

Spyware Profiling™

Technology today for tomorrow's threats

A White Paper Presented
by Tenebril, Inc.



Abstract

Traditional malware-fighting methods do not provide effective protection against spyware. Sophisticated spyware mutates, neatly sidestepping signature-based solutions; likewise, spyware is too complex to be ensnared by the behavior-based solutions that other malware-fighters utilize.

Based on cutting-edge research, and proven overwhelmingly successful in the field, the solution is *profiling*. Only the unique, patent-pending Tenebril™ Spyware Profiling Engine™ identifies, disables, and removes spyware threats, including mutations.

Table of Contents

SPYWARE: A DANGEROUS NEW THREAT	3
CURRENT SOLUTIONS FAIL	3
Mutation Easily Defeats Signatures	3
Limits of Behavior-based Detection	5
PROFILING: THE NEXT-GENERATION SOLUTION	5
Spyware: Similar to Spam, Not Viruses	6
The Solution in Practice	6
CONCLUSION	8

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

Tenebril, the Tenebril logo, SpyCatcher, GhostSurf and Spyware Profiling Engine are all trademarks of Tenebril, Inc. Microsoft and Windows are registered trademarks of Microsoft Corporation. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Tenebril disclaims proprietary interest in the marks and names of others.

©Copyright 2005 Tenebril Inc. All rights reserved. Reproduction in any manner whatsoever without the express written permission of Tenebril Inc. is strictly forbidden. For more information, contact Tenebril.

Information in this document is subject to change without notice.

Spyware: A Dangerous New Threat

Odds are, you're already infected. According to a study cosponsored by the National Cyber Security Alliance, at least 80% of Internet users have spyware on their computers. Spyware exposed the London offices of Japanese bank Sumitomo Mitsui to the world's largest attempted bank robbery: £220 million, according to silicon.com. And NetworkWorld reports that certain spyware sends stolen information to North Korea, where the data is analyzed and sold to criminals and terrorists. As bad as viruses are, as much overt damage as they cause, spyware might be worse—because some strains can infect the PCs in your enterprise for months and never reveal their presence.

Obviously, spyware is a huge problem. But what exactly is it? Ask a dozen IT professionals to define spyware, and you'll get a dozen different definitions.

To aim broadly enough to maximize protection, we'll define spyware as any software program installed on your PC that collects data about your behavior and sends that data elsewhere, without your knowledge or consent. Often, spyware masquerades as a more benign application (making it a Trojan Horse as well), and is near impossible to completely remove. Spyware comes in many forms: from the merely annoying "adware" that bombards you with ads, to the dangerous keystroke loggers that capture your passwords and other sensitive data.

Spyware and its subcategories are major concerns for IT managers now, and the threats are expected to grow.

Current Solutions Fail

When any new threat appears, whether real-world or digital, one of the first responses is to try existing protective measures. For example, in the war against spyware, security companies quickly pressed signature-based solutions into antispyware duty. Soon, new antispyware companies began assembling vast lists of spyware fingerprints for their products, as if the biggest list could provide the best protection.

Unfortunately, this is not the case. As we'll see, signature-based solutions are inherently limited, no matter the size of their signature databases. This is because fingerprinting is *reactive* security. It only comes into play after a malefactor has been identified. It is useless against new threats. Worse, more and more spyware can mutate on the hard drive, constantly staying one step ahead of signature-based solutions.

Another existing technology that's proven incapable of reliably catching spyware is behavior-based detection. This type of technology can succeed against some malware, but it's woefully inadequate against all but the most simple spyware.

Mutation Easily Defeats Signatures

Mutation, more than any other factor, renders signature-based solutions ineffective. Below, we'll examine common mutation techniques, and see how they easily outsmart even advanced signature-based solutions. We'll also compare true mutation to polymorphism, an easily defeated "straw man" that some antispyware vendors misleadingly hype as the ultimate threat.

Mutation Techniques

There are two significant types of mutation that threaten the PCs on your network today: *update* mutation and *programmatically* mutation. Note that the term "mutation" applies to applications that rewrite themselves. It doesn't include applications that simply change their filenames; even signature-based solutions can handle such basic threats.

Update mutation is a technique in which spyware applications automatically download application updates from the Internet and install them over the original application files. Spyware programmers can implement this technique relatively easily, which makes it the most common type of mutation.

Original File and Signature

Signature:

0110010101110101010111011011010

Malware File:

Code segment 1	Code segment 2	Data segment
0110010101110101010111011011010		

Update mutation is also very powerful. The malware's creator can change his application code in arbitrary ways—*after* infection. Unlike programmatic mutation, the creator retains full control of his distributed application, allowing him to respond to any developments in detection technology. The practical usefulness of this is based on how often the creator pushes updates. In the update-leapfrog game of spyware versus fingerprint, a signature-based product can eventually catch "lazy" spyware that changes infrequently. But more aggressively updating spyware quickly outdistances signature-based products. In fact, smart creators can upgrade their deployed spyware almost instantaneously, in order to make use of the very latest exploits. As a result, your PCs end up being hammered nonstop by spyware that stays just out of reach of your signature-based "solution."

Update Mutation

0110010101110101010111011011010		
x		x
Code segment 1	Code segment 2	Data segment
1101010101110100100111011110100		

Segment Mutation

0110010101110101010111011011010		
		x
Code segment 1	Code segment 2	Data segment
0110010101110101010111000110101		

Programmatic mutation occurs on the host automatically, without communication from the application's creator. Using this technique, spyware applications rewrite their code segments—the actual operations the software executes. For example, the rewrite might shuffle opcodes, or modify register allocation 'on the fly'. Since the resulting program code is dynamic and has the potential to be completely unique, this form of mutation entirely defeats fingerprinting.

Code Mutation

0110010101110101010111011011010		
x		
Code segment 1	Code segment 2	Data segment
1101000001011100100101011011010		

Mutation is a relatively new phenomenon in malware, but it's already grown into a critical problem. And because reusable code for spyware creators is freely available on the Internet, the problem is getting even worse: many malware creators can easily reuse existing mutation functionality when they build new spyware.

In our testing (detailed later in this document), fingerprinting techniques failed to catch about 15% of all infections, due to rapid mutation. That incidence percentage is on the rise, but in your network, where even a single missed infection is dangerous, 15% is already highly significant. For all the damage it can do to your company's bottom line and reputation, that 15% might as well be 100%.

Polymorphism: an Unimpressive Threat

Several antispayware vendors publicize their ability to stop *polymorphic* malware as the ultimate solution. But while it may sound impressive, "polymorphic" merely means any malware application with multiple strains or that chooses its name and/or location at install time, normally from a preset list. Obviously, these are not very robust methods of camouflage, and even basic fingerprinting solutions can usually handle them.

Although polymorphism mildly challenges vendors to keep their signature databases up-to-date, the problem does not require antispayware vendors to improve their detection engines. Compare this to mutation—the real danger, which stymies signature-based methods.

You can consider polymorphism a watered-down version of mutation. After all, malware that appears as multiple strains is similar to update mutation, only on a

much smaller (and, most importantly, finite) scale. Thus, any solution powerful enough to detect true mutations will also catch polymorphic variants.

Limits of Behavior-based Detection

Behavior-based detection—essentially application-level surveillance—works against certain viruses and worms. Unfortunately, though, this limited success has led other vendors to lump spyware into the same category as viruses, and simply claim preemptive victory against it without convincing evidence. Indeed, as we'll see below, application-level surveillance for spyware is trivially proven to be ineffectual, and that proof is mathematically ironclad.

In theory, behavior-based detection identifies the runtime behavior of an application. From that foundation, classification is supposed to be straightforward. Imagine that you are the malware detector: if you noticed an application that monitored a web browser and displayed pop-ups based on surfing habits, you would have no doubt that the application is adware. Because behavior-based detection examines behavior, not fingerprints, it should be effective even when confronted with mutating applications; further, it removes the ponderous burden of fingerprinting and classifying the growing universe of malware applications. If it worked, such a technology would be a "silver bullet" in the spyware space.

But behavior-based detection as described is simply not possible for all spyware. Thanks to the field of computability theory, we know the process of discovering fundamental runtime behavior of code is provably "undecidable"; that is, mathematically speaking, no algorithmic solution exists. However, against some threats—viruses, for example—it is sufficient to *approximate* a solution. In antivirus, for example, vendors use small abstract code segments, or patterns, to conclude that a file *could* do something very specific. This works to catch viruses that have compact, code-level pathologies, but does not scale to complex applications like most spyware. Another approximation strategy involves looking at the executable's working set. This approach, like the previous, shows what actions are possible, but not what the program will do in practice. Regrettably, the working sets of application-sized software, including both trusted programs and malware, are too similar to provide the necessary guidance.

Profiling: The Next-Generation Solution

As we've seen, there are currently two popular methods for detecting malware: fingerprinting and surveillance. Each has an obvious real-world analog for identifying criminals.

Unfortunately, spyware is as effective as real criminals at evading these micro and macro extremes. Unidentified spyware (whether a new spyware strain or a freshly mutated one), has no "criminal record" and thus easily bypasses fingerprinting-based identification. And like a real-world criminal mastermind, spyware can be far too complex to be caught through simple one-to-one monitoring. In the real world, what fills the gulf between forensics and surveillance? Profiling.

It's possible to assemble a list of traits that are relatively unremarkable on their own, but revealing in combination. These combinations—profiles—can spot previously unidentified spyware before it strikes. The Tenebril Spyware Profiling Engine at the core of SpyCatcher has brought this technology into the field, where it has succeeded overwhelmingly. As we've established, other technologies let threats slip through their grasp; but the comprehensive protection in SpyCatcher identifies, disables, and removes even mutated spyware more effectively than any other solution.

In the next section, we'll learn how spyware is more spam-like than virus-like, and we'll examine the comprehensive protection the Spyware Profiling Engine offers against real-world threats.

Spyware: Similar to Spam, Not Viruses

Since traditional antivirus technology doesn't offer much practical protection against spyware, the security industry has naturally begun investigating other methodologies. And although no existing malware-fighting technology can be repurposed for comprehensive protection against spyware, there is one threat that offers illuminating lessons: spam.

At a high level, spyware and spam have a number of similarities. Most notably, both threats are extremely dynamic, and thus largely resistant to signature-based solutions. In each case, its creator has the ability to train against the opposing security technology. He can change his payload in arbitrary ways, and he has technology to propagate those changes instantly (i.e., update mutations and email, respectively). Any solution that defeats these threats must respond flexibly, and must be assiduously crafted in order to avoid false positives. Critically, while the deliverable for each threat—a mutating application or an e-mail—is not identifiable via signature, it *does* exhibit certain characteristics that allow for flexible identification. And here the threats diverge: an anti-spam filter can look for simple keywords, but antispyware must track numerous other seemingly unrelated attributes. In fact, the spyware attributes might appear benign when examined singly; it is only when viewed together that the pattern—the profile—emerges.

There are lessons to be learned from anti-spammers' *implementation* of their solution, as well. By encoding detection steps as rules, anti-spam makes its inner workings easy to understand and control for IT administrators and, at the same time, very difficult to probe from the outside. A spammer, by virtue of his being outside the e-mail service provider, perceives any anti-spam system as a "black box." He must feed inputs into the anti-spam engine and receive only "spam" or "not spam" outputs, without understanding how a particular output was generated. This probing process is undirected, as a "no" output does not lead the spammer any closer to knowing how to get a "yes." The emergent behavior of anti-spam engines is the result of a complex interplay between rules, and as a result it is not possible for an outsider to correctly anticipate its decisions. As an effective antispyware solution, the Spyware Profiling Engine delivers this "black box" functionality, complete with the increasingly effective emergent behavior that arises out of the criteria of the profiles assembled.

In addition to slowing the malware creator, the profile-based architecture has another critical advantage: self-learning. Many anti-spam engines use Bayesian algorithms to tune their use of rules and evolve with spam. If an e-mail is characterized as spam, matching rules with previously low weights will be strengthened, which will catch new types of spam and ultimately strengthen other rules. Similarly, the Spyware Profiling Engine learns and adapts, becoming more efficient with every threat it stops.

With the Tenebril Spyware Profiling Engine, we have uniquely combined the best of existing solutions with a ground-breaking new approach. The following results speak for themselves.

The Solution in Practice

To test the Spyware Profiling Engine, we ran SpyCatcher on three infected machine states. We chose machine states that matched common user and software profiles. The test cases were:

- *Low security.* Running an unpatched version of Windows XP Home Edition, without Service Pack 2. Standard ActiveX security settings. The spyware mix included some worm-borne programs, and we installed a few common adware-supported programs as well.
- *Software enthusiast.* We installed many adware-supported programs, including those outside the mainstream, as well as popular pirated software from "warez" sites.
- *Aggressive surfer.* Changing ActiveX security settings to accept all signed content, we visited sites known to install spyware, including (www.iowrestling.com) and (www.007arcadegames.com), as well as high-visibility sites in the "free stuff" and "explicit content" genres.

In each case, we allowed the machines to run for some time after the original infection process in order to allow spyware to update itself, embed itself in the OS, mutate, and install partner applications. In the "software enthusiast" and "aggressive surfer" test cases, we installed Windows XP Service Pack 2 prior to infection and used default security settings. We used no other firewall software, antivirus, or antispysware software during the infection stage.

Each of the above test cases was generated specifically for this experiment, and had not been used for fingerprint acquisition or other analyses. The Internet-based white list used by the signature-based layer of SpyCatcher contained only standard entries classified prior to enabling the Spyware Profiling Engine, as part of the normal fingerprint acquisition process. It was not augmented in any way for special use in this testing. In other words, there was no cheating.

The Tenebril Spyware Research Center staff expected the Spyware Profiling Engine to play a larger role in finding spyware than just handling the aggressively mutating types; it should also handle spyware that uses update mutation slowly enough to afford some fingerprint detection, but fast enough to be outside the current version of our database. To measure the incidence of purely mutating files, we also included a test case from our archive:

- *General-case infected state.* This infection state includes the spyware most commonly found in the wild over all profiles. The Tenebril Spyware Research Center staff has created signatures for all the spyware in this machine state, and as a result rules-based detection will capture only mutating files.

To measure coverage, we considered the detection rate for executable files only (both DLLs and EXEs). We did not include data files, registry entries or other malware incidentals, as they are not within the domain of spyware profiling.¹

The results of this experiment are listed in the following table. Note that the Tenebril technology runs two different scans—both signature-based and profiling—at *the same time*. While other solutions actually require that you run them multiple times, the Tenebril technology accomplishes all its work concurrently. In testing this technology, we analyzed the results for each of these parallel scans. Any spyware that the fingerprint-based pass didn't find, the Spyware Profiling Engine *did* find and remove. Thanks to this efficient single-pass approach, no spyware remained.

¹ Notably, spyware profiling improves detection of non-executable malware elements even though it does not deal with them directly. Much non-executable detection is done automatically, by tracing connections between, for example, detected executables and registry entries, links and other non-executable elements. As a result, the increase in detection rate at the executable level has a similar effect for other data types.

Tenebril Spyware Profiling Engine Performance: Detection Results				
Test Case	Signature- Based Scan	Spyware Profiling Analysis	False Positives from Spyware Profiling	Undetected Spyware Executables
Low security	29	9	0	0
Software enthusiast	31	6	0	0
Aggressive surfer	43	8	0	0
General-case inf. state	27	4	0	0

Table 1. Detection results for representative spyware mixes. Spyware profiling is applied concurrently with the signature-based scan; files detected by the Spyware Profiling Engine do not match any entries in the spyware fingerprint database.

The results show, at a high-level, that spyware profiling succeeds, and that it plays a significant role in providing comprehensive protection.

Looking deeper, the results for the "general-case infected state" indicate that about 15% of high-incidence spyware uses a mutation mechanism that defeats fingerprinting. The Tenebril Spyware Research Center staff added all the executable spyware files from this machine to our signature database; even with full coverage, four files evaded detection through mutation and required the profiling approach. All the layers of defense in SpyCatcher were needed in order to provide 100% protection.

The other test cases also reveal valuable data. The number of files caught by the Spyware Profiling Engine is fairly stable as the total number of spyware executables fluctuates. Deeper inspection reveals that the same mutating spyware applications appear, in large part, over the various test cases. Even though the direct infection vectors were independent, the end result is a mix that, in the mutating spyware subset at least, is relatively homogenous. The "low security" test case contained the most hard-to-detect spyware executables, even though its total number of executables was low. We hypothesize that those applications that have more technologically advanced infection mechanisms are more likely to use other advanced techniques like mutation.

The results also show that the Spyware Profiling Engine augments the signature-based approach significantly. The "general-case infected state" PC had, by percentage, a lesser amount of files fall through to the profiling scan. This proves that, in the wild, spyware profiling can make up for latency in signature generation. All the features work together to provide comprehensive protection.

Conclusion

Spyware is now a major threat to the enterprise. And despite the marketing claims of some security vendors, yesterday's malware-fighting methods can not protect against today's spyware attacks. Able to radically mutate, the most dangerous strains of today's spyware easily bypass signature-based solutions. Fingerprinting schemes can perhaps claim up to 85% success against spyware threats, but that's cold comfort when you encounter any of the other 15%. Behavior-based solutions also fail, having been mathematically proven unable to monitor modern spyware due to its size and complexity.

Stopping this new threat requires a new solution. Built using cutting-edge research, and proven overwhelmingly successful in the field, that solution is *profiling*. Only the unique, patent-pending Tenebril Spyware Profiling Engine identifies, disables, and removes spyware threats, including mutations.