The Web2.0 Attack Vector

Adaptive multi-layered protection against modern dynamic web2.0 threats and a proposed SWG (Secure Web Gateway) implementation guidelines

Iftach Ian Amit ▸ 7/1/2009

The Web2.0 Attack Vector

Adaptive multi-layered protection against modern dynamic web2.0 threats and a proposed SWG (Secure Web Gateway) implementation guidelines

The Attack Vector

Web attacks are not composed of a single event such as access to a malicious website, or the downloading of an infected file. Attacks move along an attack vector which is the most effective way to infect a victim. Along the vector there are four main stages – accessing a resource on the Web (website), running active content within the website (scripts, "Web 2.0" dynamic content), downloading content (executables – usually a Trojan or a Trojan downloader), and delivering an additional payload as well as reporting back to the attacker (other Trojan, stolen information sent to perpetrator).

The attack vector represents the different techniques and access layers which are applied in most of the standard web attacks we face today:

- *Initiation* – happens when a resource is accessed on the Web (a website – usually a legitimate one)
- *Activation* – happens when active content (scripts, Web 2.0 elements) hidden in the HTML code "run" on the victim's machine. This content is run just like other legitimate active content which the website is composed of.
- *Infection* – occurs when executables (Trojan, Rootkit, Key-logger) are downloaded to the victim's machine after successful exploitation in the activation stage, and are installed on the system.
- *Attack* – the payload (Trojan) that runs on the system is operational, and downloads additional malicious components, as well as sends out data that will be used to generate revenues for the criminals (financial information, classified documents, personal information, etc.).

The graph titled Figure 1 shows the attack stages and the multiple inspection layers required to provide security at each stage of the attack

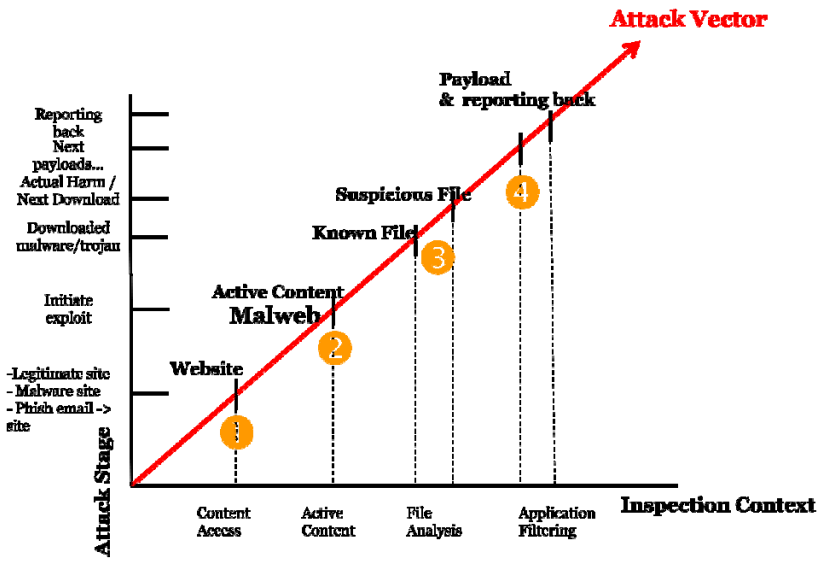

Figure 1

A Secure Web Gateway (SWG) is designed to handle the web threat, and as such should employ a multi-layered approach that corresponds with the attack stages:

Layer 1: Content Access

The first layer of a SWG solution is URL filtering for malicious websites, as well as for potentially offensive and non-work related material such as pornography, hacking, and gaming sites. Additionally, the layer filters unauthorized and invalid web certificates and conducts ActiveX code blocking by using white and black lists that contain a directory of downloadable and non-downloadable code. In this layer, most of the functionality provided should be aimed towards productivity and acceptable use rather than security (although trivial security issues such as known malicious sites are blocked at this stage), as most of the attacks today are conducted through legitimate sites that would usually not be blocked by URL categorization solutions.

Layer 2: Dynamic Web Threats

The second layer of protection is focused at filtering malicious content embedded in web pages. This layer is rather difficult to achieve as it should employ content analysis engines (rather than signature or heuristic engines) that detect zero-hour exploits, and malicious content that is part of the legitimate content delivered by the website. Such malicious content (MalWeb) is composed of the same building blocks which are used by modern sites – specifically JavaScript, VBScript, ActiveX and Flash. Additionally, Layer 2 inspections should check for HTTP headers anomalies to further protect the organization from attacks. Most of the MalWeb detection is done in this layer, and the malicious parts of the web content should be stripped out before being delivered to the requesting user. This enables the protected networks behind the SWG to still take advantage of the functionality offered by the legitimate sites that were compromised to deliver malicious code in addition to the benign content.

Layer 3: File Analysis

When performing file analysis (the more traditional security offered by anti-virus products), the SWG should inspect files against known and unknown threats that can be transported in executable and binary files. Inspection should be performed by using simple signatures to detect known malware and smart signatures that use intelligent problem solving methodologies or heuristics to find patterns that signify the presence of a zero-hour exploit. In addition, modern SWG solutions should add a secondary scanning engine by utilizing a 3rd party AV provider (OEM) to scan for known and traditional threats more efficiently.

Layer 4: Application Control

The final layer in an effective implementation of a multi-layer security model needs to filter and block outbound and inbound communications. Application layer filters should detect malicious outbound and inbound communication protocols used to transport malware such as Trojans, spyware, key-loggers and viruses. This layer should also provide detection and control over communications such as peer-to-peer file sharing, instant messaging (IM) and other productivity related applications. SWG providers should also offer a basic DLP (Data Leakage Protection) solution for gaining additional control over the content of the data that goes out of the organization.

Web attack use-case

Following is a standard use-case of how web attacks are conducted these days, and how an SWG should behave at different layers of the attack:

A user browses to a (shopping/news/sports/etc...) website. The site is a legitimate site and is accessed by the user on a daily basis. However, the site in question was recently compromised as attackers have injected a malicious script into it, and now, in addition to

the content it usually delivers to its users, it also contains MalWeb. At this point, the SWG should not block or alter the browsing behavior (since the site does not violate the acceptable use policy of the organization, and is not categorized as malicious).

As the website content is sent to the user, the SWG should examine the content carefully – the legitimate parts should be identified as such, and the MalWeb sections should trigger the SWG content inspection mechanisms. As MalWeb is interpreted code (i.e. – non compiled code), a "profiling" of the code's behavior is needed in order to determine – based on what the code intends to do – if it is benign or malicious. When the SWG detects the MalWeb part of the page, it should strip it, and leave the rest of the content intact so that the user can still have access to it. Even zero-hour vulnerabilities should be detected in this way – since the code behavior is inspected rather than a signature!

Bearing in mind that there is no such thing as 100% secure, the next layer in the attack (if the MalWeb somehow passed the detection), actually introduces the persistent component – the Trojan. This is an executable file (and is analyzed even when represented as text in an HTML page), which should be scanned by the SWG using both signature and heuristic engines, as well as by behavioral engines to determine whether it is a known threat; even if a signature is not triggered, the file should be checked for infringing behavior that matches the "profile" of what a threat would behave like (i.e. Trojans, key-loggers, rootkits, spyware - all have their behavioral profiles which differ from legitimate software). At this point the SWG should block the malicious files, and keep the network clean of threats.

With all the network layers in place, it's important to remember that content does not only get onto systems via the network. If a user has brought in an infected USB-flash drive, a desktop AV solution is an important last-mile layer to have in place. Nevertheless, if a system does get infected (abusing the latency in delivering signatures for emerging threats), the SWG component on the network should be able to detect the communications to/from the infected system carrying either commands to the Trojan, or stolen data being sent to the attackers for revenue generation. Such scanning should not only over traditional HTTP/FTP protocols, but inspect all ports and protocols since Trojans will usually try to disguise themselves from routine communications that are inspected by other security solutions.