December 2011

IMPERVA®

# Hacker Intelligence Initiative, Monthly Trend Report #6

## Security Trends 2012

*Hacking is inherently innovative. This means security teams, like Mr. Gretzky, need to keep their eye on where things are going – not just on where they've been. As 2012 approaches, security has evolved dramatically from just one year ago. The word "hacktivism," for example, is almost a household term. Likewise, the group Anonymous is anything but. Indeed, cyber security remains one of the most dynamic and fluid disciplines worldwide.*

> *A good hockey player plays where the puck is. A great hockey player plays where the puck is going to be.*
> **– Wayne Gretzky**

*Imperva's Application Defense Center (ADC), led by Imperva CTO Amichai Shulman, is exclusively focused on advancing the practice of data security to help companies shield themselves from the threat of hackers and insiders. For 2012, the ADC has assembled a comprehensive set of predictions designed to help security professionals prepare for new threats and attacks in cyber space.*

***Trend #9: SSL Gets Hit in the Crossfire***

***Trend #8: HTML 5 Goes Live***

***Trend #7: DDoS Moves Up the Stack***

***Trend #6: Internal Collaboration Meets Its Evil Twin***

***Trend #5: NoSQL = No Security?***

***Trend #4: The Kimono Comes Off of Consumerized IT***

***Trend #3: Anti-Social Media***

***Trend #2: The Rise of the Middle Man***

***Trend #1: Security (Finally) Trumps Compliance***

## Trend #9: SSL Gets Hit in the Crossfire

While a growing number of web applications are delivered over the HTTPS protocol (HTTP over SSL), attackers are increasingly focusing their attacks against the various components of SSL. We are seeing a rise in attacks which target the worldwide infrastructure that supports SSL. We expect these attacks to reach a tipping point in 2012 which, in turn, will invoke a serious discussion about real alternatives for secure web communications.

Ironically enough, while attackers are keeping busy attacking SSL, they are also abusing its privacy features in order to conceal their own mischievous deeds. We therefore expect to see more general purpose web attacks being launched over SSL connections.

First, a little backgrounder. The Secure Sockets Layer (SSL)[1] cryptographic protocol is the de facto standard for providing data integrity and confidentiality for web transactions over the Internet (sometimes SSL is used interchangeably with the term HTTPS which is the application of SSL protocol to HTTP traffic). SSL encrypts pieces of application layer data over TCP connections providing confidentiality. It can also be used to test for the identity of the server, the client or both. SSL uses an efficient cryptographic algorithm for encrypting data and a computational intensive protocol for authentication and key exchange (the key is used by the encryption algorithm). The key exchange protocol employs asymmetric cryptography a methodology that requires the existence of a worldwide Public Key Infrastructure (PKI). PKI defines a procedure for binding digital certificates with respective websites by means of a chain of Certificate Authorities (CA). The binding is established through a registration and issuance process that ensures non-repudiation.

In the last couple of years, we have seen a growing awareness for attacks against confidential (e.g. Firesheep) and authenticity (Man in the Middle attacks, Phishing). As a result, web application owners are constantly extending the use of SSL to more applications, and to more parts of their applications. A good example is the evolution of the Google interface. At first, only the login page was encrypted. In the next stage, the whole Gmail service supported encryption – by default. Google has now even added the search functionality to be accessed via HTTPS.

With the growing usage of SSL, attackers are increasingly targeting the SSL layer. Unfortunately, most of the research community is focused on pointing out inherent protocol vulnerabilities, or common implementation mistakes that could potentially be attacked. While, the attacker community is focused on other, more practical types of attacks:

- **Attacks against PKI**. Over the past year, attackers have repeatedly compromised various CA organizations. These include, DigiNotar, GlobalSign, StartSSL, Comodo and Digicert Malaysia. These attacks were a direct consequence of the commoditization of certificates, where smaller, less competent organizations have started to obtain a bigger share in the Certificate Authority market. As it stands now, any CA can issue a digital certificate for any application – without any required consent from application owner. A hacker, who gains control on any CA, can then use it to issue fraudulent certificates and impersonate any website. Additionally, there are concerns that some root CAs (whose trust is hardcoded into browser software) are inherently dubious (e.g. controlled by unfriendly governments). Some efforts are made to amend PKI issues but they are far from broad acceptance[2].
- **The theft of issued certificates**. We believe this attack will prevail over the next year as application certificates are no longer limited to being stored by the application. This is the consequence of the monolithic nature of SSL. While SSL prevents access to traffic by attackers it has no built-in mechanisms that restrict access to it by collaborative 3rd parties. For example, proxies, load balancers, content delivery networks (CDNs) need to access the certificate's private key in order to access application data. Also DLP and WAF solutions require similar key access. In these cases, it would be preferable that the intermediate proxies would be able to look at message headers, or to be able to read traffic without changing it. However, this granularity is not supported by SSL. As a result, the digital certificate is now stored in many locations – some residing outside of the site's physical environment and out of the application's owner control. These open up additional attack points which provide higher success rates for attackers.

[1] SSL per se is now obsolete and replaced by the Transport Layer Security (TLS) protocol. However SSL is still the commonly used term.

[2] Another worthy example is the convergence project http://convergence.io/

› **Denial of Service attacks**. The heavy computational burden incurred by the SSL-handshake process leaves SSL-protected resources prime candidates for effective Denial of Service (DoS) attacks. Together with an increased consumption of computer resources per session, a multitude of simple attacks can be devised very efficiently.

In addition to the attacks against SSL and its infrastructure, hackers will leverage SSL to carry out their attacks with increased confidentiality. For example, intermediate proxies cannot add headers to indicate original sender IP address – leading to the loss of traceability. Another problem is the loss of information when following a link from an SSL page to a non-SSL page. An attacker can exploit this implementation in order to cover the tracks of various Web attacks. Furthermore, many security devices which require inspection of the Web traffic lose this sort of visibility due to the encryption of the traffic.

## Trend #8: HTML 5 Goes Live

Over the last few years vulnerabilities in browsers' add-ons (third party components such as adobe's Flash Player or Oracle's Java) were the main cause for "zero-day" exploits. These are un-patched application vulnerabilities that are exploited in order to install malware on web users' machines. We predict in 2012 hackers will shift their focus to exploiting vulnerabilities in the browsers themselves in order to install malware. The reason is due to recently added browser functionality – mainly driven by the adoption of HTML 5 standard.

The HTML 5 standard was created to enable browsers to support a richer end user experience in a standardized way. Most notably, HTML 5 adds support for audio, video, 2D graphics (SVG), 3D graphics (WebGL) that previously required the end user to install a dedicated add-on. (e.g. Adobe Flash Player to watch online video).

While the new features are attractive to web developers, they are also very beneficial for hackers. We see security repercussions for the following reasons:

1. **New code is generally more vulnerable**. When you write code you are doomed to create bugs and security vulnerabilities along with it. When you add a lot of new code – you are doomed to create a lot of new vulnerabilities.
2. **Compressed media types are more vulnerable**. Modern media types (such as video) are usually highly compressed and optimized to ensure the efficiency of their transmission and display. Decompressing involves a lot of buffer manipulations which are notoriously vulnerable.
3. **Hardware access**. Many browsers use the assistance of hardware components[3] – mainly for Javascript and graphics acceleration – in order to achieve higher efficiency and create a smoother user experience. Since hardware is run under high permission access levels, and usually cannot be protected by the operating systems, exploits targeting the hardware components are very attractive to attackers. This type of privileged access provides the attackers with a method to exploit buggy hardware drivers straight from a webpage.
4. **End users control**. Currently, most browsers contain a mechanism which turns off a vulnerable browser add-on. In the case of HTML 5, the implementation is embedded within the browser so that a vulnerable add-on might not necessarily be turned off. At the very least, it changes the security model from "opt in" model (actively download an add on) to "opt out" (disable an existing component.)
5. **Javascript control**. New HTML 5 features can be controlled and manipulated via Javascript. This gives rise to new vectors of Javascript-related attacks (mainly, but to not limited to, XSS). These new attack vectors will use the new elements, and the interactions between them, in order to break the already fragile Same Origin Policy (SOP). For more on SOP, click here.
6. **Ubiquity**. It's much more cost-effective to create a cross browser exploit than to create an exploit aimed at a specific one. The ubiquity of HTML 5 provides them with just that.

[3] Microsoft Announces Hardware-Accelerated HTML5 http://www.microsoft.com/presspass/press/2010/mar10/03-16mix10day2pr.mspx

## Trend #7: DDoS Moves Up the Stack

Distributed Denial of Service (DDoS) attacks are gaining popularity and were part of high profile hacking campaigns in 2011, such as the Anonymous attacks[4]. We predict that in 2012 attackers will increase the sophistication and effectiveness of DDoS attacks by shifting from network level attacks to application level attacks, and even business logic level attacks.

A Denial of Service (DoS) is a relatively old attack aimed at data availability by exhausting the server's computing and network resources. Consequently, legitimate users are denied service. A Distributed Denial of Service (DDoS) is an amplified variation of the DoS attack, where the attacker initiates the assault from multiple machines to mount a more powerful and coordinated attack.

Today, DoS attacks require the attacker to invest in a massively distributed network which can create enough traffic to eventually overwhelm the victim's resources. At the other end of the DoS spectrum, there's the SQL shutdown command. An attacker exploiting an application vulnerability can use this particular command to shut down the service using just a single request, initiated from a single source, which, from the attacker's perspective, proves cheaper and is just as effective. Historically, we have seen DoS attacks gradually climb up the protocol stack. From the most basic Network layer (layer 3) attacks, such as the UDP Flood, through the Transport layer (layer 4) with SYN flood attacks. In the last years, we also saw the HTTP layer (layer 7) being targeted with such attacks as the Slowloris[5] (in 2009) and RUDY[6] (2010) attack.

We predict that in 2012 we will see hackers advance one more rung. This means creating DDoS attacks by exploiting web application vulnerabilities, or even through web application business logic[7] attacks. Indications for this trend are already emerging. For example, the #RefRef tool[8], introduced in September 2011, exploits SQL injection vulnerabilities used to perform DoS attacks.

There are several reasons attackers are moving up the stack:

1. **Decreasing costs**. In the past, attackers have taken the "brawn over brains" attitude. This meant that they simply inundated the application with garbage-like requests. However, these type of attacks require a large investment on the attacker's side, which include distributing the attack between multiples sources. In time, hackers have discovered that they can add "brains" to their attack techniques, significantly lowering the heavy costs associated with the "brawn" requirements.
2. **The DoS security gap**. Traditionally, the defense against (D)DoS was based on dedicated devices operating at lower layers (TCP/IP). These devices are incapable of detecting higher layers attacks due to their inherent shortcomings: they don't decrypt SSL, they do not understand the HTTP protocol, and generally are not aware of the web application. Consequently, the attacker can evade detection in these devices by moving up the protocol stack.
3. **The ubiquitous DDoS attack tool**. Working over the HTTP layer allows the attacker to write code independent of the operating system. For example, by using javascript. The attacker then gains the advantage of having every web enabled device participate in the attack, regardless of its operating system – be it Windows, Mac or Linux. More so, it allows mobile devices- running iOS, Android, or any other mobile operating system – to participate in such attacks.

The good news is that enterprises can prepare themselves against these application-targeted DoS attacks. How? By adding application-aware security devices, such as Web Application Firewalls (WAFs). These devices can decrypt SSL, understand HTTP and also understand the application business logic. They can then analyze the traffic and sift out the DoS traffic so that eventually, the business receives – and serves – only legitimate traffic.

[4] http://thelede.blogs.nytimes.com/2010/12/08/operation-payback-attacks-visa/?partner=rss&emc=rss

[5] http://ha.ckers.org/slowloris/

[6] http://www.slideshare.net/AlesJohn/owasp-universalhttpdo-s-9207289

[7] Web application logic attack can be performed by profiling the victim web application for resource consuming operations (such as searching a large database) and then constantly applying that operation to deplete the victim server resources.

[8] http://www.refref.org/

## Trend #6: Internal Collaboration Meets Its Evil Twin

We expect to see a growing number of data breaches from internal collaboration platforms used externally. Why? Internal collaboration suites are being deployed in "evil twin" mode, i.e., these suites get used externally. As a result, organization will look for tools to protect and control access to such platforms. We estimate that in 2012 the number of Internet sites based on such platforms will increase dramatically. As a consequence, the number of security incidents due to inadvertent public exposure of confidential data will grow.

The past couple of years brought up an extensive increase in the use of collaboration suites within organizations. Platforms such as Microsoft SharePoint and Jive are now used by many organizations to share information and manage content. While most enterprises use these applications within the organization, some have also **extended** the use to partners and even to the public through an internet facing website. In fact, based on Forrester research, SharePoint is listed as the number one portal product (source: http://www.topsharepoint.com/about) and with the latest release of SharePoint 2010, it also offers a great platform for building collaboration sites with external partners or robust externally-facing sites. Extending an internal platform to external use always comes with a price tag to be paid in security. An example of such security breach took place when the Mississippi national guard accidentally exposed personal information of nearly 3000 soldiers on their external Microsoft SharePoint website (source: http://www.itbusinessedge.com/cm/community/news/sec/blog/national-guard-data-exposed-in-accidental-security-breach/?cs=43893)

There are two major factors that impact the risk of extending an internal platform to external use:

1. **Data segregation**. Data segregation has two manifestations with respect to externalizing internal systems. Ensuring that the stored sensitive data does not become accessible through the less restricted interfaces of the platform is not an easy task. For the entire lifetime of the systems, controls should be put in place to allow collaboration and sharing of sensitive information within the organization while keeping it out of the reach of the general public.
2. **Threat profile**. Threat profile is related to the difference between internal and external threats. The size of potential attacker population increases instantaneously as well as the technical and hacker skills of it. At the same time, the impact of a disclosure or a breach increases dramatically over that of an internal breach. To make things even worse, search engines like Google constantly crawl and update their indexing policies so that th e public interface of the application, as well as any breaches or mis-configured entry points are quickly apparent to the whole world. For example, an updated Google policy to index FTP servers resulted in a breach affecting 43,000 Yale-affiliated individuals. Google hacking tools, such as SharePoint GoogleDiggity and SharePointURLBrute, can easily be used to identify insecure configurations.

Organizations aimed at reducing the risk of massive exposures should start budgeting and planning for the next generation of collaboration suite monitoring and governance tools. Some of the characteristics to look for are:

- Policies to monitor and protect internet and intranet facing sites.
- Flexible deployment that doesn't impact the use of application or the network architecture.
- The ability to identify excessive user rights to content.

## Trend #5: NoSQL = No Security?

The IT world is quickly embracing Big Data. Huge data stores are the next big step in analyzing the massive amounts of data that is being collected in order to identify trends. For example, new start ups use these systems to analyze trillions of DNA strips to gain an understanding of our genealogy. To well-established companies who are adopting the technology to map and time transportation systems across the world to make our traveling easier and cheaper. While Big Data is becoming a buzzword in information systems, there has not been much investigation into the security implications. Many predict that in 2012 we'll see a growing interest in Big Data and it's underlying technology, NoSQL. We predict that the inadequate security mechanisms of these systems will inhibit enterprises from fully integrating these systems as third party components within the corporation.

NoSQL is a common term to describe data stores that store all types of data – from structured to unstructured. Due to this diversity, these data stores are not accessed through the standard SQL language. Up until recently, we categorized our conception of data stores in two groups: relational databases (RDBMS) and file servers. The new kid in town, NoSQL, opened our minds to a database that, unlike the conventional relational concepts, does not follow a structural form. The advantage? Scalability and availability. With a technology where each data store is mirrored across different locations in order to guarantee constant up-time and no loss of data, these systems are commonly used to analyze trends. These systems are not suitable for financial transactions requiring a real-time update, but could be employed at a financial institution to analyze the most efficient or busiest branch.

However, as applications using NoSQL are being rolled out, little time has been taken to think or re-think security. Ironically, security in database and file servers have seen their share of problems over the years. And these are systems that have gained mileage over the years which allowed this type of security inspection. We cannot say the same about NoSQL.

Many may claim that the developers of different NoSQL systems have purposefully pushed out security aspects from their systems. For instance, Cassandra has only basic built-in authentication procedures. This lack of security is considered their feature and built in mind that database administrators do not need to trouble themselves with security aspects. Security, then, should be an offloaded process to be dealt with by a dedicated team.

We believe the NoSQL systems will suffer from a number of issues:

› **Lack of expertise**. Currently, there are hardly enough experts who understand the security aspects of NoSQL technologies. When building a NoSQL system, there is no obvious security model that fits. The lack of such a model makes the implementation of security a non-trivial process and requires extensive design. As a result, security features that need to be considered get pushed out over and over again.

› **Buggy applications**. Until third party solutions roll out to provide the necessary security solutions, it is the NoSQL applications that will carry the security load. Issues include:

  - Adding authentication and authorization processes to the application. This requires more security considerations which make the application much more complex. For example, the application would need to define users and roles. Based on this type of data, the application can decide whether to grant the user access to the system.
  - Input validation. Once again we are seeing issues that have haunted RDBMS applications come back and haunt NoSQL databases. For example, in Blackhat 2011, researchers showed how a hacker can use a "NoSQL Injection" to access restricted information. For example, "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" contains a new separate chapter focused solely on the security of programming frameworks used for NoSQL.
  - Application awareness. In the case where each application needs to manage the security, it will have to be aware of every other application. This is required in order to disable access to any non-application data.
  - When new data types are added to the data store, the data store administrator would have to figure out and ensure what application cannot access that specific data.
  - Vulnerability-prone code. There are a certain amount of NoSQL products, but a magnitude more of applications and application server products. The more applications, the more code in general prone to bugs.

- **Data Duplicity**. In NoSQL systems, data is not strictly saved in particular tables. Instead, the data is duplicated to many tables in order to optimize query processing. As a result, it is not possible to classify credit cards according to a particular sensitive table. On the contrary, this type of data can be found in different places: transaction logs, personal details, specific tables which represents all credit cards, and other locations which may have not even been considered.
- **Privacy**. Although our focus is on security, privacy concerns cannot be ignored. Take for example a healthcare platform where providers get together and share patient data. A patient might access the system for genetic information, and later access it in respect to drug info. An application which analyzes this data can correlate the information to find purchasing trends relating to genetics and health. The problem is that this type of correlation was not considered when the data was initially inserted. As a result, the data was never anonymized allowing anyone to identify specific individuals from the bigger picture.

NoSQL is still in its infancy. It will take awhile until we will see these systems fully deployed at the majority of enterprises. For this precise reason it is so important to invest in the in the security of these systems.

## Trend #4: The Kimono Comes Off of Consumerized IT

After being caught off-guard by the process of consumerization of IT, professionals are trying to regain control of corporate data. The problem is that they are doing it the wrong way. Instead of trying to control data at the source, IT organizations try to regulate the usage of end-user devices and de-cloud data access. We expect organizations to spend a lot of time, money and effort on these techniques and technologies next year – with very poor results.

The consumerization of IT refers to the process in which corporate data is increasingly being processed by end-user devices and applications chosen and provided by the end-users themselves. Smart phones, tablets and custom personal laptops are leading this trend with their increasing processing power and storage capabilities, combined with their growing diversity of available applications. These are augmented by the increase of a remote work force and individuals who use home computers and home networks on a regular basis to access corporate resources. This process by itself posses many challenges to an organization that are related to the compromise of information on the device (either physically through loss and theft of the device, or digitally through malware), as well as the compromise of enterprise networks through a compromised device. Coupled with the move of corporate data into the cloud – where corporate data is stored outside of the organization – an even a more difficult problem emerges. With these issues in mind, the organization completely loses control over the entire interaction between end-users and corporate data.

There is a growing trend among IT professionals to try and regain the control of end-user devices. Through different means, organizations are trying to enforce "proper" usage and settings of non-corporate devices. IT departments are attempting to enforce policies such as password strength, device lockup and even remote wiping in the case of device loss. For example, access through the ActiveSync protocol to Microsoft eMail servers can be restricted to devices that implement a specific security policy. Some enterprises also go as far as to try and regulate the devices that are allowed to access enterprise data to those models who posses certain security capabilities. We anticipate that the next step will be to require that certain security solutions be installed on those devices that are allowed to connect to the network (e.g. Lookout or any other mobile AV). In order to reduce the risk of device compromise, enterprises are also trying to enforce any web access from the device to be relayed through the enterprise network where it can be monitored and controlled (which, of course, has severe implications in the case of SSL protected web resources – as explained in a different trend). Further, this approach hopes to bridge gap that exists between user devices and cloud applications that hold enterprise data. The approach described above is bound to fail for quite a few reasons. Most of them stem from overlooking past experience and human nature:

1. **Past is prologue**. The past couple of years have shown that enterprises are failing to prevent the compromise of enterprise computing equipment. Extending the scope of the problem to a larger variety of devices only magnifies the problem:
2. **Maintaining availability**. Organizations that delegate information availability and network accessibility issues to the cloud and then take the approach of tunneling all user device traffic, are going to face major networking issues. Consequently, they will find themselves spending time and money on creating and maintaining the high level of world wide availability which they wanted to avoid in the first place.
3. **User privacy**. There are unsolved issues regarding the impact to user's privacy and the liability of the enterprise to personal information stored on these devices. For instance, remote wipe-out tools cannot differentiate between corporate and personal information.

This upcoming year, organizations are going to spend quite a lot of money and effort before realizing how little improvement this approach brings to enterprise data security. When they do realize the failure of these measures, they are going to look for a different set of solutions that are going to be more tightly coupled to the data itself. Such solutions include monitoring requirements for access to the data stores and strict control of that access.

## Trend #3: Anti-Social Media

As many more organizations are making their way into the social media space, we expect to see a growing impact to the integrity and confidentiality of the enterprise's information. Moreover, hackers will continue to automate social media attacks, further exacerbating the situation. The heart of the problem resides in three separate issues inherent to social networks:

1. **Sharing** – The most important thing to understand about social networks and the tools built on top of them is that they are designed for sharing information – not restricting access to it.

   Enterprises that try to use social media as collaboration suites for internal, sensitive business data – which require different levels of access privileges – are bound to encounter massive data breaches. The reason is not due to flawed access controls and privacy mechanisms. Rather, the restriction of information through these channels is in complete contrast to the concept of such environments which is, in fact, all about sharing. Consequently, organizations should keep an operational copy of all their data in a business system that can provide decent access controls. Data that can be made public can be exported out of this system and posted to the social network. This way, restricted information is kept inside business systems (regardless of whether they are on premise or in the cloud), while public information can be retrieved to publication on the social platform.
2. **Control** – Organizations need to understand that there is nearly an absolute lack of control over interactions with members of the social platform. In the real world we attempt to control the types of social interactions we experience by carefully choosing our social circles as well as the places we hang out. This is not possible in the cyber world. Comment spam, defamation, false claims and bad language are the norm.

   Keeping your social cyber environment clean of these is a difficult task. Further, cyber cleansing claims resources in a manner proportional to the popularity of the enterprise. Measures range from sifting and sanitizing comments to engaging closely with the social networks in case of defamation. Enterprises who fail to invest these resources will quickly find that true followers are fleeing the scene. In the meanwhile, the brand name erodes – defeating the purpose of entering the social network scene.
3. **Lack of Trust and Proper Identification** – There is no real way for enterprises to avoid copy-cats.

   In today's social platforms, there is no solid way to tell apart the real owner of a brand from impostors and copy-cats who are trying to take advantage of the popularity of a specific brand, to abuse it or to erode it. The identity of message posters cannot be verified in any way and there are no real tools to evaluate the trustworthiness of messages and their content.

The consequences could be general brand erosion or attack campaigns targeted towards enterprise's social circle. Mix these three concepts with the growing use of automation and you get social network mayhem. In the past couple of years we have witnessed the impact of the power of automation when applied to social networks:

- In February 2011, the Lovely-Faces.com website showcased hundreds of thousands of scraped Facebook user profiles.
- In September 2011, another group demonstrated an application that automates the process of "friending". Based on this process, the application creates a collection of all personal information, including photos, from those who accepted the friendship request.
- Recently a group of researchers demonstrated the power of "social botnets". These are fake profiles. However, these accounts can automatically grow a network of friends of actual real accounts. The research proved that the flawed "friend of a friend" trust model enabled this type of botnet proliferation. Further, their research found that individuals were three times more receptive to accepting a friendship request if the requester already shared a mutual friend with them.
- Software automating account generation and various data mining research projects exist.
- This Fall, DHS started setting up policies to monitor Facebook and Twitter. Automating this process will be at heart of this project in order to sift through the incredibly high volume of traffic.

Unfortunately, we do not see any market solutions ready to handle the above issues. Facebook as well as other social media platform providers are currently keeping full control and are attempting to fight some of the issues (mainly automation and fake accounts) from within. One such initiative is Facebook's Immune project. This has proven to be mostly futile so far (for instance, there's a clear conflict of interests between Facebook's attempt to remove fake accounts and its attempt to show constant unbelievable growth). Rather, the solutions must be incorporated into existing platforms by enterprises themselves.

These solutions will have to rely on third parties that offer trust and data control services over the social media platform. Currently, we are not aware of any such existing solutions, leaving a void space ripe for research.

## Trend #2: The Rise of the Middle Man

In 2010, we predicted the industrialization of hacking. What is the impact of industrialization to hacker's business models? In 2012, with the increased supply and demand for compromised machines, as well as for sensitive corporate info, we predict the rise to a new cyber crime job role: the broker. This individual is responsible to match the buyers of stolen data, or compromised machines (aka "bots"), with the sellers of the data (or bot renters). In the same way stocks and investors gave rise to stock markets, hackers need a middle man.

The success of bot herding opened up a large market where lots of hackers have many corporate machines under their control, each potentially holding a vast amount of data. However, waiting for individuals to approach and buy this type of data from them is simply too much of a slow and ineffective approach – causing the hackers to be a victim of their own success. Instead, we are seeing that this situation actually opens up the wholesale opportunity for a middleman to bridge this gap.

## Trend #1: Security (Finally) Trumps Compliance

In 2012 we expect to see security decisions driven not by compliance but for the simple reason of...security.

It sounds simple enough, but in previous years we have seen the influx of laws and regulations which drove the budget and security solutions. PCI, SOx and world-wide Data Privacy Acts were all used as the reasons to feed the security budget. But this approach often backfired. Anecdotally, when one CIO was asked about the key lesson from a major breach his firm experienced answered, "Security is not about surviving the audit."

Smart companies used these regulations as springboards to enforce the case of security. In fact, both a 2011 Ponemon survey and the 2010 Verizon Data Breach Report showed that PCI did improve the organization's security stance. However, regulatory compliance is not equivalent and does not confer security. It is enough to turn to Heartland Payment Systems for such an example. The company passed its PCI evaluation, and yet, they had suffered one of the biggest breaches in history.

This past year we have seen a shift in the corporate attitude for several reasons:

1. **Breaches are costly**. Security breaches such as those suffered by Epsilon, RSA and Sony dominated front page news. The high profile breaches highlighted the impact of security. Brand damage, loss in brand, legal costs, notification costs, service outages and loss in shareholder value all became news of the day. In fact, the day after Sony's breach announcement, the stock price dropped steeply. DigiNotar, a CA company was breached in September (see SSL trend) went underbelly later that month. While actual assessments of the cost of these past year breaches have not yet been made public, we can return to the Heartland Payment Systems breach for a lesson. For nearly two years financial analysts watched as large legal payments for damages were settled before the market could feel comfortable about Heartland's ability to stabilize revenues.
2. **Companies with an online presence, regardless of size, are targeted**. Not only were large corporations affected by breaches in the past year. Hackers have become very adept at automating attacks. According to the 2011 Verizon Data Breach Investigation Repot, hackers have "created economies of scale by refining standardized, automated, and highly repeatable attacks directed at smaller, vulnerable, and largely homogenous targets". In other words, in a world of automated attacks, everyone is – or will be – a target. This point was exemplified in August 2011 when *USA Today* published that 8 million websites were infected by malware. Our own research highlights how applications are likely to be probed once every two minutes and attacked seven times a second.
3. **Hacktivism brings (in)security to the frontlines**. Hacking groups such as Anonymous and Lulzsec have received headlines when they repeatedly hacked into different corporations, large and small. Visa, Paypal, Sony Pictures, Fox.com, PBS.org as well as countries such as Tunisia, and government agencies such as Infragard all felt the hackitivist wrath whose attacks targeted applications and infrastructure.
4. **APT becomes an actual threat**. Advanced Persistent Threats (APT) attacks are sophisticated attacks which relentlessly target corporations and governments for espionage and destruction. However, with good branding from worldwide Marketing and PR teams, this term has become the alternative description to a compromise following a corporate-phishing attack. The fear of such an attack is boosting the security budget. A recent survey by ESG indicated that due to APT concerns, 32% of respondents are increasing security spending by 6-10%.
5. **Intellectual property requires protection**. Organizations are beginning to understand the risk and consequences of a compromise of their bread and butter. The biggest risk of exposure of intellectual property is actually caused unintentionally. For example, through an employee leaving the company with corporate info obtained rightfully over time. Or, through a mis-configured server holding confidential documents (see trends on the externalization of collaboration platforms). Organizations also face the risk the deliberate theft of data from vengeful or malicious employees. For instance, this past year a former Goldman Sachs employee received an eight year sentence for stealing proprietary software code. Compromise of intellectual property may even be performed by the hands of external hackers. In the past we saw how hackers were solely focused on credit card numbers, login credentials and other such generic commodities. Although this type of data is still on the attacker's radar, we are starting to see hackers focusing also on intellectual property. As a point in case, consider the RSA attack which involved the data relating to the SecureID tokens.

6. **Shareholders are now involved**. The SEC has recognized the impact of a security breach to a company. As a result, recent updated SEC regulations require reporting information security breaches to shareholders. If in the past breaches could have been swept under the carpet, this regulation will make it harder to do so.

For these reasons, we will increasingly see how companies will perform wise security decisions based on actual security reasoning. Furthermore, the abundance of regulations – which ultimately try to set a minimal bar of security – will make it too costly for organizations to handle on a regulation-by-regulation basis. Instead, enterprises will implement security and then assess whether they have done enough in the context of each regulation.

## Conclusion

How did we come up with these trends? There were several factors:

- **Hackers** – As a part of Imperva's hacker intelligence initiative, we monitor hackers to understand many of the technical and business aspects of hacking. The insights provided from our investigations help us see what hackers are doing or in this case, plan to do. In some cases, hackers make small tweaks to existing attacks or come up altogether new ones.
- **The good guys** – Many of our customers are smart, really smart. We meet with them regularly to understand their challenges and concerns to understand emerging trends.
- **Weather balloons** – We monitor traffic in cyberspace. This helps us understand statistically how hackers may be shifting focus regarding attacks.
- **Intuition** – Many in the ADC have been in security for many years in the private sector, the military and academia. We've seen a lot in those years.

Our hope is to give security teams a comprehensive, substantive set of predictions to help you prioritize your security activities for the coming year. Be safe!

## Hacker Intelligence Initiative Overview

The Imperva Hacker Intelligence Initiative goes inside the cyber-underground and provides analysis of the trending hacking techniques and interesting attack campaigns from the past month. A part of Imperva's Application Defense Center research arm, the Hacker Intelligence Initiative (HII), is focused on tracking the latest trends in attacks, Web application security and cyber-crime business models with the goal of improving security controls and risk management processes.

**Imperva**
3400 Bridge Parkway, Suite 200
Redwood City, CA 94065

Tel: +1-650-345-9000
Fax: +1-650-345-9004
www.imperva.com