SECURE your network.

2010 Full Year Top Cyber Security Risks Report

In-depth analysis and attack data from HP DVLabs.

Contributors

Producing the Top Cyber Security Risk Report is a collaborative effort among our HP DVLabs, HP TippingPoint IPS, and other HP teams such as the Application Security Center. We would like to sincerely thank OSVDB for allowing print rights to their data in this report. For information on how you can support OSVDB:

https://osvdb.org/account/signup

http://osvdb.org/support

We would also like to thank Malware Intelligence for contributing to our Web Browser Toolkit section of the report.

http://www.malwareint.com/

Contributor	Title
Mike Dausin	Advanced Security Intelligence Team Lead
Marc Eisenbarth	DV Architect
Will Gragido	Product Line Manager, HP DVLabs
Adam Hils	Application Security Center Product Manager
Dan Holden	Director, HP DVLabs
Prajakta Jagdale	Web Security Research Group Lead
Jennifer Lake	Product Marketing, HP DVLabs
Mark Painter	Application Security Center Content Strategist
Alen Puzic	Advanced Security Intelligence Engineer

Overview

In the latest version of the Cyber Security Risks Report, the HP DVLabs team reviews the threat landscape for all of 2010. The report looks at the current threats targeting the enterprise as well as how these have evolved over the last year. The goal of this report is to arm enterprise IT, network and security administrators with information on the attacks targeting their data centers and networks, so that they can implement the necessary protections to maintain business function.

Key findings from the report include:

- **The number of discovered vulnerabilities has plateaued, but the number of attacks against known vulnerabilities continues to rise.** Data from the report indicates that the annual number of vulnerabilities being discovered in commercial computing systems has remained steady from 2009 to 2010. At the same time, targeted exploits that take advantage of these known vulnerabilities have continued to increase in both severity and frequency. This means that unpatched or unupdated systems are putting enterprise data centers at a huge risk for being compromised.
- **Web application vulnerabilities continue to be a gaping hole in enterprise security deployments.** Data from the report indicates that nearly half of all reported vulnerabilities exist in Web applications – meaning services that use the Web as the portal for users to access or interact with a piece of software. In this report, HP DVLabs takes a close look at the security of some of the most popular content management systems (CMS). The leading cause of vulnerabilities in a CMS are unpatched or poorly patched plug-ins rather than the core system. For the always online enterprise, poor patch management represents a large hole in the overall security of the organization.
- **Attacks are becoming more productized and marketable.** The report looks at Web exploit toolkits, which are essentially attack frameworks that can be bought, sold, or traded. HP DVLabs delves into the toolkits themselves to explain the sophistication of today's security exploits and how they compromise enterprise systems. The creation of security exploit toolkits follows similar processes as are used in the development of commercial software, resulting in extremely sophisticated and well thought-out attacks.

HP DVLabs compiled the report using data from a worldwide network of HP TippingPoint Intrusion Prevention Systems, vulnerability information from OSVDB and the Zero Day Initiative, security scan data from HP DVLabs, and Web application data from HP WebInspect.

Vulnerability Trends – 2010 Review

As in previous years, HP DVLabs has once again collected and analyzed a tremendous amount of data to identify significant vulnerability trends in 2010. The data and conclusions discussed below originate from:

- The Open Source Vulnerability Database (OSVDB), which is an independent source of detailed, current, and technical information on security vulnerabilities.
- The HP DVLabs team, the Zero Day Initiative (ZDI),—a program operated by HP DVLabs that rewards a global network of security researchers for responsibly disclosing vulnerabilities—and the HP Application Security Center.

The combination of these data sources gives HP DVLabs the unique ability to correlate vulnerability data from research-based endeavors as well as hands-on, tactical investigations, generating credible and relevant information that is immediately useful to today's IT security professionals.

Based on data from OSVDB, the number of vulnerabilities increased approximately 10% from 7,260 in 2009 to over 7,900 in 2010. While this increase is not welcome news to security professionals, the overall trend the past four years is still down, below the four-year average of roughly 8,500 vulnerabilities. Vulnerability disclosure seems to have hit a plateau. While the creation of new software typically produces new vulnerabilities, this is tempered by improved software development practices including fuzzing and QA. It is also possible that attackers are content with current vulnerabilities, and therefore do not invest as heavily in vulnerability research as they once did. HP DVLabs findings assert that vulnerability researchers, reverse engineers, and penetration testers discover or stumble upon vulnerabilities all the time. However, an attacker, such as a botnet operator, is not likely to invest in that type of research activity. For example, while Conficker and project Aurora utilized a zero-day vulnerability and Stuxnet utilized several zero-day vulnerabilities, the average botnet operator lacks the sophistication of the Conficker and Stuxnet attackers. It appears that a majority of attackers are content to utilize the list of known vulnerabilities accumulating year after year in widely used applications such as Web browsers, Web applications, social networking sites, Web 2.0 interfaces, as well as the associated plug-ins with all of these tools

The following chart (Figure 1) depicts year-over-year vulnerability disclosure, based on OSVDB data. The spike in 2006 is followed by a lower, two-year plateau, which again is followed by another lower plateau in 2009-2010.

Figure 1:

Year-Over-Year Vulnerability Disclosure Data

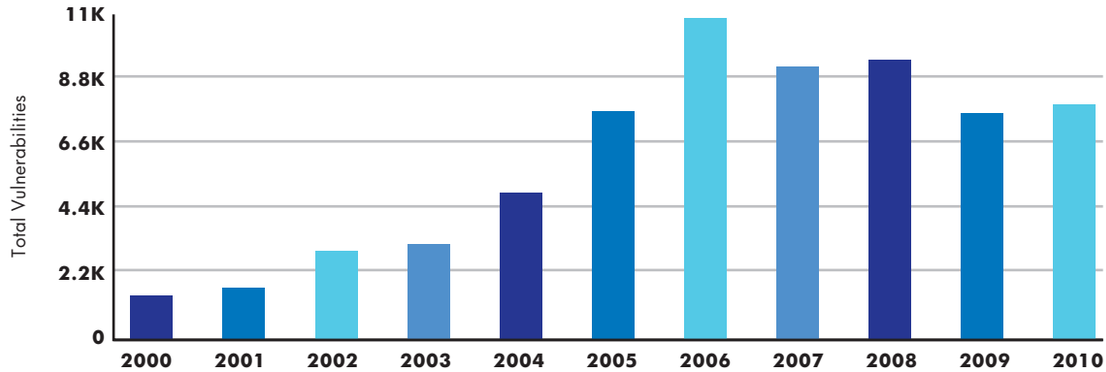
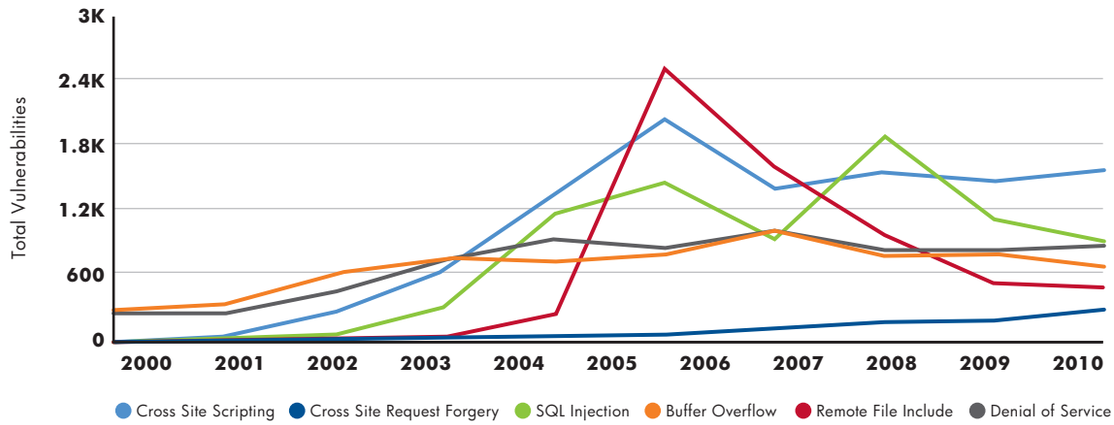


Figure 2:

Vulnerability Type by Year



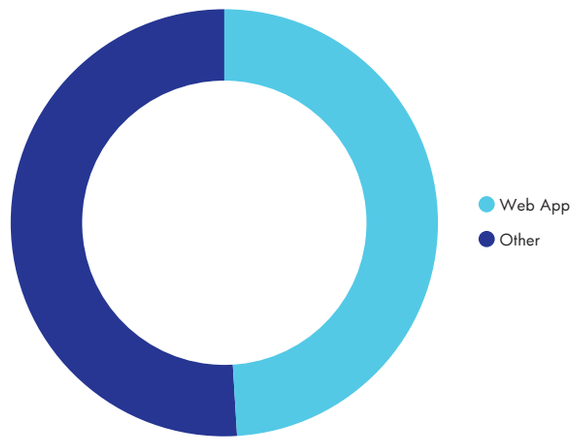
Looking more deeply into the types of vulnerabilities, the above graph (Figure 2), again from OSVDB, shows trend data about the more prevalent types, such as Cross-Site Scripting and SQL Injection. The period from 2006 to the present time seems to define the modern era of the vulnerability landscape, with an equal share originated in Web applications as are originated in traditional targets such as operating systems and legacy services like SMB. The data also indicates lifecycles with peaks, valleys, ebbs, and flows in the number of disclosed vulnerabilities. For example, PHP file-include vulnerabilities peaked in 2006, SQL Injection peaked in 2008, and Cross-Site Reference Forgery (CSRF) is ebbing slowly higher in recent years.

Vulnerability Trends - Web Applications

Web applications have continued to dominate the threat landscape in 2010, sustaining a steadily increasing trend over the last few years. The staggering number of Web application vulnerabilities combined with more effective exploitation methods (see section on Web exploit toolkits) demonstrates why attackers continue to target these systems. As shown in the following chart (Figure 3), Web application vulnerabilities comprise nearly half of all vulnerabilities.

Figure 3:

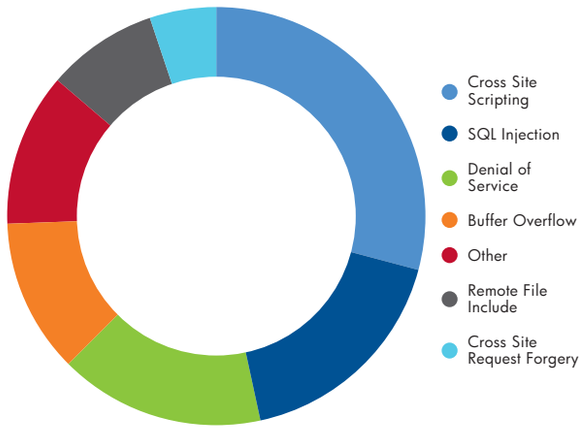
Web App Vuln Disclosure v All Vuln Disclosure, OSVDB 2010



Delving into the various Web application vulnerabilities reveals that Cross-Site Scripting (XSS) still comprises the most significant number of disclosed vulnerabilities, followed by SQL injection, and then Denial of Service (DoS). This is demonstrated in the chart in Figure 4. SQL Injection remains a popular option for database theft and drive-by SQL Injection by botnets. The ASPROX botnet overwrites portions of a compromised website's database to insert IFRAMES, which redirects website visitors to a malicious URL that infects the visitor's computer with malware, thereby adding it to the legions of zombie computers that make up the botnet.

Figure 4:

Web App Vuln Disclosure v All Vuln Disclosure, OSVDB 2010



Up until now this report focused on vulnerability disclosure, which may or may not reflect the complete picture of vulnerability trends unfolding on the Internet. In an effort to get a clearer picture of the real world vulnerability landscape, the HP Application Security Center (ASC) has compiled results from over 100 security assessments performed against a variety of customer Web applications. The ASC team took a high-level snapshot approach, testing the applications for a cross-section of common vulnerabilities.

Of the surveyed applications, an amazingly high 71% suffered from a command execution, SQL Injection, or Cross-Site Scripting vulnerability. It is important to note that any application that suffers from one of these types of vulnerabilities would fail a PCI compliance audit. Another 49% of the applications had at least one critical command execution or SQL injection vulnerability -- either one of which could allow a knowledgeable and determined attacker to completely compromise the system. Though small in comparison yet still disconcerting, 22% of the security-assessed applications were vulnerable to both SQL Injection and Cross-Site Scripting attacks.

The assessment determined that Cross-Site Scripting existed in not only the highest percentage of applications, but also in the greatest quantity across all assessed systems. A minor positive note is that eleven of the application assessment scans returned no vulnerabilities in these categories.

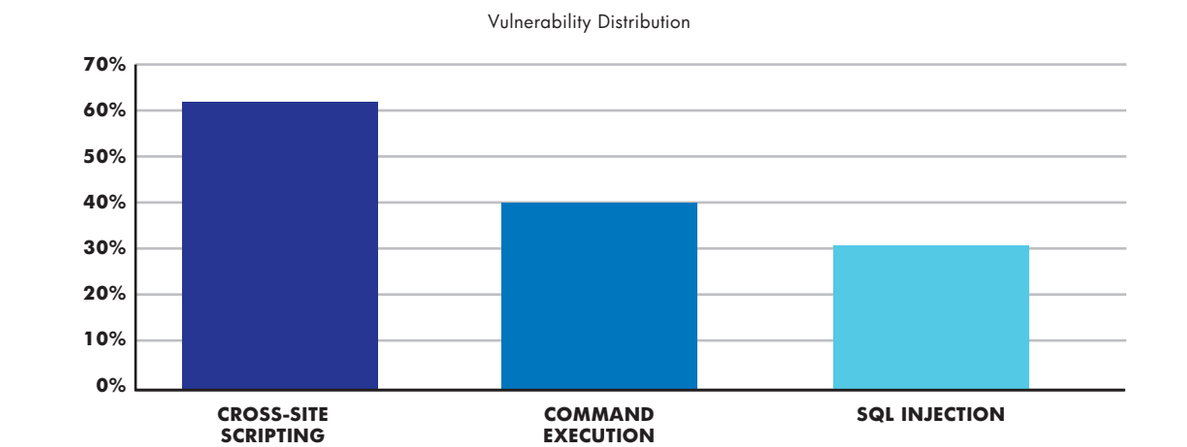
The following chart (Figure 5) displays the overall statistics, broken down by percentage. Each percentage reflects how many sample applications were susceptible to the vulnerability labeled on the horizontal axis.

Under the right circumstances, those could possibly lead to a complete system compromise. Twenty-two percent of applications were vulnerable to both SQL Injection and Cross-Site Scripting.

Here's how the overall statistics break down by percentage. Each percentage reflects how many of our sample applications were susceptible to that specific type of vulnerability.

Figure 5:

Percentage of Attacks in Web Applications Sampled



As Web 2.0 technologies such as AJAX, Flash, and HTML 5 enable organizations to create richer, more complex Web applications, vulnerabilities become more prevalent and more challenging to detect. The numbers listed above are concerning, but not surprising. To mitigate risk responsibly, organizations should test code in development, scan for vulnerabilities in QA before staging, and test applications in production on an ongoing basis.

HP DVLabs has delved further into the assessment of Web applications by performing in-depth analysis of Internet-hosted websites. It has investigated common open-source applications such as Wordpress, Joomla, and Drupal, each a type of content management system (CMS) commonly used for hosting blogs and online discussion groups. The investigation revealed an interesting differentiation between the core application and application plug-ins.

Figure 6 shows the percent of vulnerabilities reported in core application and in application plug-ins, from 2006 through 2009. For all CMS applications, OSVDB shows that the majority of vulnerabilities occur in the core application. This data is slightly misleading due to the large number of distinct CMS applications. When HP DVLabs focused on the three most popular applications, Wordpress matched the percentage shown by the total CMS population, while both Joomla and Drupal exhibited an astonishingly high percent of vulnerabilities in plug-ins.

Figure 6:

CMS Vulnerabilities 2006 - 2009

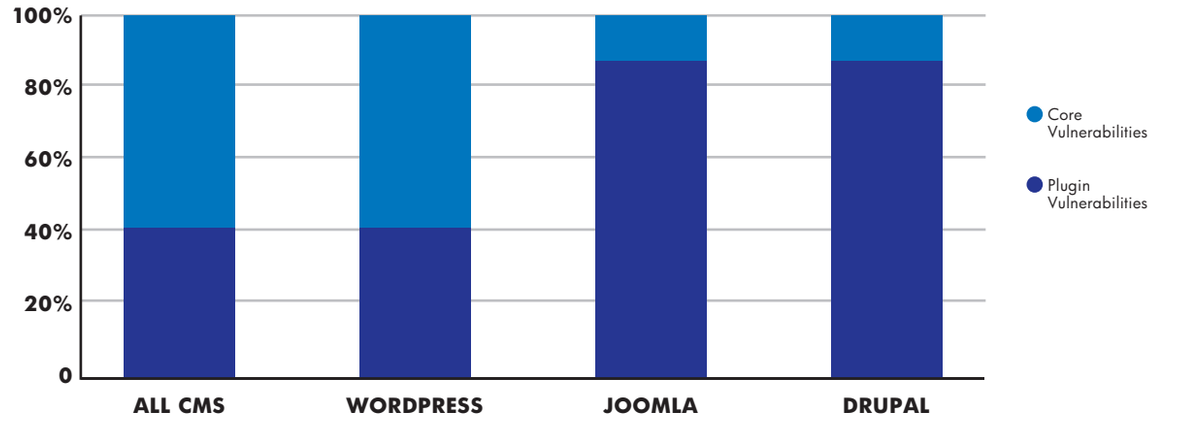
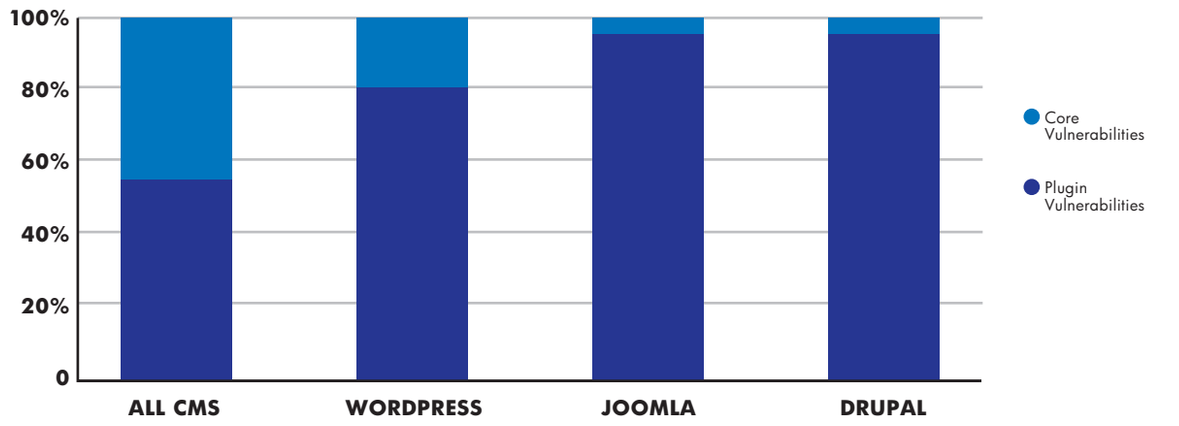


Figure 7:

CMS Vulnerabilities 2010

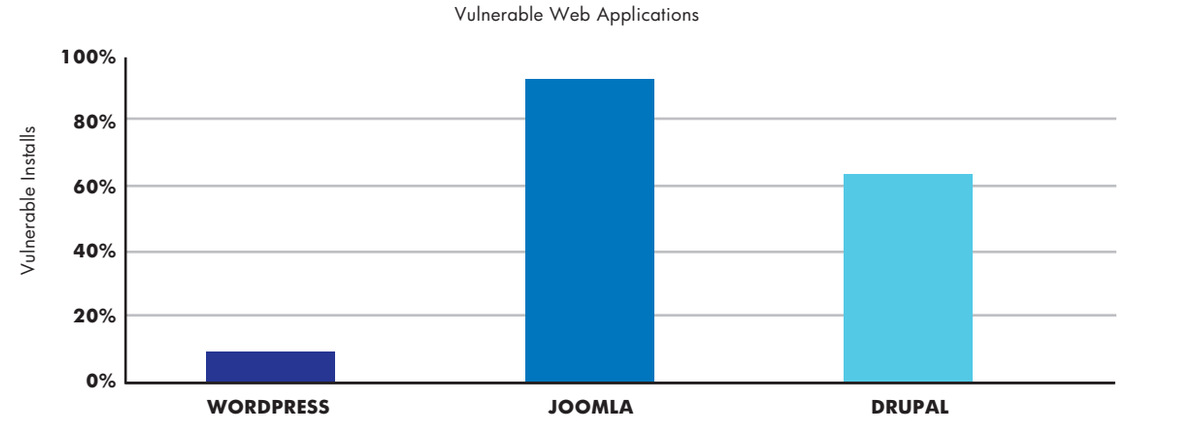


When viewing statistics solely from the year 2010, the results differ slightly (Figure 7). While the ratio for the entire CMS population remains similar to the multi-year trend, the ratio for the popular CMS applications skews even more heavily towards plug-ins being the source of vulnerabilities. A possible explanation might be increased diligence taken by the core application developers following a number of high-profile exploits against their platforms, thereby reducing the number of vulnerabilities in the core application and increasing the percentage of them in plug-ins. Further, plug-in developers may not place as much emphasis on security as those developing core applications, and may therefore be less concerned with locating and patching vulnerabilities.

HP DVLabs built a system to track websites running common Web applications, such as the CMS applications. A survey of the entire IP space of the Internet determined that there are approximately 104 million active hosts, of which at least 9.2% are running Wordpress, Joomla, or Drupal. Many of the installations featured one or more plug-ins to the core application.

Of the 9.2% of active hosts, HP DVLabs took a sampling of approximately one million hosts to perform more detailed analysis. Analysis of this data showed that patch rates in open source software seem to lag behind in Asian countries and in many of the largest global Internet Service Providers (ISPs). Low patch rates of commercial software—such as Microsoft products—in Asian countries have been widely publicized and are frequently attributed to piracy of such software. However, the investigation revealed that this trend of low patch rates exists not just in commercial products but in open source products as well. The trend of low patch rates at ISPs indicates that ISPs are typically reactive to security incidents rather than proactive in following the guidance of security vulnerability announcements. The reasons for this is unknown, however because customer uptime is so important for ISPs, they likely weigh the possibility of application instability introduced by a new patch against the likelihood that a vulnerability will actually be exploited in the real world.

Figure 8:
Vulnerable Web Applications



In the chart above (Figure 8), HP DVLabs demonstrates why patching is extremely critical in Web applications and their associated plug-ins.

The prevalence of vulnerable Web applications on the Internet is staggering. With so many potential targets available to be exploited, it is no wonder the Internet succumbs to massive numbers of SQL Injection and PHP file-include attacks, and data breaches continue to occur unabated.

Vulnerability Trends - Zero Day Initiative

The Zero Day Initiative (ZDI), founded by HP DVLabs in 2005, is a program for rewarding security researchers for responsibly disclosing vulnerabilities. The program is designed such that researchers provide HP DVLabs with exclusive information about previously unpatched vulnerabilities they have discovered. HP DVLabs validates the issue and works with the affected vendor until the vulnerability is patched.

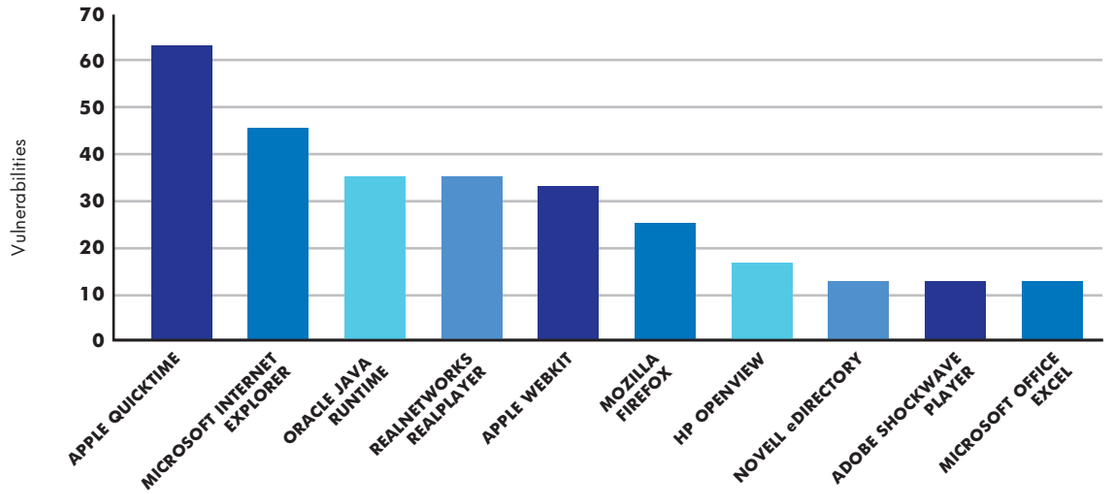
This program provides HP DVLabs with a unique set of data about new security research as well as information about the patch cycle for vendors. This information is then used by HP DVLabs to create filters that are deployed to the HP TippingPoint IPS.

The large market for client-side applications, as well as easier access to reverse engineering tools, has spurred significant interest in security research and vulnerability discovery. Researchers around the world seem to be growing in number, and many are interested in a responsible way of helping software vendors improve their products while still being compensated for their time and effort. Most of the discoveries are made with fuzzers whose sophistication has grown substantially due to new research over the past few years.

While the number of vulnerabilities publicly disclosed seems to have leveled out over the last five years, the ZDI program has risen in popularity and has purchased and disclosed many more vulnerabilities year after year. Between 2005-2010, HP DVLabs and the ZDI purchased and disclosed 750 previously unknown vulnerabilities, most of which were of high or critical nature in popular products used across both large enterprises and the average user.

Figure 9:

Top 10 Vulnerabilities Disclosed through ZDI From All Time (2005 - 2010)



In the table above (Figure 9), you can see the top ten applications with vulnerabilities disclosed through the ZDI. Eight out of the ten are related to popular client side applications with seven of those being related in one way or another to Web browsers.

Focusing solely on the year 2010 (Figure 10), HP DVLabs and the ZDI either discovered or acquired, and disclosed to affected vendors, 320 vulnerabilities in a wide range of products. Below you can see the top ten vulnerabilities disclosed through the ZDI in 2010, the majority of which are client-side related. Seven of the ten are related in one way or another to Web browsers.

Figure 10:

Top 10 Vulnerabilities Disclosed through ZDI in 2010

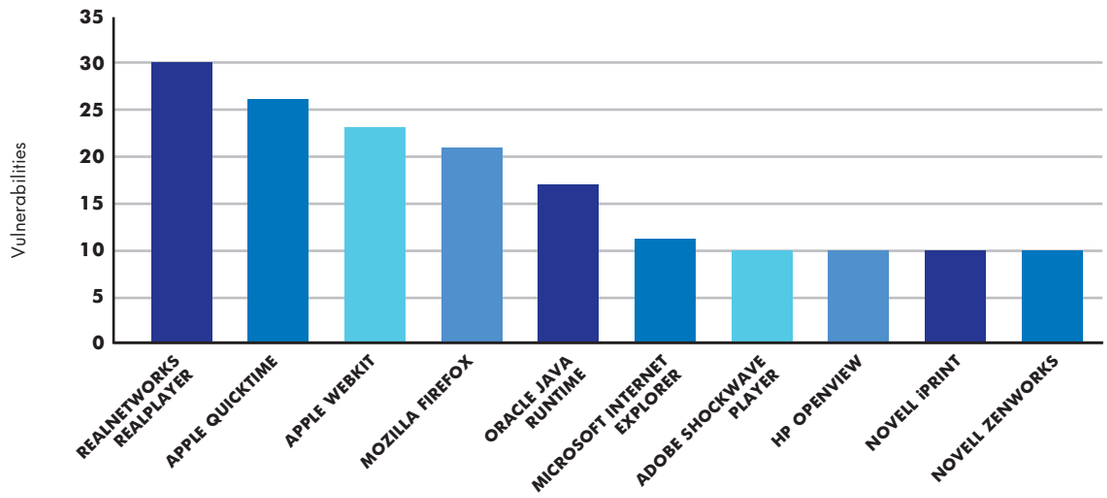
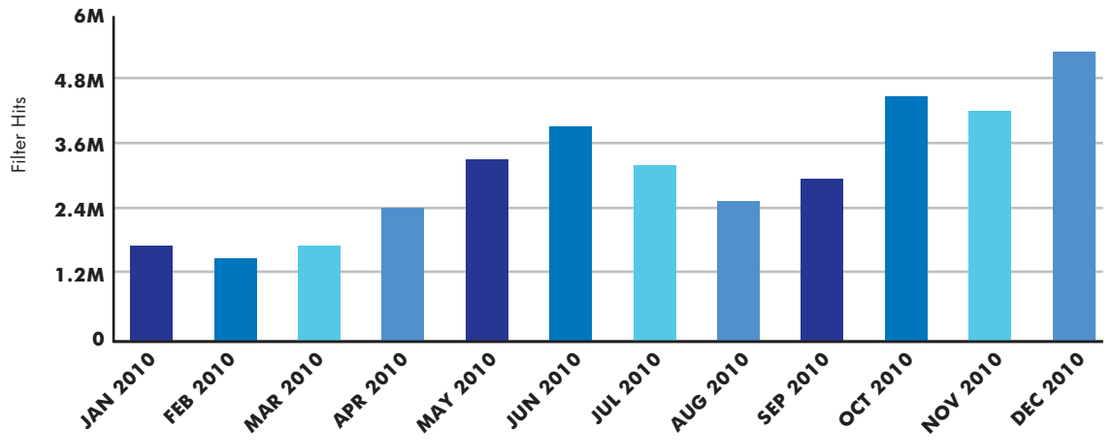


Figure 11:

Client-Side Attacks, Based on HP TippingPoint IPS Filter Hits



Attack Trends - HTTP Client versus Server Side

Both HTTP client-side attacks and HTTP server-side attacks saw a significant increase over the course of the 2010 sample period. The bulk of attack types are malicious JavaScript and PHP file-include attacks.

The chart above (Figure 11) depicts the number of client-side attacks, by month throughout 2010. The highest number, in December 2010, reached approximately five million attacks.

The following chart (Figure 12) depicts the number of server-side attacks, by month throughout 2010. They are much more prevalent than client-side attacks, with the highest number reaching about 23 million attacks in July 2010, which is almost five times more than the peak amount client-side attacks.

Figure 12:

Server-Side Attacks Based on HP TippingPoint IPS Filter Hits

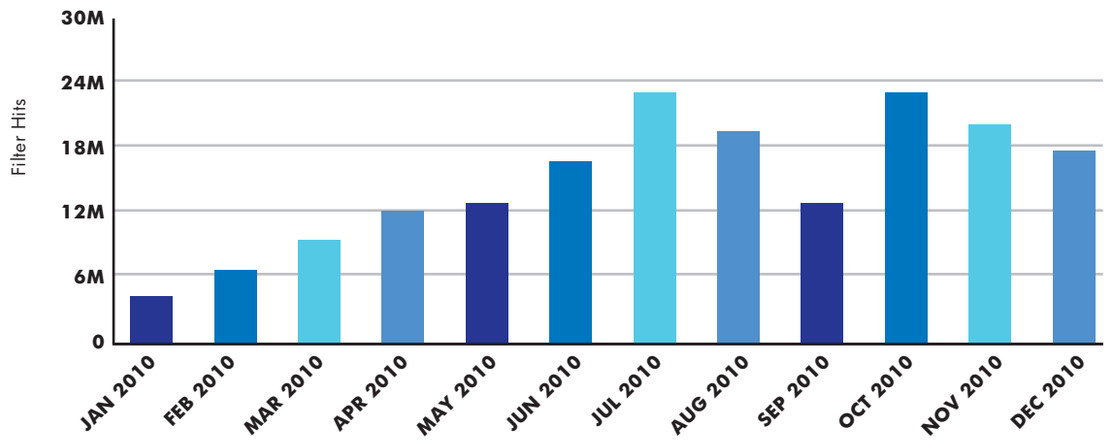
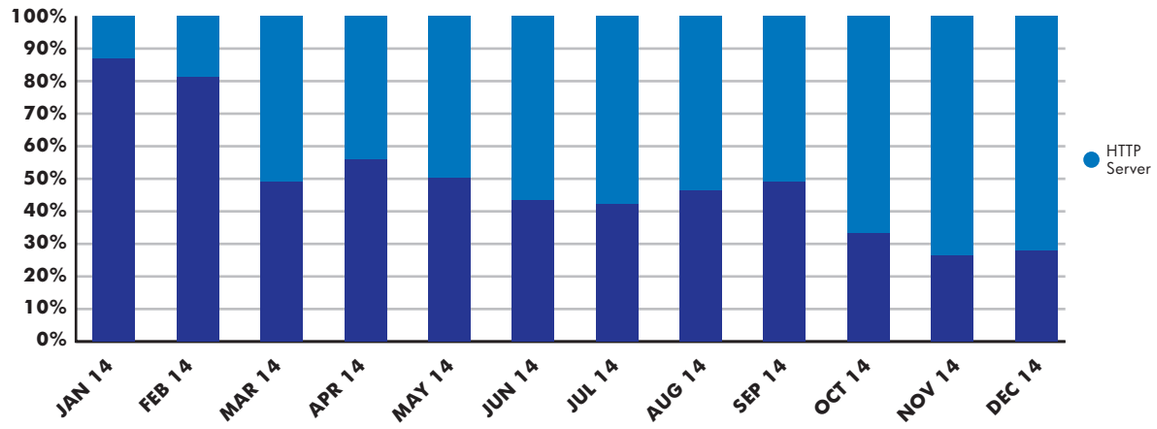


Figure 13:
SMB and HTTP Attacks



Recall that the vulnerability discussion focused on the increasing presence of Web application vulnerabilities, reaching nearly 50% of overall vulnerabilities, while traditional vulnerabilities diminished. Attack data pulled from HP TippingPoint IPS devices correlates with the vulnerability data from OSVDB and the ZDI. The above chart (Figure 13) shows an almost 60% shift from a legacy (i.e. SMB) type attack, towards an HTTP-based attack, over the course of only 12 months. HP DVLabs expects this trend to continue as more and more functionality is moved onto the Web and away from legacy services such as SMB.

- One more important point should be made about SMB and HTTP-based attacks. Nearly 100% of the observed attacks are automated, botnet, or worm-based attacks. Very few appear to be targeted against a specific machine or host. This is a completely different attack pattern than we see with HTTP. While the majority of HTTP traffic does also appear to be automated, much of it appears targeted towards specific hosts. A common HTTP attack pattern involves an attacker focusing multiple types of attacks to find a way into a vulnerable website. In contrast, the vast majority of SMB attacks are worm-based traffic. Anecdotally, the following list depicts the wide variety of attacks used against a host system that has fallen victim to a PHP file-include attack, as uncovered by an HP DVLabs investigation:

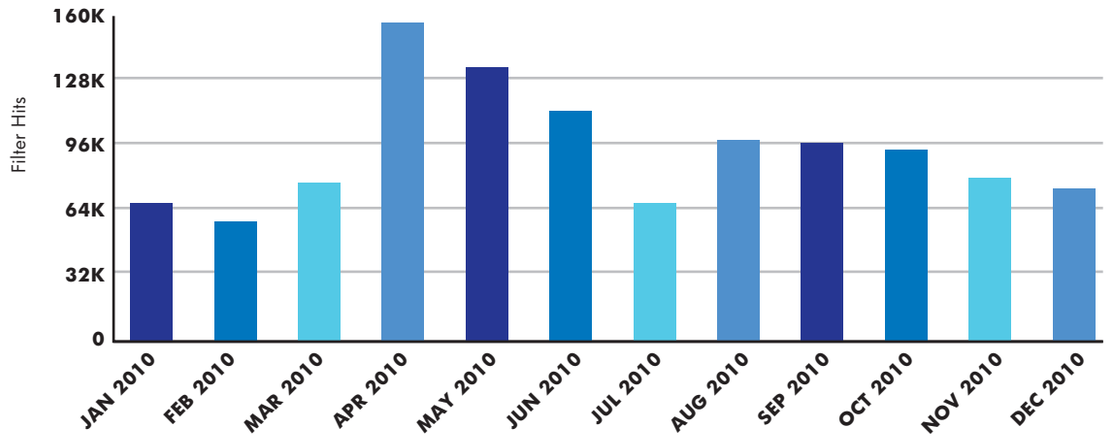
- Invalid TCP Traffic: Possible nmap Scan (No Flags)
- HTTP: HTTP CONNECT TCP Tunnel to SMTP port
- HTTP: AWStats Multiple Vulnerabilities
- HTTP: Paros Proxy HTTP Request
- HTTP: PHP File Include Exploit
- HTTP: Horde Help Viewer PHP Command Injection
- HTTP: PHP File Include Exploit
- SSH: SSH Login Attempt
- HTTP: Wget Web Page Retrieval Attempt
- HTTP: PUT Method Execution over HTTP/WebDAV

In great contrast to large number of HTTP-based attacks targeted against a victim host, the typical profile of an SMB attack includes a single type of attack, shown below:

- MS-RPC: Microsoft Server Service Buffer Overflow

Figure 14:

Malicious Javascript Attacks



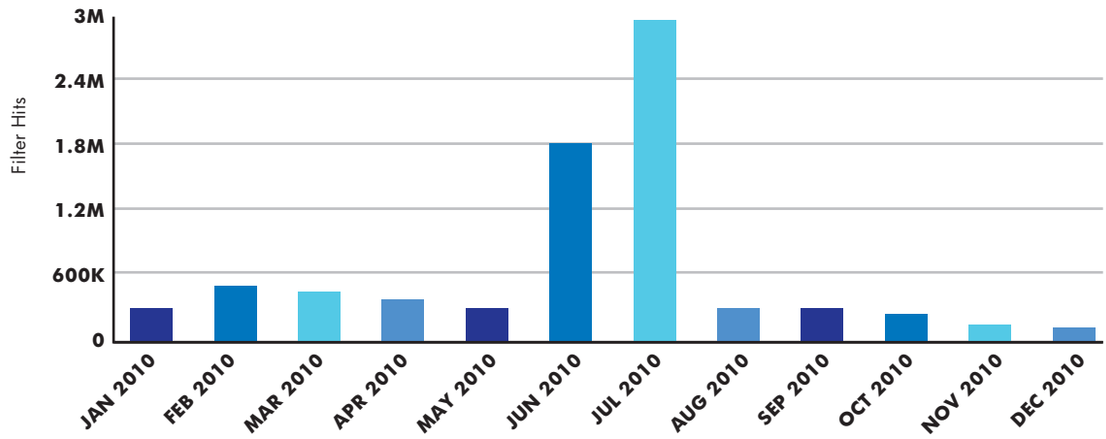
Attack Trends - Malicious JavaScript

Malicious JavaScript continues to be a popular attack type. It is considered to be one style of attack within the category of HTTP client-side attacks. Malicious JavaScript attacks are often highly obfuscated, and are specifically designed to bypass security controls. HP DVLabs accumulates statistics, such as those shown in the above graph (Figure 14), through the use of vulnerability filters operating in HP TippingPoint IPS devices. Throughout 2010, these types of attacks averaged about 90,000 per month, far lower than the overall HTTP client-side average of 1.8 million per month.

Attack Trends - PHP Remote File Include

PHP Remote file-include attacks saw a steady overall downward trend, except for a massive spike in mid-year (Figure 15). This is the nature of such attacks. They commonly compromise otherwise legitimate websites, which grants the attacker a window of opportunity to launch a widespread file-include campaign. Reputation-based detection models are designed to detect infected hosts and then add them to an Internet blacklist, thereby shunning them from interacting with the rest of the Internet. However, because file-include campaigns exploit legitimate websites, the reputation-based models sometimes lag in their detection of the infected websites. It is this window of opportunity that likely allowed the two-month spike in June and July of 2010.

Figure 15:



Attack Trends - Botnets

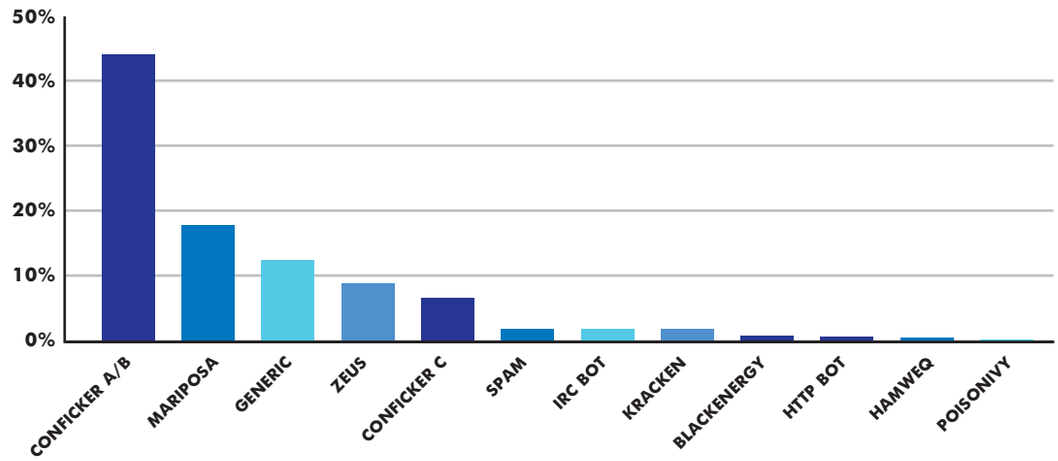
Botnets remained a huge problem in 2010. Overall, HP DVLabs tracks approximately ten million infected hosts. Amazingly, Conficker is still the most prevalent botnet, even though it was first detected in 2008. Its presence on the Internet is more than twice as much as the next most prevalent botnet, Mariposa.

HP DVLabs tracks activity for a number of botnets. The accumulated data is not only used to track the behaviors and prevalence of botnet families, but also contributes to the HP TippingPoint Reputation Digital Vaccine (ReputationDV) service, which evaluates the botnets in order to designate infected hosts as candidates for blacklisting.

The following graph (Figure 16) details the relative percentage of unique botnet drones detected, per botnet family.

Figure 16:

Numbers of Botnet Drones Per Family



Attack Trends - Denial of Service(DoS) and Distributed Denial of Service (DDoS)

Denial of Service (DoS) and Distributed Denial of Service (DDoS): Historic Review

Denial of Service (DoS) and Distributed Denial of Service (DDoS) fall into a category of Internet-based attacks that enjoy a rich and mature pedigree. The Internet threat landscape has been ravished by these attacks time and time again, and though they are considered to be a violation of the Internet Architecture Board's Internet Proper Use Policy, little is done by the Internet Engineering Task Force (IETF) to adjudicate said bad behavior. The goal of these attacks is quite simple: to deliver, in a concerted fashion, an attack of various denominations that prevents websites or services from functioning efficiently or at all. The disruption could be temporary or, as in the case of the ill-fated Blue Security 1, indefinite. The burden of addressing these attacks falls squarely upon data communications providers (traditional carriers, broadband providers, etc.), enterprise businesses, and individuals. The effectiveness of DDoS attacks, along with their ability to generate news and media coverage, is unparalleled. Recent examples have included:

Retaliatory DDoS attacks against Visa, MasterCard, PayPal, Bank of America, 4chan, and others as a sign of civil protest related to the WikiLeaks campaign. A similar attack was launched against the International Federation of Phonographic Industry (IFPI) in retaliation for the failed appeal of The Pirate Bay. In both cases, the hacktivist group 'Anonymous' used the Low Orbit Ion Cannon (LOIC) attack to cripple the targeted websites.

Retaliatory attacks were carried out by the citizens of Turkey as a protest against the state's decision to block Internet content and service.

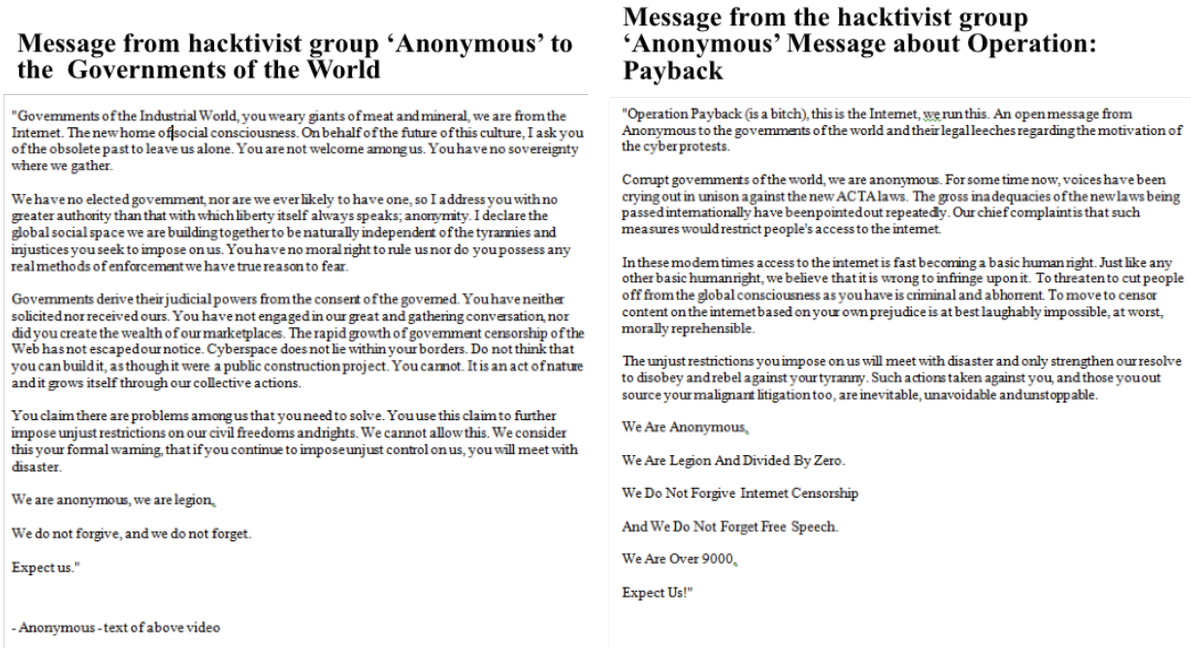
A DDoS attack targeted Ubisoft, creators of online video game 'Assassin's Creed,' rendering one of Ubisoft's new DRM servers inaccessible and other aspects of the game inoperable.

Cyber actors who make use of these attacks represent a diverse demographic profile. Many fall into one or more of the following:

- Miscreants
- Hacktivists (4chan, Anonymous, Citizenry)
- Cyber criminals
- Nation states
- Self inflicted [2]

Figure 17:

Messages by the Hacktivist Group 'Anonymous'



Message from hacktivist group 'Anonymous' to the Governments of the World

"Governments of the Industrial World, you weary giants of meat and mineral, we are from the Internet. The new home of social consciousness. On behalf of the future of this culture, I ask you of the obsolete past to leave us alone. You are not welcome among us. You have no sovereignty where we gather.

We have no elected government, nor are we ever likely to have one, so I address you with no greater authority than that with which liberty itself always speaks; anonymity. I declare the global social space we are building together to be naturally independent of the tyrannies and injustices you seek to impose on us. You have no moral right to rule us nor do you possess any real methods of enforcement we have true reason to fear.

Governments derive their judicial powers from the consent of the governed. You have neither solicited nor received ours. You have not engaged in our great and gathering conversation, nor did you create the wealth of our marketplaces. The rapid growth of government censorship of the Web has not escaped our notice. Cyberspace does not lie within your borders. Do not think that you can build it, as though it were a public construction project. You cannot. It is an act of nature and it grows itself through our collective actions.

You claim there are problems among us that you need to solve. You use this claim to further impose unjust restrictions on our civil freedoms and rights. We cannot allow this. We consider this your formal warning, that if you continue to impose unjust control on us, you will meet with disaster.

We are anonymous, we are legion,

We do not forgive, and we do not forget.

Expect us."

- Anonymous - text of above video

Message from the hacktivist group 'Anonymous' Message about Operation: Payback

"Operation Payback (is a bitch), this is the Internet, we run this. An open message from Anonymous to the governments of the world and their legal leeches regarding the motivation of the cyber protests.

Corrupt governments of the world, we are anonymous. For some time now, voices have been crying out in unison against the new ACTA laws. The gross inadequacies of the new laws being passed internationally have been pointed out repeatedly. Our chief complaint is that such measures would restrict people's access to the internet.

In these modern times access to the internet is fast becoming a basic human right. Just like any other basic human right, we believe that it is wrong to infringe upon it. To threaten to cut people off from the global consciousness as you have is criminal and abhorrent. To move to censor content on the internet based on your own prejudice is at best laughably impossible, at worst, morally reprehensible.

The unjust restrictions you impose on us will meet with disaster and only strengthen our resolve to disobey and rebel against your tyranny. Such actions taken against you, and those you out source your malignant litigation too, are inevitable, unavoidable and unstoppable.

We Are Anonymous,

We Are Legion And Divided By Zero.

We Do Not Forgive Internet Censorship

And We Do Not Forget Free Speech.

We Are Over 9000,

Expect Us!"

[1] http://www.wired.com/wired/archive/14.11/botnet.html
http://news.netcraft.com/archives/2006/05/17/blue_security_shuts_down_citing_ddos_attacks.html
http://www.securityfocus.com/news/11392

[2] During the course of researching this paper it became clear that there was evidence of cases of DoS which were the result of misconfiguration

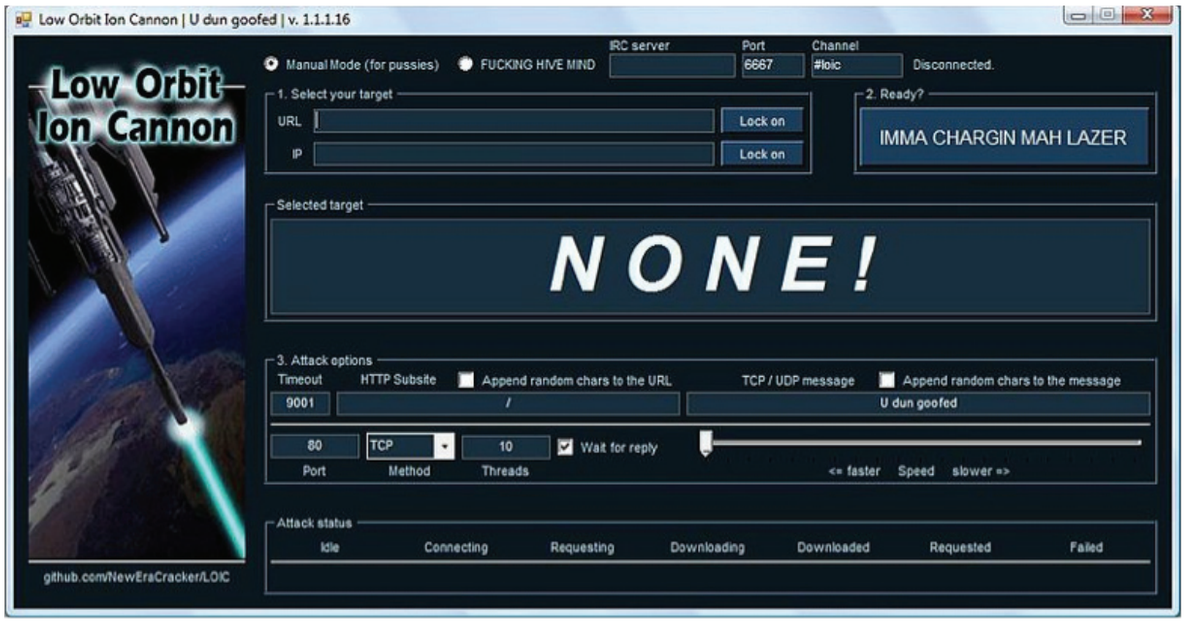
Figure 18:

Operation Payback Notice from Anonymous

Methods of Attack Associated with Denial of Service (DoS) and Distributed Denial of Service (DDoS)

Attackers employ many types of Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. Some are considered relics and infrequently used, while others are experiencing resurgence in popularity. The Low Orbit Ion Cannon (LOIC) attack performs both DoS and DDoS. This LOIC tool allows the user to launch flood-based attacks against Internet-facing hosts using TCP and UDP packets or through voluminous numbers of HTTP requests. The net effect is complete disruption and denial of services. The LOIC was used by the hacktivist group Project Chanology to attack websites belonging to the Church of Scientology and in attacks launched by 'Anonymous' against the Recording Industry Association of America in October of 2010.

Figure 19:
Low Orbit Ion Cannon



There were more than 30,000 reported downloads of the LOIC tool downloaded between December 8 and 10, 2010. Were they not routed through an anonymization network such as ToR, the source IP addresses associated with the tools would be logged and traceable by the recipient. Adding to the proliferation of this tool, a JavaScript version of the LOIC now exists and is released into the wild, enabling attackers to launch DoS of this variety from a Web browser interface.

The following table contains a representative subset of the types of DoS and DDoS attacks:

Attack Type	Attack Description
ICMP flood	ICMP floods, also referred to as Ping Floods, use the 'ping' command to send an overwhelming number of ICMP packets to targeted hosts. Variants of this type of attack include: Smurf Attacks, Ping of Death, and Nuke.
Teardrop attacks	Teardrop attacks send mangled IP-packet fragments with overlapping, over-sized payloads to targeted hosts with the intent to crash the victim's system.
Peer-to-peer attacks	Attackers leverage peer-to-peer attacks to create DDoS conditions. Attackers identify and exploit weaknesses and vulnerabilities in numerous peer-to-peer servers and clients. Several types of peer-to-peer attacks exist. In a peer-to-peer attack, there is no need for an attacker to communicate with clients. The attacker tends to act as a master manipulator, directing compromised clients to disconnect from the peer-to peer network and to connect to the victim's website. The desired result is that several thousand compromised computers aggressively begin trying to establish connections to the target site, rendering it either unavailable or inoperable. It should be noted that peer-to-peer DDoS attacks differ from botnet-driven DDoS attacks.
Asymmetry of resource utilization in starvation attacks	These attacks compromise a targeted host that has great computational power or significant network bandwidth or compromise a large number of hosts, and then direct the victim host(s) to attack as a group, creating a DDoS attack scenario. Smurf and SYN FLOOD attacks are examples of these types of attacks.

Attack Type	Attack Description
Permanent denial-of-service attacks	The concept of permanent denial-of-service (PDoS) attacks also known as phlashing involves launching an attack that damages a system so severely that it requires replacement and / or reinstallation of hardware. Unlike traditional DDoS attacks, PDoS attacks exploit security flaws that allow remote administration on the management interfaces of the target hardware. Examples of hardware which fall victim to this sort of attack include: • Routers • Switches • Printers Attackers exploit these vulnerabilities to remove and replace firmware with modified images. This often results in a condition known as "bricking" of the target device, which, renders it useless.
Application-level floods	These types of attacks manifest at the application layer (Layer 7) of the OSI model. The conditions that influence these attacks vary, as do the way in which they occur. For example: • IRC floods • Exploitation of common vulnerabilities (buffer overflows for example) • Over saturation of links • Banana or boomerang attacks
Reflected attack	A type of DDoS attack that sends forged requests to a very large number of target hosts, who are known in advance to reply to such requests.
Degradation-of-service attacks	These attacks are typically seen launched by compromised hosts on an intermittent basis. These intermittent floods create a condition of degradation, which sees performance slowed or crippled but not brought to a standstill.
Blind denial of service attacks	In a blind denial of service attack, the attacker uses one or more forged IP addresses to avoid detection by the victimized host. The victim host, because it is unable to detect the origin of the attacker, is unable to filter out the attacker's packets. This gives the attacker a significant advantage. The attacker must be able to receive traffic from the victim, and must additionally be able to subvert the routing fabric or even use the victim's own address. The TCP SYN flood attack is an example of a blind attack.
DoSNet	DoSNets are typically seen as part of greater botnet offerings. They represent the realization of the harnessed computational power of thousands upon thousands of comprised hosts.
SYN Flood	A SYN Flood attack sends a flood of TCP/SYN packets, often with a forged sender address. Each packet is handled like a connection request, causing the server to spawn a half-open connection by sending back a TCP/SYN-ACK packet, and waiting for a packet in response from the sender address. However, because the sender address is forged, the response never comes. These half-open connections saturate the number of available connections the server is able to make, preventing it from responding to legitimate requests until after the attack ends.

Deep Dive - An Analysis of Web Browser Attack Toolkits

The past several years have been witness to an unparalleled and astonishingly rapid development in the world of cyber crime – the emergence of a brand new underground ecosystem brought on by vast improvements in malicious software. Gone are the days when criminals created malware and infected millions of systems with the sole intention of making a name for themselves. Today's cyber crime is perhaps better organized than ever with emphasis on, above all, automated monetization of software exploits.

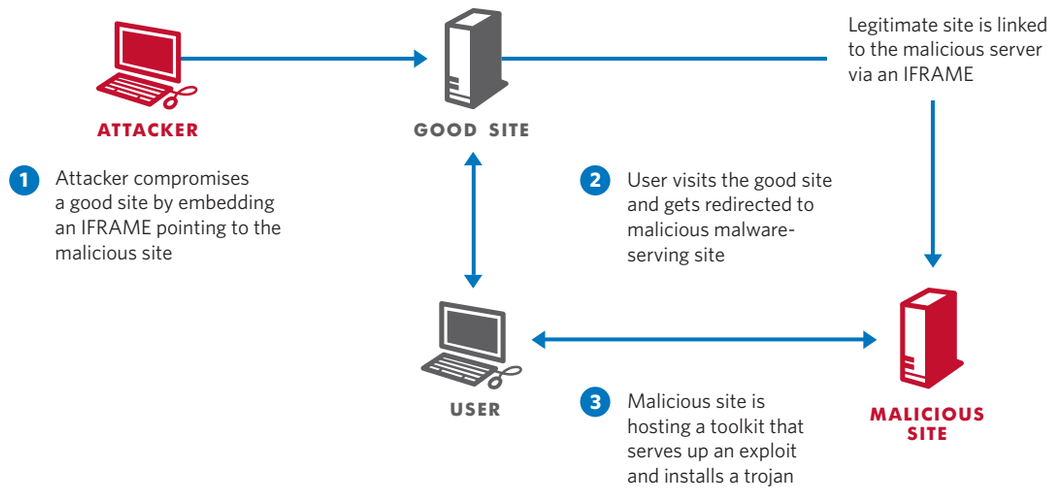
Nowhere is this more evident than with Web exploit toolkits, the invisible hand of crime that is present all around us. When it comes to conducting online crime, exploit toolkits are the weapon of choice for many cyber criminals. The trend started in 2006 with the release of WebAttacker, which is considered by many to be the first modern day Web exploit toolkit. An emerging trend ensued and soon took off, and today the Internet is subjected to hundreds of exploits originating from these toolkits.

As with any type of economic system, criminal or otherwise, the supply is heavily fueled by demand. Hence, with the ever-increasing, Web-based criminal activity, the number of exploit toolkits has skyrocketed and shows no signs of slowing down.

Web exploit toolkits attract cyber criminals with three primary characteristics: ease of use, high success rates, and large profit margins. Each of these will be discussed at length. Before delving into each characteristic, it is beneficial to first describe a typical attack scenario, as depicted in the following diagram (Figure 20).

Figure 20:

Typical Attack Scenario Utilizing a Compromised Website



In order to generate good return on investment, an attacker needs to locate and exploit a great many victims. The initial objective is to infect as many target hosts as possible in as short a time as possible. The attacker can accomplish this in several ways. The attacker can set up a fake website and hope that enough unsuspecting visitors stumble upon the trap. The attacker can also create a malicious advertisement that refers to his own exploit toolkit server, and then let the advertisement propagate through the Internet. However, the most common and most effective approach is to compromise a legitimate high-traffic website, which gives the attacker instantaneous access to a great number of website visitors. The attacker can then expose large numbers of visitors to the exploit kit and guarantee a high yield of infection.

The attacker has several options when it comes to compromising legitimate websites:

- Purchase access credentials to the server hosting the website
- Infiltrate the site through malicious ads – through the ad network
- Manually exploit the server in order to insert code that serves up malicious content

Once a legitimate server has been successfully compromised, the rest of the attack is relatively simple. The attacker places an IFRAME onto legitimate Web pages. The IFRAME exposes website visitors to the malicious code hosted on the attacker's exploit toolkit server. When an unsuspecting user visits the Web page, the page will load as usual and the user is unlikely to notice anything out of the ordinary. But in the background, the user's Web browser will also make a hidden connection to the malicious server, whereupon the exploitation can begin. If the user's machine is successfully exploited, a Trojan payload is installed on their machine, giving the toolkit administrator total control. At this point, the attack is successful and the attacker begins to monetize his exploitations.

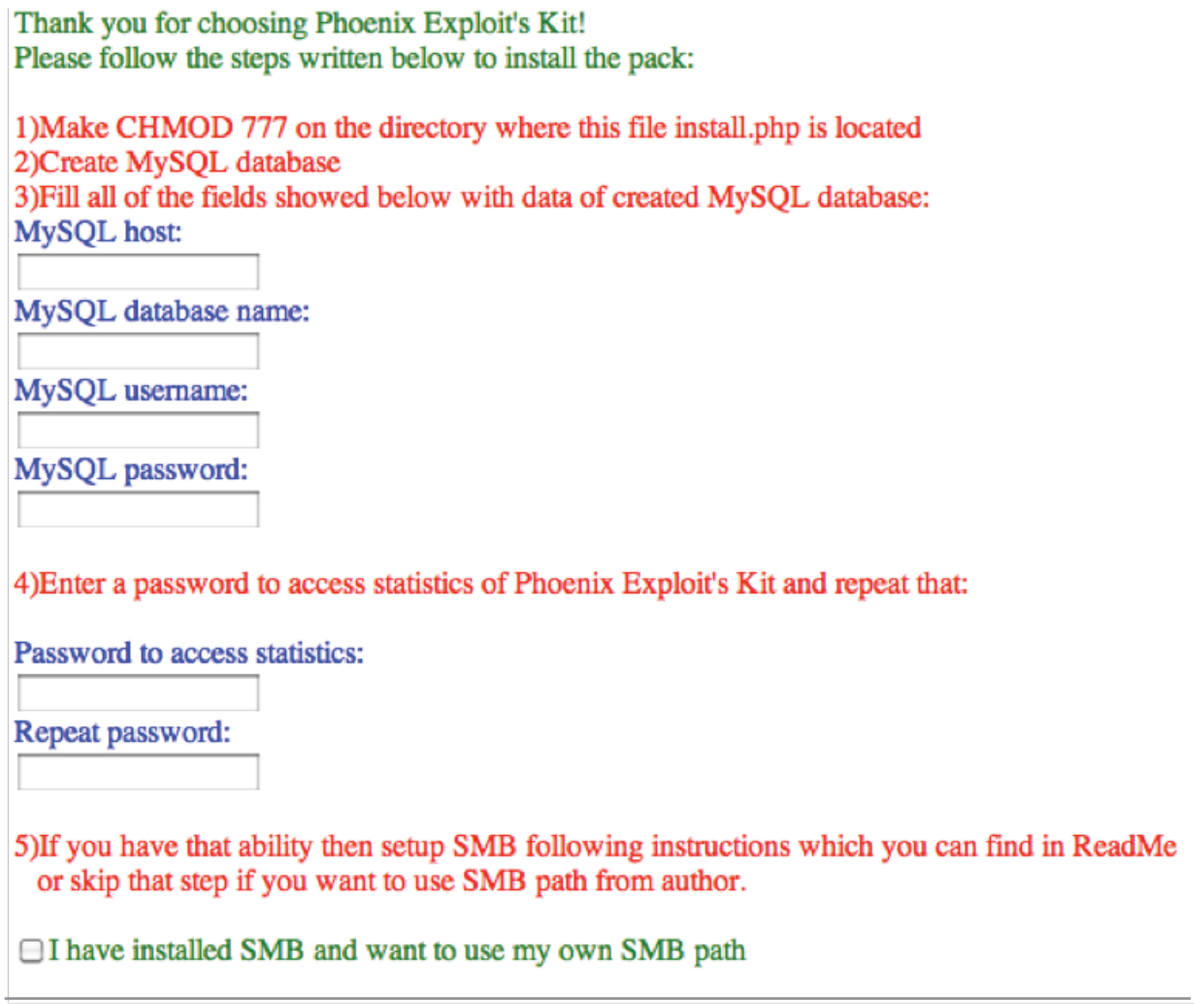
Exploit Kit Attraction #1: Ease of Use

Using a modern-day Web exploit toolkit is simpler and more productive than the manual process of running a script to exploit a machine and deliver a payload. Today's toolkits strike an uncanny resemblance to some of the popular Web content management systems. After all, a Web exploit toolkit is nothing but a content management system in which the content (most often a Trojan) is delivered through one of many different attack vectors (exploits).

Figure 21:

Login Screen for Phoenix Exploits Kit v2.4

Figure 22:
Screenshot of Install Script from Phoenix Exploits Toolkit v2.4



From a usability perspective, Web exploit toolkits are remarkably easy to use. A typical toolkit consists of an installation script, a login page, and an admin module. As shown in the above screenshot (Figure 22), the install script greatly simplifies the process of installing the toolkit onto a server. System requirements are minimal for most toolkits. Typically the only thing required is a recent version of Apache, PHP, and MySQL server. The toolkits leverage the typical LAMP server setup found in most of the Web. The installation script sets up a small database to keep track of statistics and, once complete, the attacker is ready to begin exploitation.

The admin panel is where most of the toolkit functionality lies. It allows users to keep track of visits and infection rates, which can be categorized by exploit, browser type and version, operating system, or even by a visitor's geographic data. Most toolkits also allow administrators to upload new payload binaries. Advanced exploit toolkits, which are discussed later, offer more integrated features that set them apart from basic competition.

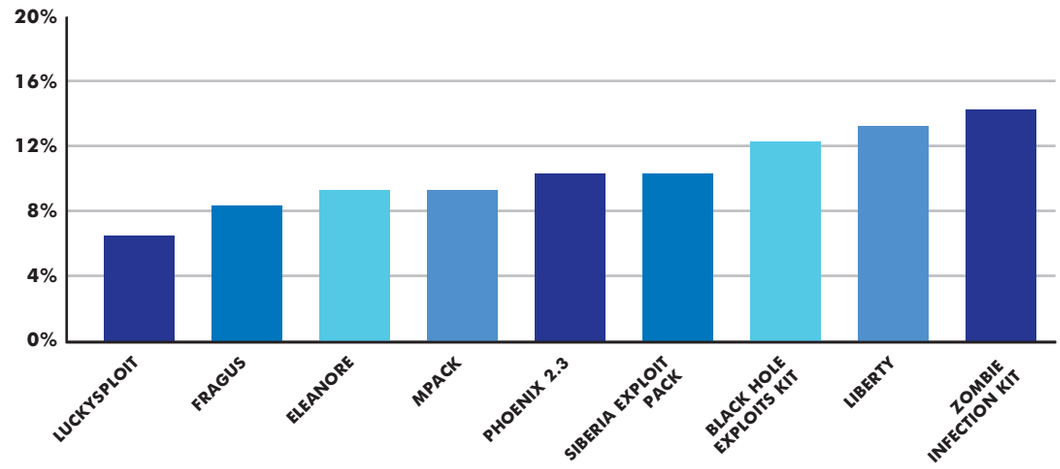
Figure 23:
Statistics Panel

The screenshot above (Figure 23) shows a typical statistics panel, which allows users to track statistics and the rate of infections, as well as details about each successful infection.

Exploit Kit Attraction #2: High Success Rates
The second contributing factor to the success of Web exploit toolkits is their incredibly high success rate. Each toolkit is pre-packaged with a set of exploits that are able to take advantage of vulnerabilities across a wide range of hosts, based on such characteristics as the host's operating system, Web browser, and collection of browser plug-ins, such as Adobe PDF and Flash. Each new release of a toolkit is likely to contain a new zero-day exploit that gives the attacker higher chances of successfully infecting targeted hosts. Some toolkits keep very old exploits (4+ years) to cover a corner case in which targeted hosts are running older, unpatched versions of vulnerable software. All of these toolkit features assist the attacker with infecting as many hosts as possible to increase profitability by monetizing the exploited systems.

Figure 24:

Infection Rates For Popular Exploit Toolkits



The graph above (Figure 24) shows the incredible infection success rates for some of the more popular exploit toolkits:

All toolkits achieve an amazingly high infection rate, with the highest sometimes over 15%. Even the lowest ranked, LuckySploit, achieves a 7.5% infection rate, itself an astoundingly high rate. To draw out a calculation, if the attacker is able to compromise a website that attracts 100,000 visitors a month, then the attacker is able to exploit 7,500 hosts each month. Each exploited host is itself compromised and is then at the mercy of the attacker.

The high infection rates can undoubtedly be attributed to zero-day exploits as well through the use of hard-to-detect Trojans like SpyEye or ZeuS. The combination of the two provides for a truly effective exploitation tool.

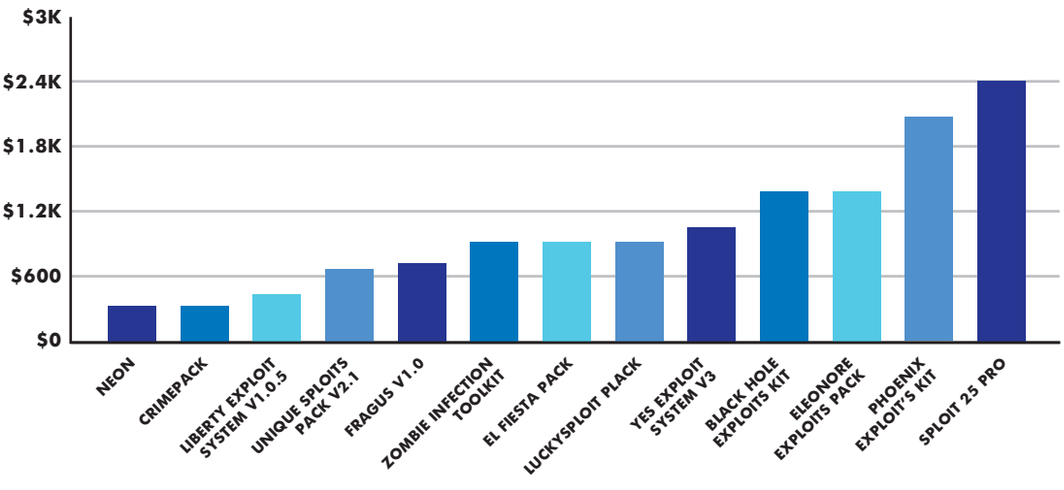
AVG's Web security research team discovered that a network of 1.2 million malware-infected hosts were compromised and administered by a Web exploit toolkit called Eleonore. As part of the two-month long study, AVG researched 165 Eleonore toolkits in use by the attackers. AVG concluded that those using the Eleonore toolkit achieved a 10% infection rate, compromising one in ten of the 12 million hosts who visited the compromised Web pages.

Exploit Kit Attraction #3: Monetization of Toolkits

Monetization of Web exploit toolkits can be split into two camps. Profit generation by toolkit creators/maintainers in one camp and monetization of exploited systems by toolkit users in the other. Beginning with the first camp, the business of creating toolkits, the following graph (Figure 25) shows estimated toolkit prices.

Figure 25:

Estimated Toolkit Prices



Similar to legitimate, non-criminal software packages, the underground CrimeWare ecosystem has moved towards a service-on-demand software sales approach. The latest toolkits come with more than just the cost of the software package. Most toolkits require additional small fees to obtain version updates. The process of deploying zero-day exploits onto toolkits has been modularized so that third party sellers can easily create and sell their own exploits, assuring that their offerings are compatible with the toolkits. This type of collaboration expands the features and functionalities available to the toolkit by enabling third parties to supplement the core capabilities, thereby increasing the number of exploits the toolkit can execute.

The real money for the CrimeWare engineers, however, is in the services they sell. These services include:

A A virus-scanning service. Some toolkits offer a service much like VirusTotal. Malicious binaries can be uploaded to the (Scan4You) service and the toolkit then returns a list of anti-virus engines that detect the given malware package. The price of this service is $0.15 per scan or $25 per month for unlimited scanning.

B Domain blacklist verification. Attackers use this service to determine if any of the domains in which they host malicious files have been blacklisted on one of many DNS/IP reputation services.

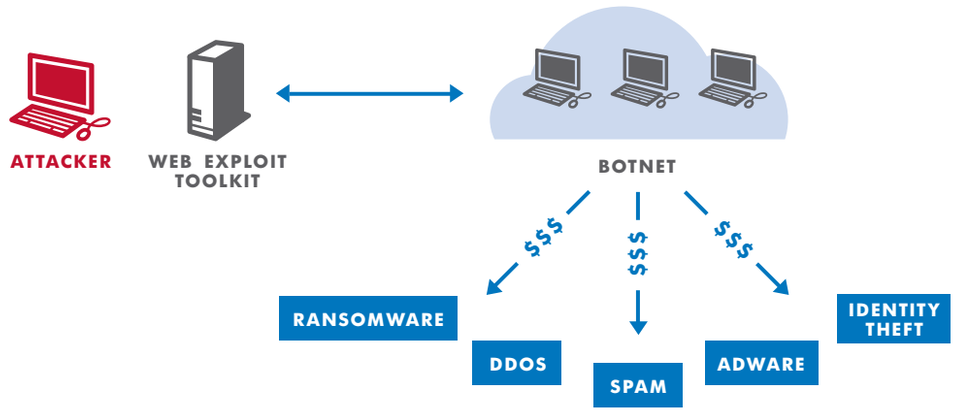
C Multiple firewall bypass verification service. For $3 per use, attackers can verify whether or not its malware bypasses certain firewalls.

All these services provide a perpetual stream of revenue to CrimeWare authors, allowing them to make more money from existing customers, which in turn funds a faster and more feature-rich delivery of their next exploit kit. It is also worth noting that some exploit kits share the same code base. In fact, some underground groups re-release their kits under different names in hopes of gaining market share.

While one camp generates revenue through the development and ongoing service of an exploit toolkit, the other camp focuses on helping attackers monetize exploited systems.

This is the real differentiator between exploit kits and other forms of malware, and is the primary reason the number of released toolkits has exploded in the past several years. The toolkits enable attackers to accomplish their most desired goal, easily and efficiently infecting as many hosts as possible. Once infection is achieved, the ability to monetize the compromised host is relatively easy. The following illustration (Figure 26) describes how a Web exploit toolkit administrator cashes in with a set of compromised hosts, collectively referred to as a botnet:

Figure 26:
Illustration of Toolkit Cashing with Botnet



Attackers can use a variety of methods to generate revenue with the infected network of hosts. They can install RansomWare on each host, which encrypts the user's data and then extorts the user to pay a ransom fee to obtain the decryption key required to unlock the user's data. Many users pay the extortion fee to restore access to their data. Other potential uses include sending out spam from the infected machines, installing AdWare to generate revenue through Ad views, and stealing private data from the compromised host through the use of key-logging software. The most common botnet use is to launch a DDoS attack. Often organizations use DDoS attacks to cripple—or render inaccessible—popular websites, commonly to send a political message or to hinder a business opponent. DDoS attacks may even be launched to extract ransom money from legitimate businesses who would rather pay the ransom than experience the prolonged downtime brought on by the DDoS attack.

Certain toolkits allow the attacker to sell botnet "services" using third party sellers, in which a toolkit administrator creates seller accounts and grants the sellers access to portions of the botnet. The profit is then split between the toolkit administrator and the seller, allowing the administrator to focus on generating more traffic for the compromised hosts, while utilizing sellers to offload the work of monetizing the infected hosts.

In summary, once a set of hosts has been compromised and the botnet constructed, the types of revenue-generating activities is seemingly limitless. The more hosts in the attacker's botnet, the more money the botnet generates. Today's Web exploit toolkits facilitate the process of easily and automatically building a botnet, and then using the botnet to monetize the exploited assets.

Impact

In April 2009, attackers employed the LuckySploit exploit toolkit to compromise the website of Paul McCartney before a big publicity event. The timing of the infection, immediately preceding an upcoming New York reunion concert, is believed to be intentional, with the attackers aiming to make the most of exploiting the tens of thousands of website visitors.

The attackers compromised the website to host an embedded IFRAME, which linked back to the malicious server hosted in Amsterdam. The attack was unique in that it not only used a different character encoding to cloak the malicious JavaScript, but it also made detection of its payload difficult to detect by using SSL to encrypt it. The payload, ZeuS malware, was deployed to harvest a large amount of Web traffic in order to infect as many machines as possible.

Attacks such as the one on Paul McCartney's website are hardly rare these days. Legitimate websites are increasingly compromised with IFRAMES linking back to exploit toolkits, and detection of these attacks is becoming more difficult.

In another example, attackers compromised the customer base of a large United Kingdom financial institution in July 2010. The customers, via their use of the financial institution's website, were infected through the use of the Eleonore and Phoenix toolkits. The infection payload installed a banking Trojan, ZeuS V3, onto the computer of each exploited customer. The attackers executed their activities through one of three attack vectors:

- Compromised legitimate websites
- Fraudulent websites using fake ads to spread the infection
- Malicious advertisements published on legitimate websites

With the Trojan successfully loaded onto a victim's computer, the Trojan patiently waited for the user to visit the banking website, at which point it would steal the customer login data and forward it to the Command & Control server. The C&C server initiated a money transfer from the victim's account, through money mules, and into an attacker-owned account. The Trojan created a ruse to decrease the probability of detection. Each time the compromised customer logged into the banking website, the Trojan displayed a fake balance statement to the customer. The exploitation of the banks customers went on for a questionable amount of time, affecting upwards of 3,000 customers. In the end, a total of £675,000 was smuggled out of victims' bank accounts.

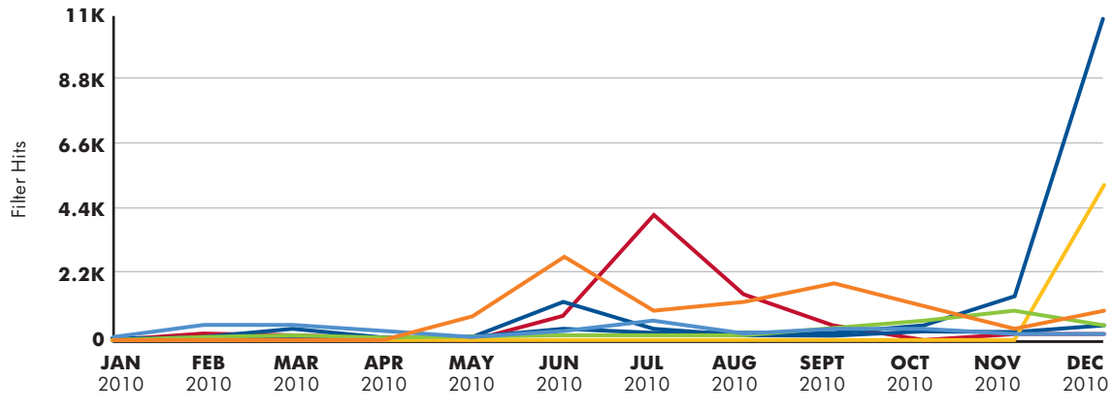
Attacks like the ones against Paul McCartney's website and the U.K. financial institution may still be carried out today, and due to the level of sophistication in today's Trojans, they can be very difficult to detect.

The impact that Web exploit toolkits have on the security of the Web is more immense than commonly imagined. New websites built to host various toolkits pop up every day, waiting for unsuspecting users to visit and be compromised. In the underground market, stolen access rights to legitimate websites are constantly sold to attackers who use the information to tunnel traffic to their own malware-hosting servers. Users even receive spam emails, sometimes from their infected acquaintances, that refer them to malicious domains intent on compromising their computers.

Attackers employ numerous methods for infecting hosts via an exploit toolkit, and the data indicates they

Figure 27:

Filter Hits on Exploits from Popular Toolkits



are successful. The above graph (Figure 27) shows data that HP DVLabs gathered in the previous 12 months. The data depicts filter hits corresponding to the CVEs of exploits used in toolkits recently analyzed by HP DVLabs.

A notable portion of the graph is the spike in malicious traffic that occurred in the middle of 2010, centered around June and July. HP DVLabs' research indicates that the spike is likely attributed to a large attack campaign using the then newly released version of the Phoenix Exploit toolkit. The spike receded quickly, probably due to the takedown of malicious domains involved in the attack.

An even more alarming spike started in November 2010 and continued into 2011. A likely cause of the spike is an increased use of exploit toolkits, again probably due to new releases of toolkits, along with attackers' desire to exploit an increase in financially-related Internet traffic brought on by the holiday shopping season.

Looking forward, HP DVLabs believes the Web exploit toolkit attacks will have a larger impact on business in 2011 than in 2010. Based on the data presented here, it is not hard to imagine large enterprise networks having their network perimeters breached and their machines compromised through the use of these toolkits. Trojans are expected to become increasingly more sophisticated, a trend already evident by the merger of SpyEye and ZeuS. Attackers will evolve social engineering techniques to attract a maximum amount of Web traffic to malicious servers hosting exploit toolkits.

Mitigation

Protecting against attacks originated with Web exploit toolkits is becoming increasingly difficult. However, there are ways to minimize the risk of infection. One of the most effective defenses is to either install patches onto host systems, although these aren't always available in the case of zero day vulnerabilities, or leverage a "Virtual Patch" using technology like HP TippingPoint's IPS (Intrusion Prevention System). It's important to think about prevention from both a vulnerability (CVE) standpoint as well as obfuscation (JavaScript) angle as well. Doing so reduces the number of attack vectors that are available to attackers and reduces the risk that the host's users will be exploited when traversing the Web. This advice is extremely important for Web browsers because they are the gateway through which Web exploit toolkits compromise host systems. Users should always be using the latest version of their Web browser, which is made easier these days by automatic updates, but still leaves browser plug-ins to think about as well.

Another defense and way to reduce risk against these toolkits is the use of URL reputation services, such as HP TippingPoint RepDV (Reputation Digital Vaccine). Deploying reputation services at user endpoints or embedded in the network through network security devices such as an IPS is aimed at preventing access to domains known to host malicious websites. It reduces the risk of infections by reducing the number of malicious websites that users are allowed to visit.

References

1. MalwareInt – Russian Crimeware Prices (http://mipistus.blogspot.com/2009/03/los-precios-del-crimware-ruso.html)
2. MalwareInt – Russian Crimeware Prices Part 2 (http://malwareint.blogspot.com/2009/08/prices-of-russian-crimeware-part-2.html)
3. MalwareInt – Campaign Infection Through Phoenix Exploit's Pack (http://malwareint.blogspot.com/2010/08/campaign-infection-through-phoenix.html)
4. Scan4You (http://scan4you.net/)
5. eWeek – Exploit Toolkits: Software That Makes Cyber-Crime Easier (http://www.eweek.com/c/a/Security/Exploit-Toolkits-Software-That-Makes-CyberCrime-Easier-411813/)
6. CyberInsecure.Com - PaulMcCartney.Com Compromised Through Exploit Toolkit, Visitors Might Get Private Data Stolen (http://cyberinsecure.com/paulmccartneycom-compromised-through-exploit-toolkit-visitors-might-get-private-data-stolen/)
7. InfoSecurity - McCartney Site Serves up Zeus Malware (http://www.infosecurity-us.com/view/1178/mccartney-site-serves-up-zeus-malware/)
8. Net Security – ZeuS-SpyEye Merger (http://www.net-security.org/malware_news.php?id=1512)
9. MalwareInt - Phoenix Exploit's Kit: From the Mythology to a Criminal Business (http://www.malwareint.com/docs/pek-analysis-en.pdf)
10. MalwareInt - State of the Art in CRiMEPACK Exploit Pack (http://malwareint.blogspot.com/2010/05/state-of-art-in-crimepack-exploit-pack.html)
11. ExploitKit - CVE Exploit Kit List (http://exploitkit.ex.ohost.de/CVE%20Exploit%20Kit%20List.htm)
12. MalwareInt – Phoenix Exploit Kit 2.3 Inside (http://mipistus.blogspot.com/2010/10/phoenix-exploits-kit-v23-inside.html)
13. MalwareInt – SpyEye Analysis (http://www.malwareint.com/docs/spyeye-analysis-en.pdf)
14. TechTarget – Siberia Exploit Toolkit Gets Update to Evade Antivirus (http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1524521,00.html)
15. M86 – Cybercriminals Target Online Banking Customers (http://www.m86security.com/documents/pdfs/security_labs/cybercriminals_target_online_banking.pdf)
16. Malwareview – Forums (http://www.malwareview.com)

Definitions

Command Execution:
A type of vulnerability which takes advantage of a lack of input validation on a website in order to run operating system commands on the vulnerable application server. Typically, this vulnerability category allows attackers to exploit Web applications that pass user data as parameters to I/O operations by appending OS commands to user supplied input using special characters such as a pipe (|).

SQL Injection:
A type of Web application vulnerability which takes advantage of a lack of input validation on a website in order to execute unauthorized database commands on a Web applications database server. When successfully exploited, data can be extracted, modified, inserted or deleted from database servers that are used by the vulnerable Web application. In certain circumstances, SQL Injection can be utilized to take complete control of a system.

Cross-Site Scripting (XSS):
A type of Web application vulnerability which takes advantage of a lack of input validation to enable an attacker to inject malicious client-side code into a Web page which is viewed by a victim's Web browser. Various forms of XSS are currently being used to phish Website users into revealing sensitive information such as usernames, passwords, and credit card details. XSS can generally be divided into stored, reflected, and DOM-based attacks. Stored XSS results in the payload being persisted on the target system either in the database or the file system. The victim will retrieve and execute the attack code in his browser when a request is made for the stored information. Execution of the reflected XSS attacks, on the other hand, occurs when user input from a Web client is immediately included via server-side scripts in a dynamically generated Web page. DOM-based XSS attacks rely on malicious modification of the DOM environment in a victim's browser. It differs from the stored and reflected XSS in the fact that the malicious data is never sent to the server. Via some social engineering, an attacker can trick a victim, such as through a malicious link or "rigged" form, to submit information which will be altered to include attack code and then sent to the legitimate server.

Denial of Service (DoS):
A type of vulnerability which allows an attacker to exhaust computer resources on a vulnerable system to a point where legitimate usage of the system in question is impossible.

Distributed Denial of Service (DDos):
A type of DoS attack which employs a number of separate computers which simultaneously launch a Denial of Service attack against a single application or system.

Remote File Include:
A type of Web application vulnerability which takes advantage of a lack of input validation on a website in order to execute unauthorized code (typically PHP or ASP) on a vulnerable server. Remote file-include attacks typically arise from a scripting language's inherent ability to include code from external URLs, or arbitrary local files. It is this ability which allows the attacker to include un authorized code from an external source.

Cross Site Request Forgery (CSRF):
A type of Web application vulnerability which takes advantage of a lack of authorization on a vulnerable Web application to allow an attacker to execute application commands on behalf of another user of the application. The typical scenario of a Cross Site Request Forgery attack involves an attacker tricking a victim into clicking on a specially crafted link which is designed to perform a malicious operation on behalf of the victim. For example, a victim may click on a malicious link which forces the victim to transfer money from the victim's bank account to an attacker's bank account.

Server Side Attack:
Any attack targeting a server or a Web application (such as a website or a file server.)

Client Side Attack:
Any attack targeting a client application (such as a Web browser or Spreadsheet Application.)

Share with colleagues

Get connected
www.hp.com/go/getconnected
Get the insider view on tech trends, alerts, and HP solutions for better business outcomes

© Copyright 2011 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Trademark acknowledgments, if needed.

4AA0-xxxxENW, March 2011

This is an HP Indigo digital print.