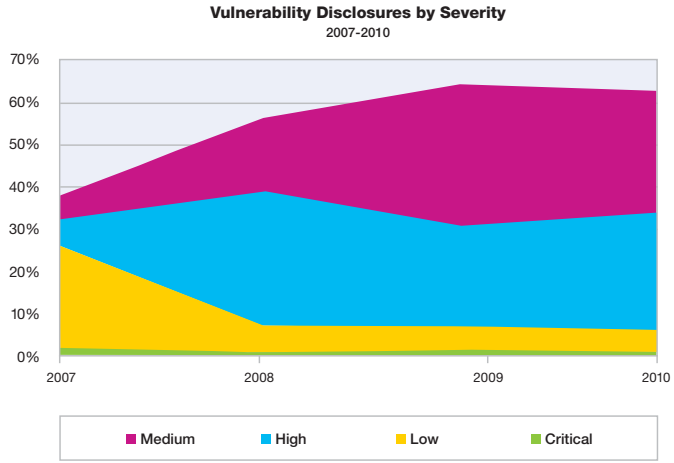


Figure 49: FVulnerability Disclosures by Severity – 2007-2010

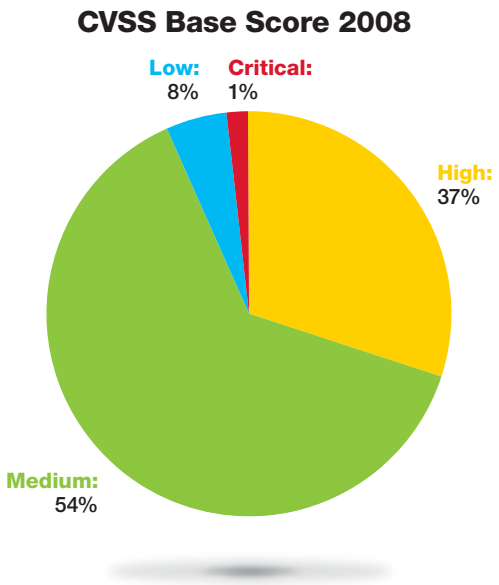


Figure 50: CVSS Base Score 2008

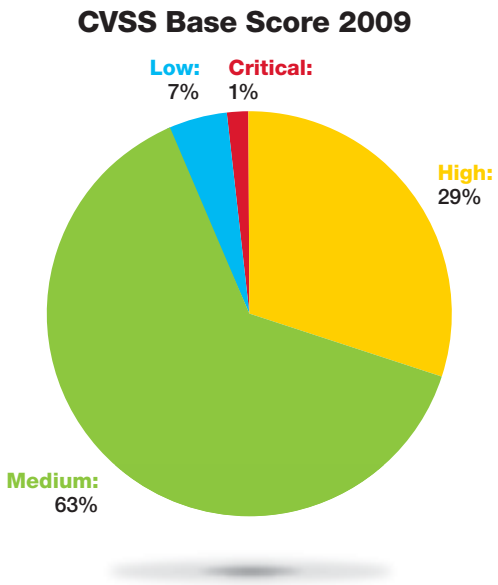


Figure 51: CVSS Base Score 2009

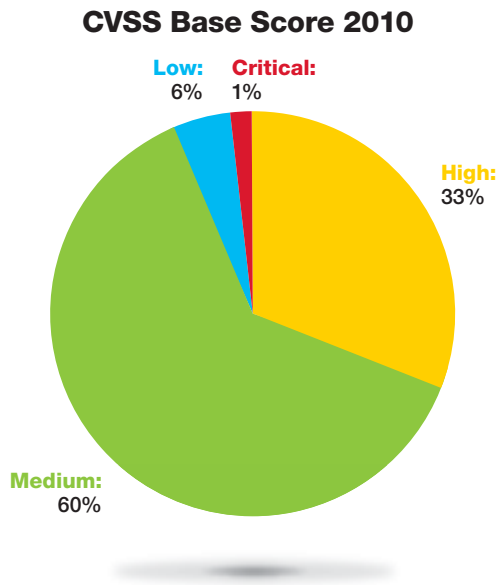


Figure 52: CVSS Base Score 2010

Were there really more vulnerability disclosures in 2010?

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

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

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

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

Public exploit disclosure

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

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

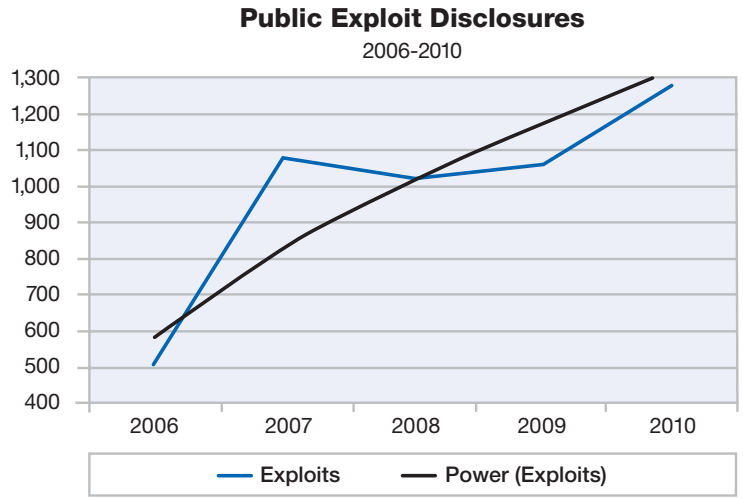


Figure 53: Public Exploit Disclosures – 2006-2010

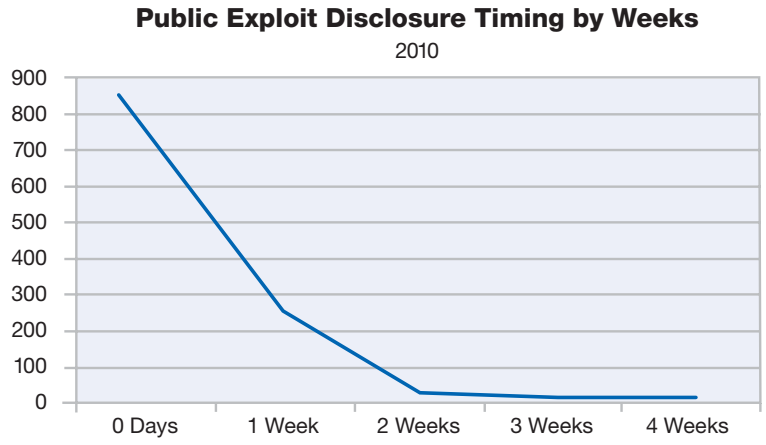


Figure 54: Public Exploit Disclosure Timing by Weeks – 2010

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

Table 11: Public exploit disclosures 2006 – 2010

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

Table 12: Public Exploit Disclosure Timing by Weeks – 2010

Vendor supplied patches

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

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

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

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

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

Table 13: Patch release timing 2010

Toward more reliable public vulnerability reporting

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

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

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

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

Shift from local to remotely exploitable vulnerabilities

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

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

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

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

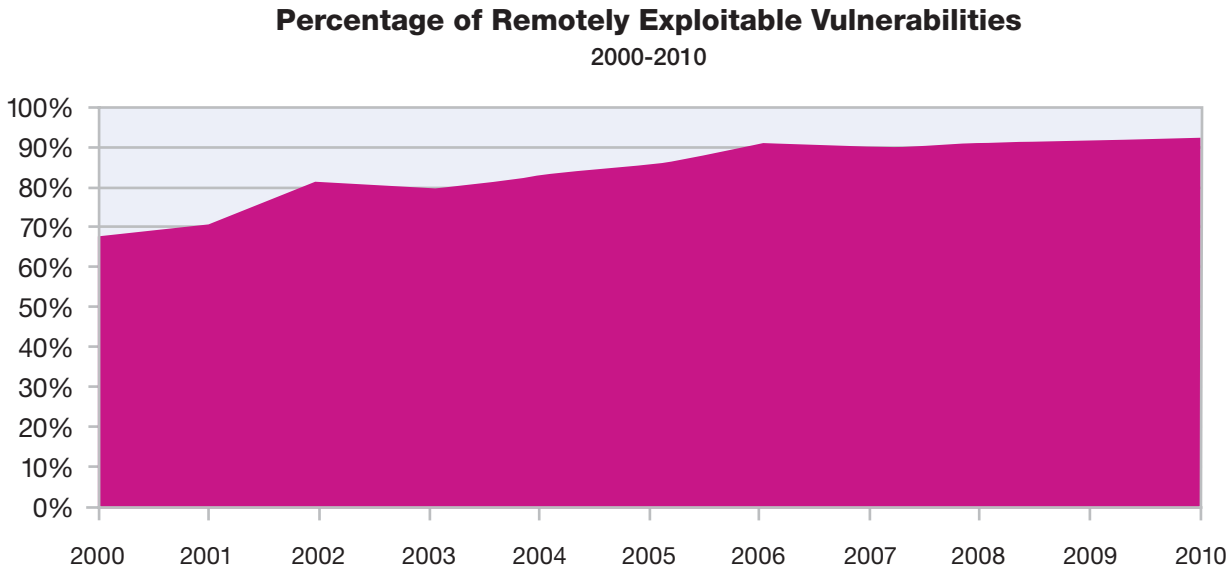


Figure 55: Percentage of Remotely Exploitable Vulnerabilities – 2000-2010

Web application vulnerabilities

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

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

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

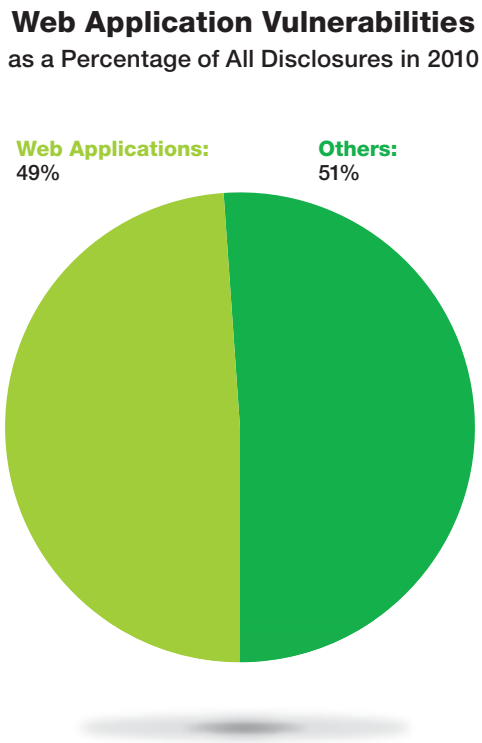


Figure 56: Web Application Vulnerabilities as a Percentage of All Disclosures in 2010

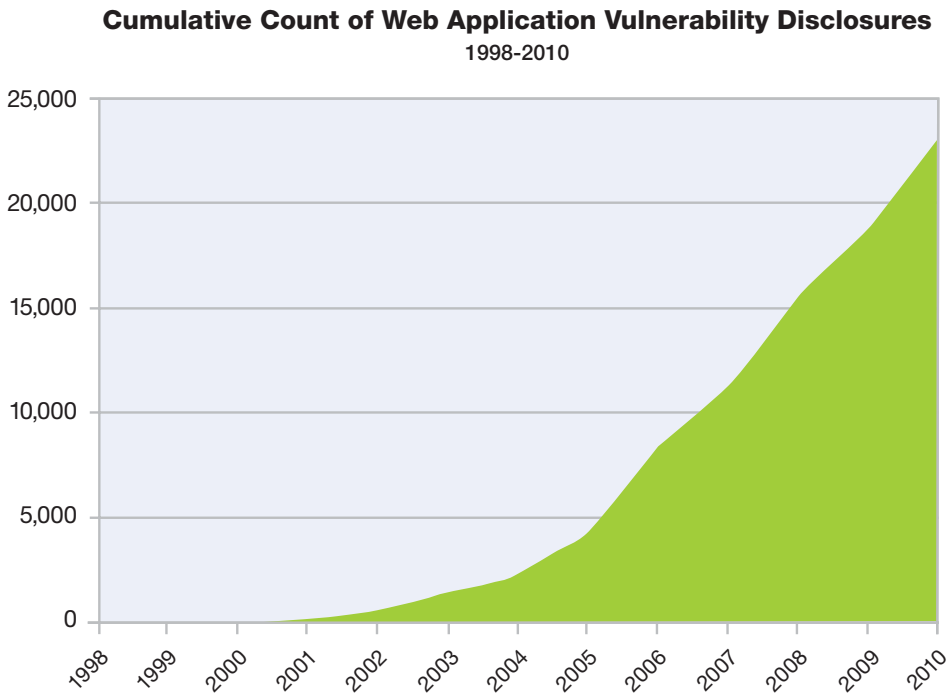


Figure 57: Cumulative Count of Web Application Vulnerability Disclosures – 1998-2010

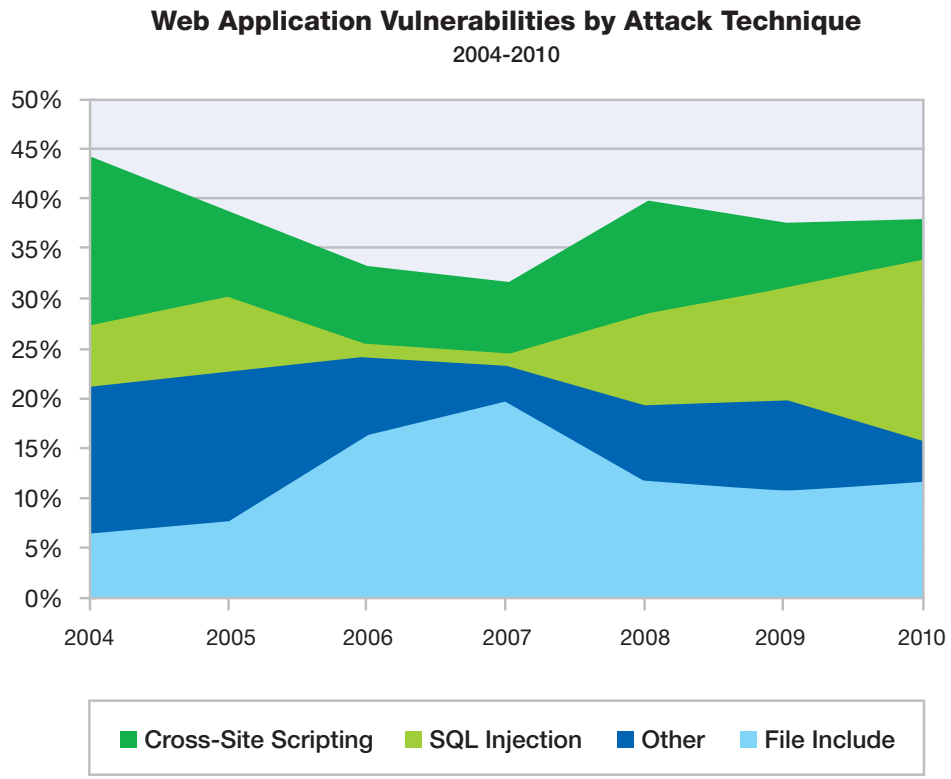


Figure 58: Web Application Vulnerabilities by Attack Technique – 2004-2010

Web application platforms vs. plug-ins

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

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

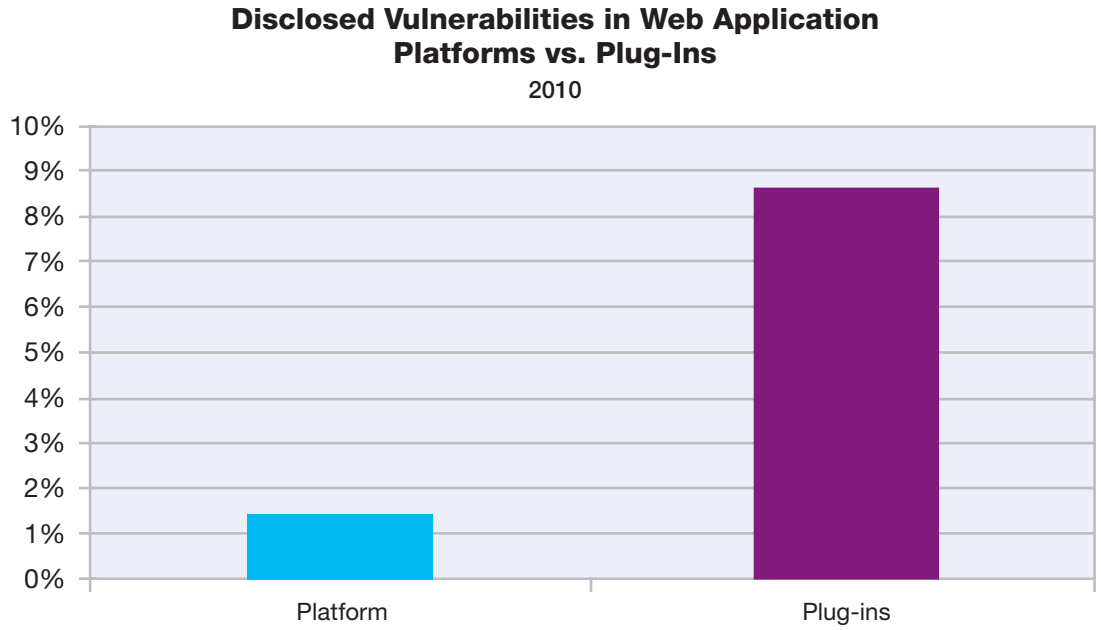


Figure 59: Disclosed Vulnerabilities in Web Application Platforms vs. Plug-Ins – 2010

Client-side vulnerabilities and exploits

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

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

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

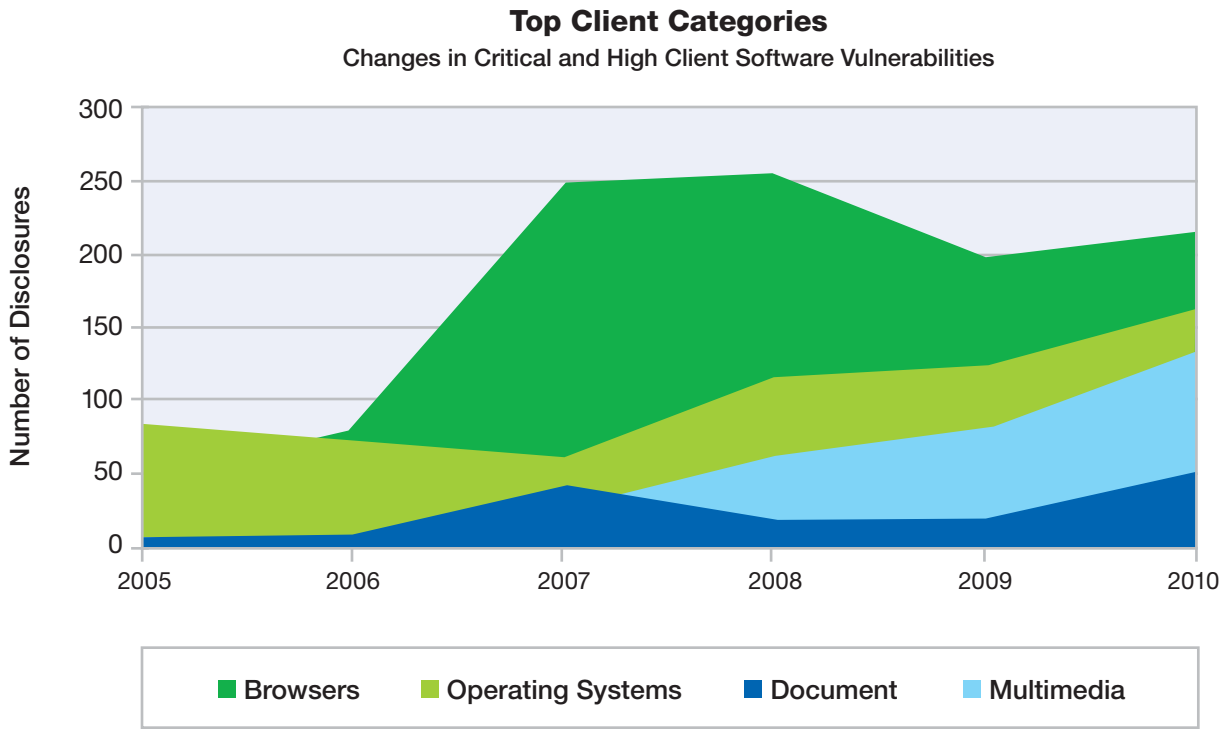


Figure 60: Top Client Categories – Changes in Critical and High Client Software Vulnerabilities

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

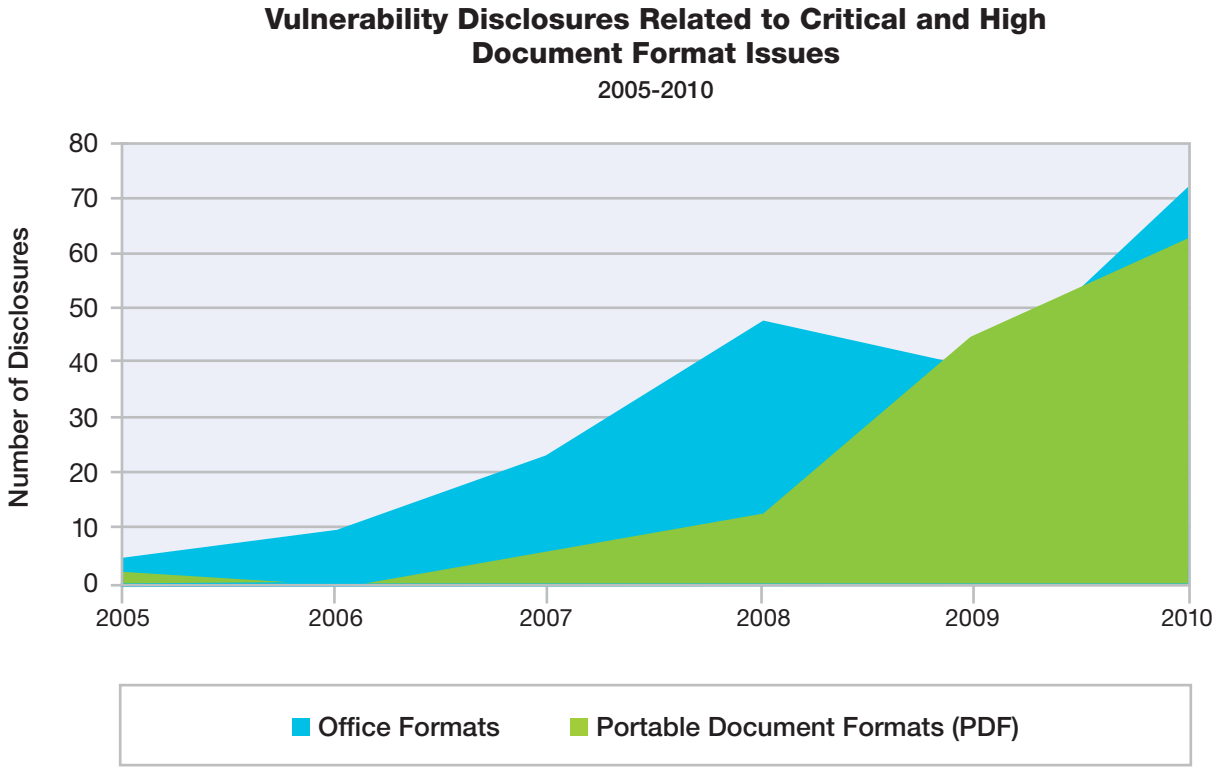


Figure 61: Vulnerability Disclosures Related to Critical and High Document Format Issues – 2005-2010

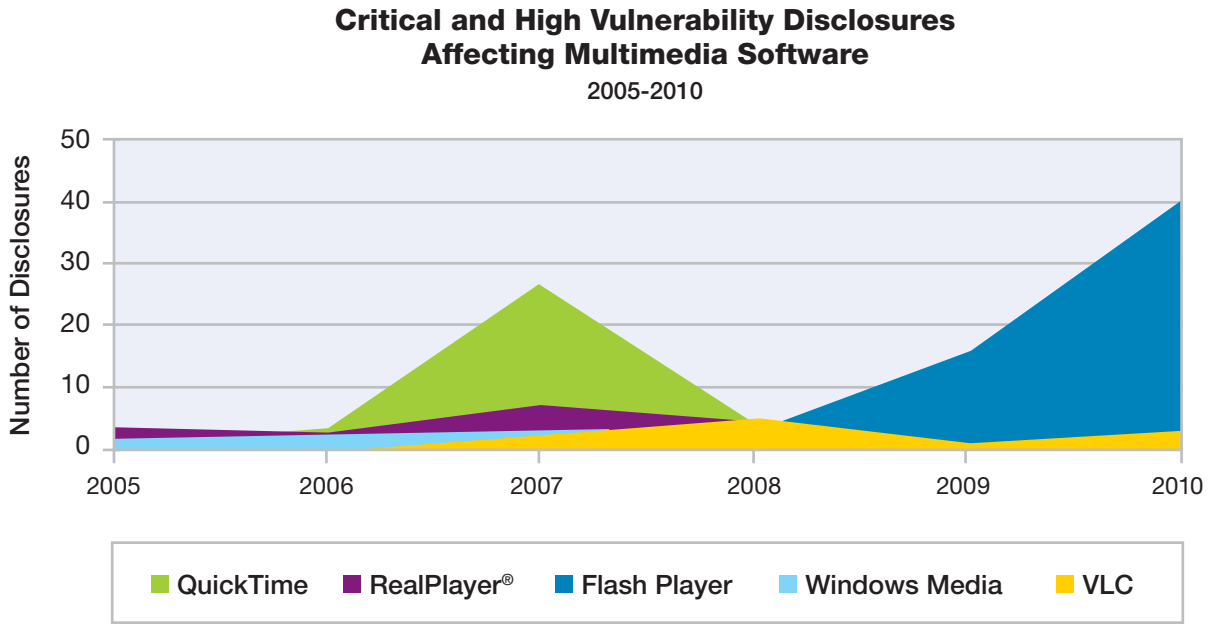


Figure 62: Critical and High Vulnerability Disclosures Affecting Multimedia Software – 2005-2010

Exploit effort versus potential reward matrix

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

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

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

Figure 63: Exploit Effort vs. Potential Reward

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

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

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

Key Recommendations

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

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

Virtualization—risks and recommendations

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

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

Virtualization system components

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

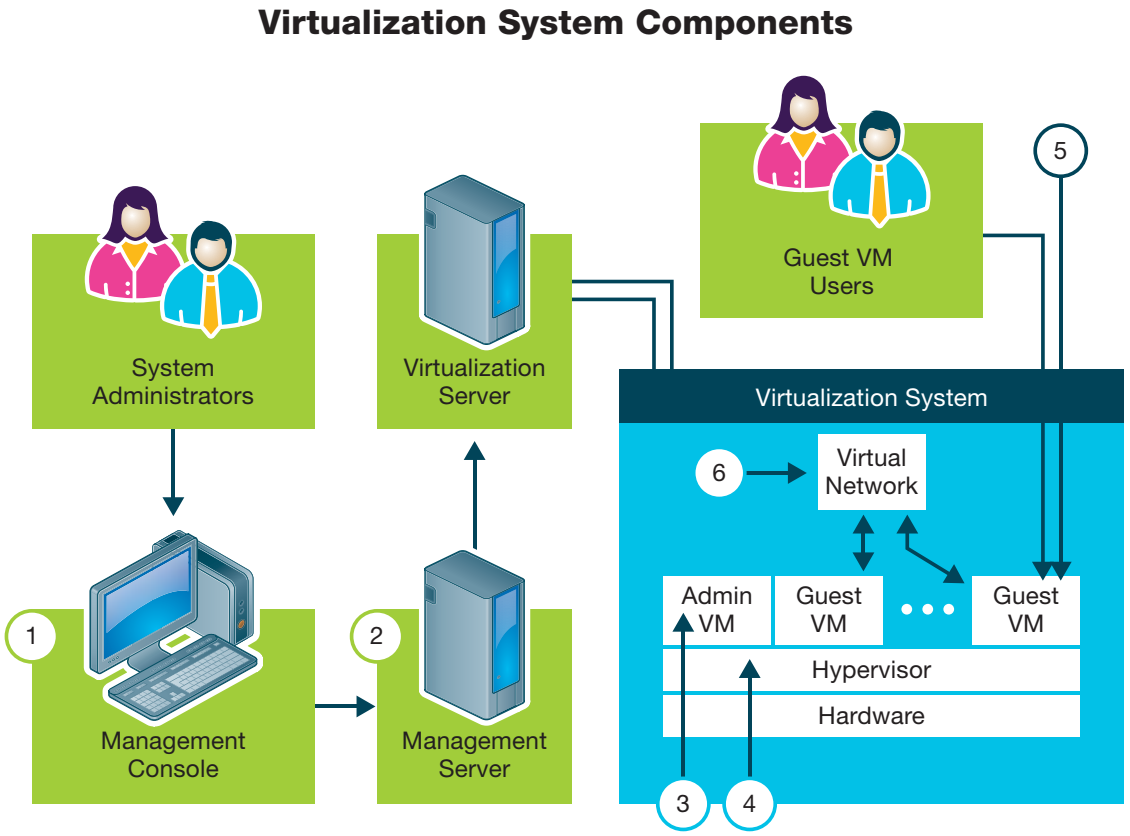


Figure 64: Virtualization System Components

1. Management console

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

Management console vulnerabilities

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

2. Management server

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

Management server vulnerabilities

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

3. Administrative VM

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

Administrative VM vulnerabilities

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

4. Hypervisor

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

Hypervisor vulnerabilities

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

5. Guest VMs

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

Guest VM vulnerabilities

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

6. Virtual network

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

Virtual network vulnerabilities

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

Vulnerability distribution

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

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

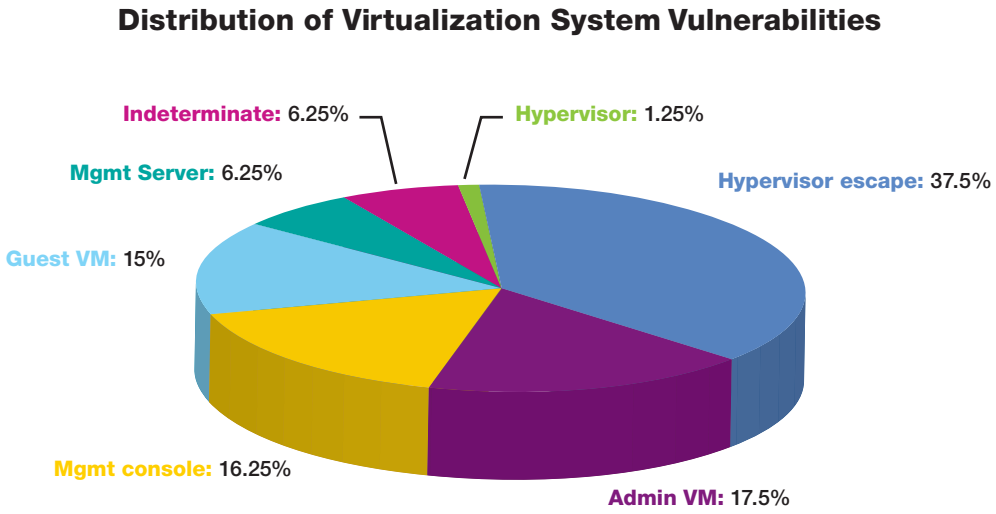


Figure 65: Distribution of Virtualization System Vulnerabilities

Attacks unique to virtualization systems

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

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

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

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

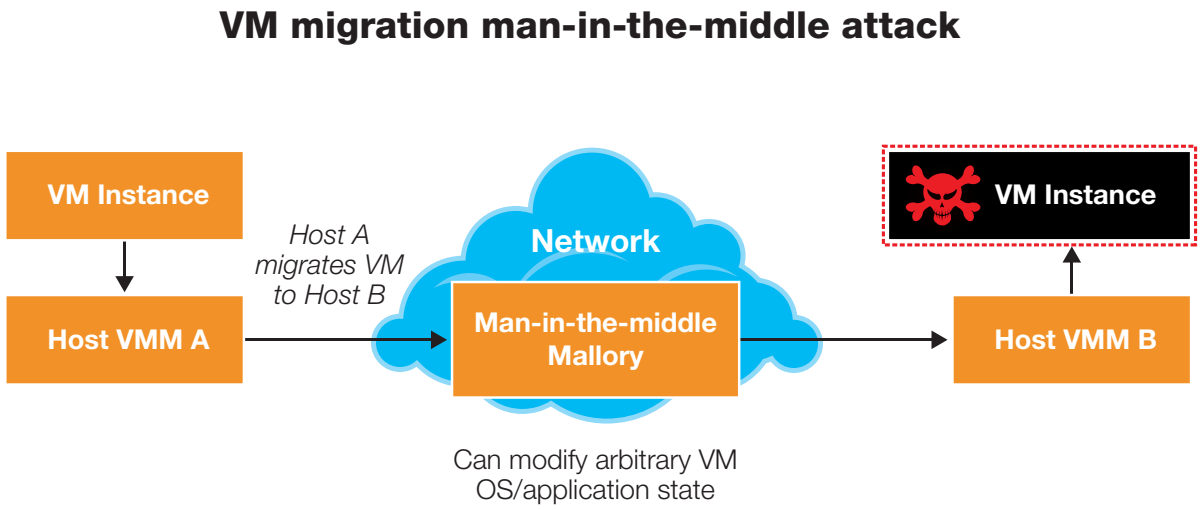


Figure 66: VM migration man-in-the-middle attack

Public exploits

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

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

Summary of security concerns

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

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

Operating Secure Virtual Infrastructure

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

Management server

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

Administrative VM

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

Hypervisor

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

Guest VMs

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

Virtual network

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

Endpoint security and systems management

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

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

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

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

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

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

A well-managed device is a more secure device

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

Basic device management hygiene should include

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

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

Case study one: large technology company

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

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

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

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

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

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

Case study two: public sector

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

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

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

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

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

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

The State of Affairs in DNSSEC

Introduction

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

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

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

2010 The year in review

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

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

Software deployment and components

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

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

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

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

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

DNSSEC challenges and stumbling blocks

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

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

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

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

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