McAfee® Anti-Malware Engines: Values and Technologies

By Christoph Alme and Declan Eardly, McAfee Labs™
Table of Contents

Cooperative Anti-Malware on Endpoint and Gateway ............................................................... 3
  The Value of Malware Protection at the Endpoint ................................................................. 5
  The Value of Malware Protection at the Network Gateway .................................................. 6

A Primer on Malware Detection .................................................................................................. 7
  Exact Detection and Identification .......................................................................................... 7
  Generic Family Detection ....................................................................................................... 7
  Heuristic Detection .................................................................................................................. 8

Today’s Advanced Malware Detection Technologies ............................................................... 9
  Automated Intelligence ............................................................................................................ 9
  Risk Mitigation ....................................................................................................................... 10
    False-Positive Prevention ..................................................................................................... 10
    Code Security ..................................................................................................................... 10
  Cloud-Based Detection .......................................................................................................... 10
  Emulation ............................................................................................................................... 11
  Unpacking .............................................................................................................................. 12
  Statistical Classification ........................................................................................................ 13
  Fuzzy Fingerprinting ............................................................................................................ 13
  Generic Exploit Detection ..................................................................................................... 14
  Behavioral Analysis .............................................................................................................. 15

ProActive: Behavioral Anti-Malware Add-On for Gateways ....................................................... 15
  ProActive Technology in Action ............................................................................................ 17

About The Authors .................................................................................................................... 21

References ..................................................................................................................................... 21
Cooperative Anti-Malware on Endpoint and Gateway

In this report, we describe the McAfee anti-malware engine core technologies, as well as its values, ProActive behavioral detection technology, and how the engine flavors—anti-malware engine for endpoint and anti-malware engine for gateway—combine to form the optimal anti-malware architecture for your enterprise.

A rudimentary, off-the-shelf gateway that simply runs another anti-virus engine, as it would run on a desktop, implements redundant anti-virus protection, but not with the aforementioned gateway-level anti-malware values. Anti-virus software vendors obviously have to achieve similar goals in their products. The same problems and goals are usually solved and achieved by similar, if not identical, technologies. Thus, a combination of two anti-virus engines rarely accomplishes anything more than running the same, or similar, technology twice. This may offer a little fail-over redundancy, but no substantial added value and especially, no defense in depth. As Gartner puts it, “There is a modest 2% to 10% variation in the comparative effectiveness of signature-based malware engines. ... Consequently, moving from one signature-based detection engine to another still leaves gaping holes in protection.”[1]

In practice, you would usually see the false-positive rate double in this legacy approach. When mixing two likewise anti-virus engines on one device, such as a gateway, you would also see the memory footprint double and performance diminish. So let us investigate the new ways of cooperative anti-malware that truly deliver added value, based on end-node-optimized technologies.

Two-vendor setup: This is the intelligence “black hole.” Suspicious findings from the desktop protection will probably not improve the detection of your gateway product, and vice versa.

Single-vendor end-node technologies: These are integrated best of breed. Suspicious findings at either end of the network will also improve the other side’s protection.

The McAfee anti-malware engine for gateway provides the required added value for gateways through truly unique technologies: ProActive behavioral anti-malware add-on, and Artemis in-the-cloud protection. These technologies keep the vast majority of malware out of your network. Most threats are stopped at the perimeter, before they reach your endpoints, and this approach provides...
the lowest performance and management costs. Now let's take a look into detection efficiency numbers.

As a founding member of the Anti-Malware Testing Standards Organization (AMTSO), McAfee is committed to fair and independent anti-malware testing. The McAfee anti-malware engine offering, both in its endpoint and its gateway flavors, is a regular participant in tests performed by the independent AV-Test GmbH, one of the most well-known and credited virus-testing organizations in the world, formed as a spin-off from Magdeburg University, in Germany. Furthermore, our anti-malware engine (again in both flavors) participates at the respected VirusTotal.com site—a free web service that has been awarded one of the best “Security Web Site” products by PC World magazine. Comparing detection rates of our Anti-Malware for Endpoint and Gateway against main competitors—measured in 2009 and 2010 by AV-Test—shows not only how McAfee leads in detection rate overall, but also the high-level of consistency at which these detection rates are achieved. A reliably high detection rate over months and years is of great value to any enterprise.

In combination, anti-malware technologies for endpoint and for gateway allow strict enforcement and the highest proactive catch rates at the network perimeter, keeping the majority of threats outside of your network, and effectively protecting your desktops. The following chapters will detail the engine technologies involved.
The Value of Malware Protection at the Endpoint

Running anti-malware on any endpoint node is a mandatory requirement in all enterprises today, and its foremost value of protection against malware is obvious. In this chapter, we’ll focus on some values that are specific to anti-malware running on the desktop environment.

Insight
Running on the endpoint, the anti-malware engine runs exactly where the bad things happen. It is “onsite,” seeing all potential attack vectors into the protected system—both network originating (mail, web) and purely local ones (CD drives, USB sticks). And with on-access and in-memory scanning it can gain insight into the real, at-runtime gestalt of threats. Some rootkits and backdoors, for example, live solely in memory and never touch down on the local disk.

Accuracy
All user installations of the McAfee anti-malware engine worldwide scan an estimated amount of more than

4.7 trillion files per day

underlining the need for accuracy and reliability of the engine. This is also constantly put to the test in public, transparent tests conducted by reputable organizations such as AV-Test, AV-Comparatives, Virus Bulletin magazine, and others.

Disinfection
The most important job of an anti-malware solution is to protect from malware. Sometimes, systems may have already been infected before the anti-malware solution comes into play. In these cases the disinfection capabilities of an anti-malware engine are needed, to try to clean up as much of a user’s data as possible (documents, media files, executables) after a malware attack has happened. Also,
any malware family that infects files needs cleanup that goes beyond just deleting the malware file itself. For example, W32/Conficker.worm drops an (obfuscated) autorun.inf—a file interpreted by Windows CD AutoPlay functionality—onto any connected removable storage drives (such as USB sticks) to propagate through that vector.

**Manageability**

Although the manageability value might be negligible for home users, it is essential for enterprise use. Remote configuration and control of thousands of endpoint installations, from one central management console such as McAfee ePolicy Orchestrator, significantly reduces the cost of ownership of endpoint anti-malware protection through ease of deployment, configuration, and monitoring.

**The Value of Malware Protection at the Network Gateway**

Gateway-level anti-malware protection implements unique, gateway-specific technologies that provide the following values.

**Always On**

Whether for mobile workers’ laptops, guest laptops, or less prevalent operating systems (Linux, Mac), ensuring that effective and up-to-date anti-malware protection is running on each client in the corporate network is a daunting task. This solution mandates a central point of anti-malware enforcement at the network perimeter.

**Enforcement**

An enterprise environment requires strict enforcement of policies against various potentially unwanted programs and suspicious behaviors that can also be found in software that is not necessarily malware. The false-positive rate of a gateway-level anti-malware solution needs to be low to keep it manageable and to keep the total cost of ownership low. However, in contrast to desktop security, a false positive at a gateway has no negative effects on an affected client (other than blocking access to a mail, website, or download). This unique sensitivity of the gateway layer allows strict and effective policy enforcement.
Performance  Anti-malware at the gateway-level—if based on gateway-specific anti-malware technologies—provides real defense in depth at the lowest cost: In combination with a web gateway’s caching abilities, for example, resources can be scanned just once for many clients in a large network, rather than having each client PC run the same scan on its own.

Resilience  Because no security provides 100 percent protection, malicious downloads or mails can still pass through the gateway and infect desktops. A roaming user may have caught an infection while outside of the corporate protection. Malware often tries to disable desktop protection and its update mechanisms. With the ProActive add-on in place, the web gateway, however, is not affected by malicious takeovers of client PCs and can still detect suspicious network traffic, and block and isolate the infected system. Protection on outbound traffic helps to diminish corporate liability, as attacks driven through an already infected guest laptop, for example, would otherwise appear to have their source on your corporate network.

A Primer on Malware Detection

Exact Detection and Identification

In contrast to the important generic and heuristic detection techniques that we shall explain later, techniques largely based on experience and assumptions rather than proof, exact detection has the seemingly trivial benefit of blocking malware because the detection knows what malware it is. Usually based on signature scanning, exact detection and naming of malware results in the best possible accuracy, performance, and user confidence. Repair facilities of an anti-malware engine also benefit from a sound identification of the given malware strain. False-positives are kept to an extreme minimum, sparing administrators from manual whitelist maintenance. Yet the efficient application of millions of detection rules in parallel, on a scanned file, is an art of its own. This requires highly optimized algorithms and optimal code implementations for storing parallel detection rules, and applying them for fast search and accurate classification.

The analysis required to create such sound detections, resulting in the accuracy described above, is also this method’s Achilles heel. With targeted and short-span malware more prevalent than ever, samples of particular malware may never appear on an anti-malware vendor’s radar, or the malware may have left the scene by the time analysis is done and signature updates are shipped. This can be partly countered with a high degree of automation of this process and fast response times. (See “Automated Intelligence.”

Generic Family Detection

Seen one, seen ‘em all: Generic detection extracts key characteristics of one or a few samples of a malware family or exploits, and creates a “one size fits all” detection rule that will catch as many
variants of the same family, or exploits for the same vulnerability, as possible. Generic detection rules use similar techniques as exact detections do, and with only a slight increase in the false-positive risk we can achieve a significant gain in proactive protection.

The first reaction to new malware can be a fingerprint- or signature-based detection, and more generic detection rules can follow quickly to combat any upcoming variants of a new malware family or exploits.

Sometimes, the quality of generic detection becomes visible no sooner than months or even years after its creation. This was the case with the prominent Animated Cursor vulnerability, probably the most exploited vulnerability in 2007. This “new” vulnerability turned out to be based on the same bug that had already led to the Animated Cursor and Icon Format Handling vulnerability (MS05-002) in 2005. Two years later, attackers used a slight adjustment to trigger the vulnerability that apparently was still present in Windows’ user32.dll. Thus a generic Gateway detection for the 2005 vulnerability could also protect customers proactively against the Windows Animated Cursor Remote Code Execution vulnerability (MS07-017) two years later.

**Heuristic Detection**

Where there’s smoke, there’s fire. Heuristics combine a few known facts with experience to recognize a general classification. This family of detection techniques generally provides outstanding proactive detection capabilities against new malware variants, new malware families, and even against unknown vulnerabilities and exploits.

Given such outstanding benefits, the slightly increased perceived false-positive rate associated with any heuristics does not spoil the attractive cost-performance ratio. With a moderate false-positive rate, ranging between a low 0.002 percent (normal websites) up to 0.04 percent (executables) as the maximum, the anti-malware engine for gateway with ProActive add-on enabled will require only very infrequent maintenance by network administrators. However, the gain in proactive security is tremendous and very much worthwhile. Although blocking a user-triggered, desired download is usually perceived as a false-positive, another view of the detected behavior often justifies the decision: Are anti-disassembling protection, anti-debugging tricks, and rootkit-like file-hiding behaviors suspicious? Yes, they are; and we might call them malicious, too. But when we see such behaviors by a game’s copy-protection module or multimedia software’s Digital Rights Management component—both applications themselves undoubtedly benign—do such behaviors thus become benign in general? This is the gray area of perceived false-positives at the gateway.

Whatever the perception, if a download is blocked or a website cleaned erroneously, whitelisting the affected URL takes just minutes, compared with hours or days to run a forensic analysis of an
infected computer, determine whether any sensitive data was stolen, and clean the system. The latter keeps the administrator busy and keeps the affected user from working. Depending on what data or identity might have been stolen, the financial damage per infection ranges from roughly $50 for a very simple infection to up to millions of dollars when corporate officers’ PCs are compromised by information-stealing malware. When Google was attacked in the course of “Operation Aurora,” for example, the attackers gained access to Google’s source code repository and stole the code of its proprietary, single sign-on authentication system for Google Mail, Google Apps, and others.[3] Although only Google can know the exact cost of this breach, it surely was nowhere near the $50 end of the spectrum.

Today’s Advanced Malware Detection Technologies

Automated Intelligence

At McAfee, threat information—such as samples of new potential malware or suspicious websites—can be automatically submitted from endpoint nodes and appliances; analyzed, classified, and correlated; and incorporated into database updates. It can also be made available real-time through Artemis in-the-cloud protection.

All this happens automatically, 24 hours a day, 7 days a week, generating a multitude of protections from the simplest fingerprints and signatures for individual files, through generic family descriptions for similar samples, to advanced behavioral patterns correlated from extracted common behavior traits of various malware samples of different families.
**Risk Mitigation**

**False-Positive Prevention**

Our automated intelligence, which feeds malware detection data into an anti-malware engine’s next database update, includes both “positive” and “negative” training. Positive training means new malware detections are learned, while negative training refers to the automatic removal, or reduction in aggressiveness, of detection rules that could lead to false alarms.

Gateway-specific anti-malware technology also takes the user’s surfing behavior and website context into account. Depending on how a user reached a website, or an executable download, impacts its classification. See “Contextual” for details.

**Code Security**

A second area of risk that enterprise-class anti-malware engines must protect against is the potential of new software code to introduce bugs and thereby potential security vulnerabilities. At McAfee, we apply coverity™ industry-leading code analysis tools and processes throughout the complete product lifecycle. Plus dedicated teams perform security audits to ensure the McAfee anti-malware engine code meets the highest levels of security standards. Finally, the Authenticode™ digital certificate of the engine binary allows users, and products, to ensure the engine has not been tampered with.

**Cloud-Based Detection**

The enormous growth of malware families and variants has led to a parallel growth of anti-virus detection databases. McAfee Artemis cloud-based detection technology allows us to keep a vast amount of detection rules in the cloud, thus reducing the in-memory footprint of the anti-malware engine, while still allowing access to all global anti-malware intelligence. Without a cloud technology such as Artemis, database updates would need to be about 600 percent larger than they are today, and they would no longer efficiently load into memory.

From a protection point of view, cloud-based detection further allows us to correlate global events, such as outbreaks of new, suspicious files, and provide protection in real time.

*McAfee Artemis cloud technology tracks an outbreak of a new malware.*
Emulation

Emulation is the anti-malware engine’s ultimate weapon against polymorphic malware, and for accurate proactive detections in general. Polymorphic malware changes the encryption method and key, with which it encrypts the original virus code, with every replication step, but the original virus code/body stays the same. Bypassing any encryption and peeking under the surface allows our generic detection to spot any polymorphic malware family.

Emulation means the engine simulates a virtual computer—its CPU, memory, operating system API and resources—and simulates execution of a suspect file in this virtual environment. The suspect file’s code is disassembled, just as an operating system would do it, but the instructions have effect only on software-simulated, safe data structures. This disassembly helps us gain a high level of insight into the suspect file, allowing for an accurate threat prediction. Emulation also is the base technology for Generic Unpacking, which we outline in the next chapter.

An emulator has to cope with the many features of the simulated computer and operating system, such as processes and threads, files, and anti-debugging and anti-emulation tricks. Many anti-emulation techniques exist: Malware authors have used undocumented CPU instructions (for example, salc, icebp) to fool emulators and, as new instructions are added to chips, these new instructions are quickly picked up by new malware variants, too. Examples here include new MMX instructions (used by Downloader.zq and Downloader.ash variants, for example) and FPU instructions (W32/Alisa and W32/Sabia virus families, for example). Another common technique on Windows is the use of timing checks—checking the execution time a set of instructions takes and either exiting or following a different code path if the timing check fails (W32/Volage and W32/Alisa virus families, for example). Also on Windows, using Structured Exception Handling (SEH), Thread Local Storage (TLS) callbacks, and any uncommon Windows API functions are popular techniques to try to defeat an anti-malware engine’s emulator. The McAfee anti-malware engine has facilities to deal with such problems and our researchers keep abreast of the latest developments in this area.
Unpacking

Malware authors often wrap their malicious payloads with runtime packers. More than 200 packers and protectors are known. Runtime packers are legitimately used to minimize the download size of an executable, but in many cases they are employed to add an obfuscation layer over the program code. Once started, the executable runs a decoder loop that opens the packed section and then transfers execution (in memory) to the unpacked section.

Seeing a runtime-packer in use is not enough evidence to classify a program as malicious, because benign legitimate applications also use packers. Nonetheless, the presence of a packer can raise a yellow flag in an anti-malware engine. When other suspicious behaviors are encountered, the combination results in a heuristic classification as malware. Runtime-packed executables expose some obvious geometrical facets, such as the information entropy of sections, the presence of decoder loops close to the executable’s entry point, or jumps that transfer code execution into writable (data) sections. The chart at right, for example, shows the distribution of information entropy along the sections of Worm.NetSky.C, a variant that was runtime-packed with the ASPack packer. Reading from right to left, execution starts at the end in the .aspack section, which is not packed and registers medium entropy in the chart. But that section contains the decoder loop that unpacks the malicious content in the three sections to the left. Due to being packed, these sections show much higher entropy. After unpacking, the execution moves into the (now unpacked) .text section.

An anti-malware engine should include support for unpacking such prevalent packers such as ASPack and many others (UPX, MEW, and FSG, to mention a few). The McAfee anti-malware engine goes beyond this step by using our unique generic unpacking, which allows the engine to unpack even unknown, custom runtime-packers—thus supporting proactive, and accurate, classification even on hidden or new packers.

Malware today uses many other forms of content obfuscation, largely to attack static/gateway-level scanning, beyond runtime packing. For example, obfuscating an exploit’s executable payload with XOR, ROL, and other simple algorithms (used by the Operation Aurora attack); base64 encoding the executable payload and placing it directly into an HTML page (used by the widespread Gumblar infection of legitimate websites); or by placing an obfuscated version of an executable payload behind a GIF or JPEG stub to make them look like...
legitimate media files to a network-level scanner (used by the Mezzia Trojan downloader).

The “proven” concept of packing malicious code to hinder analysis by scanners and human analysts has long been popular. Even real encryption, using the RSA cipher algorithm, is already in use by web malware such as the MaxSploit toolkit. Script code obfuscation has also been enriched with anti-emulation tricks, such as moving the obfuscation key away from the content using the document’s URL (location.href) or HTTP header values (the Fragus toolkit uses last-modified header) to carry that piece and have it available for reconstruction at runtime.

**Statistical Classification**

Heuristic detection techniques compare properties, or facets, of a new file against similar facets harvested from our broad view of prevalent threats, to determine a match with a proactive level of fuzziness. Facets are any basic properties of a file, varying with the file format, or even some smaller signatures. Some facets are of a geometrical (or structural) nature, such as the information entropy we discussed in the previous section. Others range from a simple file-size property—which applies to any files—to advanced Authenticode™ digital certificate checks, which would apply only to Windows executables.

The ongoing “commercialization” of malware leads to the reuse of code portions or even of large parts of a binary, with only small data areas being changed with every variant. An anti-malware engine’s ability to heuristically associate new malware variants with known malware families, based on file facets and statistical matching, makes a strong stand against this trend, especially in the area of Trojans and adware and spyware. Malware families such as Zlob and the various SpySheriff rogue (fake) anti-virus software, for example, are notorious for disgorging new variants in short intervals.

The detailed view that McAfee has into the threat landscape allows us to apply machine-learning algorithms on these huge malware sample sets, and to compile databases—so-called decision trees—describing the current malware families and allowing statistical matching of any new suspect files against this form of threat intelligence.

**Fuzzy Fingerprinting**

The anti-malware engine supports a number of advanced, patent-pending fuzzy fingerprinting techniques. In contrast to exact hash identification of just one malware file, these fingerprints can
also match variants thanks to their fuzziness. Examples are the use of chains of opcodes, the use of graph data, and others to build such fingerprints. On web gateways, the engine also uses geometrical fingerprints.

**Generic Exploit Detection**

Whether a targeted attack against your enterprise or a fast-spreading worm such as Conficker, exploits are often the root of all evil. In contrast to executable malware files, exploits hide in inconspicuous data formats such as JPEG, MP3, DOC, or PDF. One of the earliest techniques for generic exploit detection was “x-ray scanning,” a cryptanalysis approach. X-ray scanning is the brute-force probing of a set of known obfuscation methods, such as XOR and ROL, searching for a set of known signatures. This obviously hinders scalability in both performance (support for more obfuscation methods slows it down) and protection (signature based).

The patent-pending McAfee XploitSeeker exploit detection, part of the ProActive add-on detailed in the next chapter, takes a different approach—with a focus on scalability. It performs real-time content inspection and virtual machine emulation to detect potential zero-day or targeted exploit attacks before they get to client PCs. XploitSeeker probes content for potential machine code behavior, and for this it does not use any signatures but leverages a statistical method called Markov Models. Machine learning is used to train the models. This detection approach also is file-format agnostic, further broadening its scope. [4]

In the first half of 2009 alone, XploitSeeker has prevented infections by the most dangerous zero-day vulnerabilities seen in the wild, including Internet Explorer Buffer Overflow Zero-Day (MS09-002), Adobe Reader getIcon() Stack Overflow Zero-Day (CVE-2009-0927), PowerPoint Code Execution Zero-Day (MS09-017), and DirectShow Video ActiveX Zero-Day (MS09-032). With anti-malware protection from McAfee, both sides—endpoint and gateway—implement unique detection approaches that are optimized for their respective environment. By deploying both endpoint and gateway protection, these technologies add up, without simply running the same scan twice.
Behavioral Analysis

At a glance, two forms of behavioral analysis exist in the world of malware detection: dynamic analysis and static analysis. The dynamic variant runs on the desktop, and monitors application and system activities at runtime. If malicious, or suspicious, activity is detected, the potential threat is stopped just in time. The features of an endpoint anti-malware's host intrusion prevention system (HIPS), such as runtime buffer-overflow detection, fall into this category.

The static variant parses a file on disk or in memory, inspects its code portions, and makes best-effort assumptions about the most likely runtime behavior that the file would expose if executed on a real computer. This approach is mandatory for gateway-level use. Our technology for performing static behavioral analysis on web gateways is ProActive, outlined in detail in the following chapter. ProActive leverages the emulation and statistical classification techniques described before.

ProActive: Behavioral Anti-Malware Add-On for Gateways

All our anti-malware technologies aim to be predictive in their analysis and proactive in their actions. The patent-pending McAfee ProActive technology in particular provides web gateway-specific anti-malware features, described by its four unique technical aspects.
how downloaded files and visited websites would behave when executed on the client PC.

**Behavioral**

ProActive focuses completely on behavior traits in downloaded content because behavior is the fundamental building block in any software—including malicious software. This focus allows McAfee to proactively detect common malicious behaviors such as password stealing, backdoor communications, or viral replication, regardless of the malware variant, and before they reach the desktop.

Behavior traits are automatically extracted from samples of emerging threats accumulated by our Global Threat Intelligence database. The traits are correlated and our machine-learning algorithms train database updates. Incremental updates can ship within the minute. This intelligence is delivered to gateway appliances 24/7.

*ProActive real-time behavioral analysis, fueled by Global Threat Intelligence, terminating a malware's lifespan when it matters.*
Contextual  To ensure the most accurate classifications without losing aggressiveness against unknown threats, ProActive technology takes into account users surfing behavior and website context. How did the user trigger an executable download? What type of website is embedding in a script? The surfing path and click behavior, the categorization of the website, a digital signature of a download, and many more elements affect the overall classification. An odd-looking script would be allowed on a known popular and trustworthy business website, but would be blocked on an unknown site to which the user was redirected.

Outbound  The fourth aspect of ProActive gateway anti-malware technology is the patent-pending detection of suspicious outbound network traffic. This detection allows us to protect a roaming user with an already infected laptop from data leakage, because the malicious “phone home” activity—sending stolen passwords and other data—can be detected just-in-time at the network perimeter. Gateway-level anti-malware blocks the data flow and isolates the client from further web access until it is cleaned.

ProActive Technology in Action
ProActive XploitSeeker Detection  A zero-day vulnerability allowing remote code execution in the Microsoft Video ActiveX control (MS09-032), part of DirectShow, was heavily exploited in the wild in 2009. The flaw lay in a missing data validation check in the underlying Active Template Library statically linked to the affected ActiveX control (thus affecting other ActiveX controls as well).
McAfee ProActive proactively caught websites hosting this exploit as BehavesLike.JS.Infected.A because of its injected script code behavior. The shellcode that the exploit referenced, in many cases placed into the file ly.jpg, was caught proactively as BehavesLike.Exploit.CodeExec.EBEO (XploitSeeker) because it contained shellcode behavior traits that were used in previous exploits for other vulnerabilities. The final malicious payload that the shellcode was to download and install—a UPX-packed backdoor program—was also caught proactively as LooksLike.Win32.Suspicious.C, due to its geometrical similarity to a preceding variant. McAfee ProActive technology predicted the malicious behavior of all stages of this widespread attack and protected our customers from its very first appearance onward.

**ProActive Behavioral Malware Detection**

Targeted attacks against employees of Western governmental agencies have become a startling reality. In one exemplary case in August 2009, the targeted social engineering lure was designed to fool a government employee into downloading a maliciously crafted program that would spy on certain user data, open a backdoor port on the targeted computer, connect to DynDNS home servers, and receive further instructions from its remote attackers through a proprietary encrypted channel. With McAfee ProActive technology, we predicted and preemptively stopped the malicious behavior at the network perimeter (as BehavesLike.Win32.Backdoor.A), without its ever reaching the targeted desktop.

**ProActive Context Correlation**

Accuracy is essential to enabling Web 2.0 functions. Feature-rich, interactive websites are built upon complex scripts, which are often further obfuscated to protect proprietary code. The prevalent threat of
rogue anti-virus scams, which tell site visitors their computers are infected and offer to clean them, also relies on interactive-looking websites to attract the attention of their victims.

The fake anti-virus page below uses script code that is runtime-packed with the Dean Edwards script packer. The script dynamically builds a web form, filling in the target URL for the download, and then submits the form without user interaction to trigger the drive-by download. The script alone would be classified "suspicious," but would not be blocked by default. Why not? Because this way of using web forms has become very popular with Web 2.0 sites, and we don’t want to harm the innocent.

The executable to be downloaded also gets classified as suspicious due to its use of custom runtime packing, but again that alone would not justify blocking. However, now we can put three pieces together: the suspicious script, the script’s triggering an executable download without user interaction, and the executable’s suspicious appearance. ProActive blocks the threat as LooksLike.Win32.Suspicious.C before it reaches the desktop, while maintaining the accuracy required for users to enjoy the legitimate Web 2.0.

Although a variety of password-stealing families and variants exist, their behavior usually boils down to some means of hooking into processes or the operating system to capture data and reliably send the captured data home. And that requires avoiding noise at either the desktop firewall or the corporate firewall. To stay quiet, most malware uses HTTP—because it is one of the few ports that any corporate firewall would have to leave open. Password stealers also often inject their data-sending code into the browser process, so that an unsophisticated desktop firewall will see only a legitimate browser process sending the data.
These clients, identified as preparing to run potentially unwanted software, are locked out of all web access.

Even if we had not caught the malware at this stage, McAfee ProActive technology would have performed its analysis on the outbound data (the stolen passwords and user data) sent from SilentBanker, distinguished the unsolicited nature of the data submission from legitimate web traffic, and stopped the data leakage just in time.
About The Authors

Christoph Alme is the manager of the anti-malware engine R&D team at McAfee Labs. He is responsible for overseeing the research and development work of engine teams based in the United Kingdom and Germany. Alme is the inventor of several patent-pending key technologies in the field of proactive malware detection. Before joining McAfee through the Secure Computing acquisition, he worked at SAP and BMW.

Declan Eardly is a development team lead on the anti-malware engine R&D team at McAfee Labs. He is responsible for leading a core-engine team based in the United Kingdom. Eardly has 15 years of experience in anti-malware research and development, covering literally all malware topics from Microsoft Office malware to the latest advances in statistical malware classification. He has also worked for Dr. Solomon’s Software.

References


