Web 2.0:
The New Face of the Web

…and why your protection may need a face-lift

Google™

Introduction

This is one of a series of white papers setting out considerations for the enterprise in relation to corporate use of the Internet and concerns itself with answering the following question:

"In the face of a Web 2.0 threat landscape, does my web security posture need to be re-examined?"

This paper addresses the complex implications, interactions and unique challenges of Web 2.0 and the malware that exploits it.

Evolution of Content

Prior to 2001, web sites were relatively static, designed to push information to users in a manner that was not interactive. But proving that adversity can be the path to enlightenment, following the dot-com crash in late 2001 a new, stronger Web emerged. And unlike its predecessor, the new Web lived up to its name – sites became sticky hubs of interactive content, constantly changing and morphing based on the wants and needs of its visitors. Today, the technology that enables Web 2.0 is merely the vehicle, the transport mechanism from point A to point B. It is the user – those members of the particular web community – who ultimately drives the destination.

Unfortunately, malicious software (malware) has also evolved. And just as technology has been replaced by users as the driving force behind web sites, the computer is no longer the ultimate target of the malware – it is the user that is the target. Today, malware is almost single-purposed: to gain access to the user's private, financial, and confidential information. To gain that access, malware authors exploit the very thing that makes Web 2.0 so successful – the user's trust.

Dynamic Delivery

Modern Web sites bear little resemblance to their predecessors. Today's websites feature dynamically changing content delivered through a steady stream of user contributions, RSS feeds and third-party advertising. Commerce is increasingly the goal, with a large portion of active sites engaged in affiliate relationships, direct sales, or some other form of monetary gain.

Not only is the face of the web changing, the number of web sites is sharply increasing. In mid-2005 when the term Web 2.0 was first coined, there were approximately 66.4 million sites according to Netcraft Web Server Survey[1] data. As of April 2008, that number had increased 250% to 165.7 million. Also in 2005, Antonio Gulli of the Università di Pisa and Alessio Signorini of the University of Iowa performed a study based on search engine indexing which discovered an estimated 11.5 billion pages[2]. In 2008, the estimated number of web pages is nearly 30 billion. This figure excludes archived data by the Internet Archive Way-back Machine; in 2008 the IAWM had grown to 86 billion archived pages.

Blogging and social networking comprise the largest segment of growth, a phenomenon also driven by widespread adoption of Web 2.0 technologies. The combined impact of all these factors leads to a situation in which:

- The number of Web sites is increasing;
- The amount of third-party content on those sites is increasing;
- The reliance on active scripting is increasing;
- Social interaction and user-supplied content is increasing; and
- The number of inexperienced Web developers is increasing.

Compounding all of these challenges is a more dangerous increase – the dramatic rise in both quantity and sophistication of new malware exploiting the Web 2.0 phenomenon. And that disturbing increase is coupled with a new motive: targeting the user for financial gain.

The Criminal Element

Within the software industry, or any viable industry for that matter, there exists research and development, quality assurance testing, sales, marketing, customer service and support. With money as the motive, today's malware authors maintain a similar infrastructure. Toolkits that detect and exploit vulnerabilities on web servers are widely available. Trojans sell openly on Internet back channels, and spam services are equally inexpensive and accessible.

Despite the similarities, there is one distinct difference between the malware market and a legitimate enterprise. With a legitimate enterprise, there is typically a traceable source of accountability. In the malware enterprise, the criminal actions are spread over a disparate, unconnected and anonymous tier of players.

For example, an attacker purchasing a password stealing Trojan may then contract with someone else for the use of an exploit tool such as MPack, purchase a list of stolen instant messaging (IM) or email addresses from a different source, and lease time on a network of compromised computers (a botnet) from yet another source.

Speaking at the 15th annual Defcon conference, Dr. Thomas J. Holt, computer criminologist and professor at the University of North Carolina, dissected the malware market[3]. According to Holt, the data stealing Pinch trojan sells for as low as $30 and the seller provides technical support. The package includes the buyer's choice of packer and is guaranteed to be undetectable by signature based scanners at the time of purchase. For an additional $5, buyers can get custom revisions. A $100 server statistics software package is also available, allowing the buyer to track the infections in the same manner that a legitimate company might track sales.

[1] http://news.netcraft.com/archives/web_server_survey.html
[2] The Indexable Web, http://www.cs.uiowa.edu/~asignori/web-size/
[3] The Market for Malware, Dr. Thomas J. Holt, presented at Defcon XV

The MPack exploit framework was first spotted (by PandaLabs) in December 2006, offered for license on a Russian forum. New and improved versions quickly followed and subsequently its adoption increased. This contributed heavily to a 26% increase in web-based malware in April 2007 followed by a 36 percent increase in May 2007.

This business side of malware introduces many new challenges:

- Attackers don't need coding skills, they simply need a relatively small amount of cash;
- Malware of all types is readily available;
- Buyers can expect fully tested, high-quality malware and technical support;
- The malware industry doesn't have an organizational hierarchy or traceable source of accountability;
- The number of new malware increased four-fold from 2005 to 2007 and is projected to increase ten-fold in 2008[4].

Ironically, it is Web 2.0 – the technology that saved the web from the dot.com bust – that facilitates the interaction, commerce, and trading that takes place among criminal coders today. And just as the attackers are using Web 2.0 technologies to facilitate the buying and selling of malware, they are also exploiting Web 2.0 technologies to foist that malware onto its victims.

Exploit Frameworks

As debates on the merits of full disclosure versus responsible disclosure were waged in security newsgroups and the media, a quiet evolution was taking place. A monetary market developed for newly discovered vulnerabilities and the question of disclosure, in many cases, became a moot point. In some cases, the purchasers of pricey zero-day exploits are the authors of exploit frameworks, server-based tools used to discover and exploit vulnerabilities on the systems that access those servers.

In the early stages of web-based malware, attackers enticed visitors to infected sites via spam and other social engineering campaigns. However, as wide adoption of Web 2.0 technologies increased, many of those technologies included either exploitable vulnerabilities or were implemented in an insecure manner ripe for compromise. To increase their return on investment, attackers began exploiting these weak points, compromising legitimate and often highly-trafficked sites and outfitting them with malicious, hidden iFrames that automatically load the malware from the attacker's domain.

In and of itself, an iFrame is a standard part of HTML and allows a web developer to embed information from a source other than the page the visitor is viewing. The malicious iFrame is generally configured to display five pixels or less of display space to make it invisible to the user. Malicious iFrames silently pull a remotely located file named index.php which the browser opens. It then collects data about the visiting system which it sends back to the its server, which then delivers an exploit specific to software running on the user's system at the time of attack, making such exploits highly personalized attack mechanisms.

Today, over a dozen exploit frameworks are available, often for no or very little cost. Many are available for as little as $400 and the source code for other proven frameworks can be downloaded for free.

[4] http://www.kaspersky.com/news?id=207575629

During the course of the malicious ad run, Google estimates that up to 12 million ads may have been delivered, exposing a large number of users to the Trojan. Further, research has shown as much as 30 percent of vulnerable operating systems are insufficiently patched, leaving many of the exposed users open to infection.

Code Injection Attacks

The Structured Query Language (SQL) is used to access information contained within a database. In some cases, the web application may allow for the dynamic construction of queries based on user-supplied data. If the programmer has not properly managed the handling of user supplied queries, a code injection (and assorted other attacks) may be possible.

From October to November 2007, Google observed a SQL injection attack which resulted in malicious iFrames launching exploit code hosted on a single server. The attack targeted the combination of Active Server Pages (ASP) and Microsoft SQL Server.

A second attack, using a nearly identical technique, occurred in late December 2007, extending into January 2008. A popular home furnishings website was the first victim observed during the busy holiday season and infected over 50,000 web sites. In this second wave, the attacks typically impacted obscure pages that were not widely accessed by users. It was, however, more successful than the previous attack, signaling improvements to the attack methodology.

In early April 2008, the attackers further honed the SQL injection tool to bypass Chinese government websites to focus on English language pages and to better target pages that enjoyed high rankings in search engines. These improvements resulted in a dramatic increase in the success of the attacks, heavily contributing to a 35% increase in Google Web Security malware blocks for the month.

The Third Party Threat

It's an interesting conundrum – that which makes Web 2.0 so compelling and successful, also serves as its Achilles' heel. The interconnectivity and interactive nature of today's Web creates an environment in which third-party content is not only commonplace, it's the norm. As such, webmasters need not only be concerned with their own content and security, but also the content and security of each of their third-party providers. And even after a site has ceased to be active, it can – through third-party content and pre-established trust relationships – do harm.

In August 2007, Google detected malicious code originating from an ad server hosted on an anonymous IP address allocated to a German ISP. The malicious ads were predominantly appearing on "parked" sites – sites that have become inactive and then used to host ads using an external service. In the course of investigation, Google detected infected ads on 126 parked sites, one of which was a previously active hotel website. Links to the hotel site were found on other websites, including a major UK newspaper site. Users who clicked through legacy links on other legitimate sites were thus exposed to the risk of compromise, even though the domain itself was defunct.

Malicious third-party ad content can also impact fully functioning legitimate sites. Throughout much of August and a portion of September 2007, third-party ads infected with a downloader Trojan impacted many active, high profile sites, including major newspaper, social networking and photo sharing sites.

As discussed in Google technology partner ScanSafe's 2007 Global Threat Report, the number of malicious web events increased 61% from 1H07 to 2H07 and the amount of time a malicious website remained live increased 62% from 1H07 to 2H07. Additionally, on average 21% of all Google Web Security blocks were for zero-day threats and the amount of time a site hosting zero-day threats remained live increased from an average of 21 days in 1H07 to 61 days in 2H07.

In another attack, Google uncovered a multi-tiered rogue affiliate network that appeared to be boosting rankings for certain sites while simultaneous delivering malware via exploit. Three generic downloader Trojans, a password-stealing Trojan commonly referred to as Pakes, and a new variant of the Zhelatin family of Trojans (also know as the Storm worm) were uncovered in the course of the investigation.

Social Engineering

Web 2.0 has fostered community and interaction across all peoples of all nations, bound together through a common interest, pursuit, or need. The resulting community-based web sites result in tangible friendships with virtual strangers. But unlike what our mother's taught us, talking to strangers isn't what will get us in trouble. In the Web 2.0 world, things are much more complicated than that.

The MeSpam trojan is a personal example of social engineering. MeSpam retrieves messages and links from a remote server, and appends that information to forum posts, blog comments, and web mail correspondence from the infected user. The link, updatable via the master server, can be changed at will by the attacker, as can the actual text used in the message.

Web 2.0 communities can promote a feeling of trust between the respective members. If one person in that social community is compromised by MeSpam or similar exploits, they become unwitting and unwilling accomplices in attacks against other members of the same community. Thus if one participant in a community is felled by malware, the other members of that same community are now at heightened risk of compromise.

Getting to the Source

One hundred million web sites ago, it was relatively easy to shutdown an infected site. Security researchers initiated contact with the owner of the site that was compromised. In such cases, remediation was swift and the site owner appreciative. Failing that, the domain host could be contacted and, if that failed, the site could be blacklisted by web filtering products so those users protected by those filters would be protected. But that was one hundred million web sites ago.

Today, the problem of rogue sites is much more complex. Both the owner and the host of the malware delivery site may be located in geographic regions outside of the confines of legal jurisdiction. Further, the legitimate sites compromised in the attacks may be under the ownership of inexperienced web developers who fail to react when notified.

But even when all pieces fall into place, as soon as one site is shutdown, more spring up to take its place. The rising botnet population – large collections of infected computers under the control of attackers – ensures this ready supply of compromised machines.

Additionally, much of the dynamic content delivered on today's interactive web sites is no longer under the control of the site owner. And most often, the site itself is not hosted by the owner, thus even the most experienced web developer may not be able to ensure security from A to Z. Further, the more software and services required, the greater the exposure to un-patched and newly discovered vulnerabilities.

On average, 21% of the malware threats stopped in 2007 by Google Web Security were zero-day attacks for which signatures were not yet available.

The increased popularity of Web 2.0 introduces new challenges:

- Jurisdictions may impede timely shutdown of rogue web sites;
- Inexperienced web site owners may fail to react even when notified;
- Sites that are shutdown are quickly replaced by new ones;
- Site owners typically have little control over all the content on their sites;
- Sites are more complex and thus more vulnerable to exploit.

These challenges exacerbate the severity of today's malware, making it more difficult for both traditional security firms and law enforcement to counteract.

Blacklisting – Policy Enforcement, Not Security Protection

The equivalent of the virtual bouncer, blacklisting is done by compiling lists of known bad URLs and blocking access to the included sites. Originally compiled by collecting reports of known bad sites, blacklists today are generally created by a process known as 'crawling' or 'mining'. This consists of scouring through lists of URLs, following links, scanning sites and blacklisting any sites found to be harboring malicious code.

Blacklisting, while useful for policy enforcement and managing Internet use, is not effective as a security technology in the current Web 2.0 environment. The chief drawback, is that content on websites is constantly changing. A scan for malware even five minutes in the past is no indication of the status of the site at the time of access. Further, blacklisting can block a legitimate site that was temporarily compromised, subsequently cleaned, and no longer poses any risk to users. As such, blacklisting is too reactive with previously compromised sites, and not reactive enough with new sites that are compromised.

Additionally, the large amount of time required to crawl the web means that only a small fraction of pages can be crawled. If even 450 million pages are crawled, that represents only 1.5% of the total 30 billion pages on the web. In any event, past performance may not be indicative of future behavior – the user remains unprotected against whatever occurred between the last crawl and their current visit.

Google Web Security for Enterprise

Google Web Security, provided as Software as a Service (SaaS), scans inbound and outbound web traffic in real-time at the moment of access, using multiple layers of zero-day threat detection combined with signature-based anti-malware scanners.

Uniquely positioned in the cloud, the service has unmatched visibility, analyzing several terabytes of web code each day and compiling the industry's most comprehensive data set that dates back to 2004.

A URL reputation engine examines multiple parameters such as IP address information, country of the web server, history and age of the URL, and other criteria to assess the reputation of the site.

LEARN MORE

www.google.com/a/help/intl/en/security/web.html

Google Web Security's traffic behavior engine analyzes network traffic patterns to identify suspicious, atypical traffic suggestive of malware. A code behavior engine determines the behavior of the code by modeling program logic, behavioral rules, and contextual parameters that taken together would suggest good or bad intentions.

The Google Web Security code reputation engine compares information such as type of code, history and age of the code, frequency of the code, file structure/header/content patterns, and program logic patterns to code that is known to be good or bad.

The multiple detection engines give their assessments of the code, and these assessments are then combined to produce a comprehensive view of whether or not the new code is malicious.

Conclusion

Web 2.0 technology has brought new levels of richness and interaction to the Internet experience. But is has also brought new levels of exploit and malware technology as well. New levels of technology are required to combat this trend to protect corporate networks and intellectual property. Google Web Security, delivered "in the cloud" as a service, protects organizations of all sizes against web malware attacks in real time and enables the safe, productive use of the web.

Google™