



# Denied Surveillance

An independent analysis of non-signature-based  
Windows anti-keylogging software

**An Informatica Security Research White Paper**  
**First Edition - April 2007**

**Author:**  
Claudiu Popa, CISSP, CISA  
President, Informatica Corporation

# TABLE OF CONTENTS

1. Summary of Findings	3
2. The Security Landscape	4
a. A new strategy	4
b. Three Solutions, Two Approaches	4
c. The Keyloggers	7
d. The Analysis	8
3. Test Results	10
a. GuardedID	10
b. KeyScrambler Pro	11
c. Keylogger Hunter	13
4. Conclusion	14
5. Resources	16

Informatica Corporation ([www.InformaticaSecurity.com](http://www.InformaticaSecurity.com)) provides information security audits, consulting, services and technology. With over 18 years in the business, Informatica is a recognized industry innovator and its certified professionals are trusted business advisors. Services include executive consulting, information risk management, complete training programs, decision support and standards-based risk assessments.

This analysis was conducted independently of all vendor companies mentioned herein. Copyrights, trademarks and trade names of Informatica Corporation include: FlexSecure Verify (audits, analysis and assessments), FlexSecure LockDown (managed security), FlexProtect (corporate security support) WorkLife Security Education. All other trademarks and tradenames are the property of the software companies mentioned herein.

# 1. Executive Summary

Our analysis of three anti-keylogging utilities, GuardedID, KeyScrambler Pro and Keylogger Hunter served to verify vendors claims and present some logical arguments for the use of each of these security and privacy protection tools.

The real value of this analysis should be derived by end users & employees whose information assets are critical. Our independent review has matched three innovative security techniques against a lineup of popular Keyloggers, the distribution of which is intended to loosely reflect their popularity ‘in the wild’.

For the purpose of this analysis, we tested tool functionality within their intended boundaries. As such, it is important to note that GuardedID and KeyScrambler are both browser add-ons and not independent programs, although they do operate at different layers within the operating system. Unlike our third tool, they are only designed to protect the data being entered to forms residing on Web pages. While KeyScrambler supports both Internet Explorer and Firefox browsers, GuardedID is currently limited to Internet Explorer.

By contrast, Keylogger Hunter has no such program limitation. It functions across all Windows applications, happily keeping keylogger logs empty. Its limitation however is in the variety of keyloggers it can defeat. As seen in the table below, it is only effective against hook-based keyloggers, essentially those that listen to keyboard input by relying on operating system functions. Although this can be viewed as a shortcoming, hook-based keyloggers currently represent the bulk of this type of software; their ease-of-implementation ostensibly contributing to their popularity.

Keylogger (/method)	GuardedID	KeyScrambler	Keylogger Hunter
AKLT /GKS	✓	✓	
AKLT /GAKS	✓	✓	
AKLT /DirectX	✓	✓	
BlazingTools PK	✓	✓	✓
BlazingTools PKFree	✓	✓	✓
AIO 2.8	✓	✓	✓
Quick Keylogger	✓	✓	✓
Keyboard Spectator	✓	✓	✓

The question that this white paper brings to light (and one that we indeed leave to the reader) is as follows: Is it preferable to adopt a general-purpose anti-keylogger that works invisibly in many, if not most circumstances? Or is it best to take a granular, layered approach to malware protection and adopt a strategy that addresses all threats within a very narrow band of use, in this case the specific instances of Web form input? In this last case, the assumption is that all other types of exposure would be addressed by other layers of protection including the limitation of access through system policies, signature-based scanning and memory-resident pattern recognition. Our results and conclusions are straight-forward and should allow readers to find the right solution for them.

## 2. The Security Landscape

Over the past decade, the security landscape has changed. New technologies have opened the door for new threats, a move towards proprietary code has meant higher corporate valuations but also a greater risk of exposure for trade secret information and from the lack of public scrutiny that open source projects traditionally benefit from. Finally, the frenetic pace of technological change has widened gaps in key areas that did not benefit from the same level of attention and investment.

Two critical aspects of security have seen little change over the past decade: The first is the use of single-factor authentication for everything from newsletter access to stock trading. This disconnect between the degree of asset criticality and the access controls used to protect it has presented attackers with irresistible opportunities, particularly by leveraging the second aspect: lack of user awareness.

Users remain the weak link in security because the focus remains on technology. This opens the door for all manner of social engineering exploits ranging from the simplest spam method of using appealing email subject lines, to direct contact with the subject. This technique almost invariably targets personal information, user access credentials and financial details – the most valuable pieces of information on the open market.

If technology can be used to increase security, then it can ostensibly also be used to increase the efficiency of social engineering, Trojan infections and all manner of attack seeking to reap benefits from the ubiquity of Internet connectivity. This has worked out so well that anti-virus companies have the unenviable task of keeping up with new malware introductions numbering in the hundreds every month. Around the clock, armies of developers are building new signatures for this software to identify all the while ignoring the most important question on everyone's mind: what about the fabled Zero-Day Vulnerability?

“Zero-days” are no longer the stuff of fiction. The past year alone has shown a number of instances that, by their nature and variety, only hint at the obvious fact that their kind routinely avoids detection (and likely pulls off successful exploits on a daily basis). While this type of software is clearly able to operate with impunity during its window of opportunity, the public continues to be victimized, and personal information theft is no longer compromised one record at a time. Millions of confidential data records are lost from organizations small and large while the powerless public is only presented with the devastating outcome of the crime.

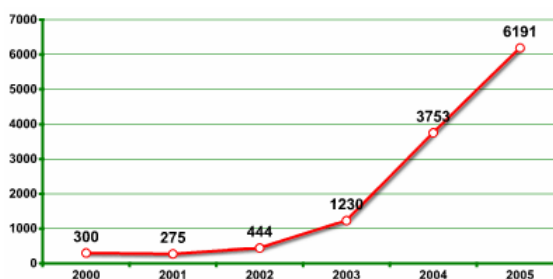
Purely academic is the question of whether the blame lies with the perpetrators, the companies that fail to adequately protect their customers or the security industry that all but creates a false sense of security. Organizations, governments

and security experts must recognize that the threat lies in apathy and ignorance while traditional reactive security techniques have proven much less efficient. While people can be socially engineered to open an email, it's more difficult to convince them to surrender their personal information and login credentials. What is significantly more effective is to capture that information during the course of their regular activities, in the safety of their usual environment and with the unfounded confidence that this data is inherently safe.

The likely scenario that unfolds on a daily basis is simple both in theory and in practice. It can be summed up in three easy steps:

1. *convince the user to infect his/her computer* by opening an unsolicited (but trustworthy-looking) message, exploiting a buffer overflow, executing Web-based code or otherwise appealing to the human instinct of needing to react (or respond) to an event.
2. *execute the malicious code*. This can be as complicated as installing an entire hacking toolkit designed to remotely take *and keep* control of the victim's computer, or as simple as installing a keystroke interception program (*keylogger*).
3. *steal the information*. For this to happen, the keylogger must be invisibly installed while still being able to interface with the outside world.

The focus is thus on the party that is doing the real work: the keylogger. These inconspicuous programs exist, right now, in hundreds or even thousands of different flavors and their number increases steadily each year (image courtesy of iDefense, a Verisign company). Some piggyback on operating system 'hooks' designed to perform legitimate keyboard functions while others use their own techniques for burrowing deep into the fabric of the operating system to undermine detection efforts.



Whatever the case may be, this type of software is designed to be inconspicuous, invisible and permanent. Key logging is a process, not a point in time event, so the problem of identifying malicious software is essentially a Catch-22: sufficient infections must exist in the wild for some to be discovered, analyzed and built into current anti-malware signatures before customers can benefit from the traditional reactive model of endpoint security. This can be a long process and given the growing effectiveness of infection methods coupled with the sheer number of infection vectors, it means that many victims must exist before the traditional model pays off. Of course, by that time the perpetrator has already been able to benefit from the attacks and all there is left to catch are the proverbial copycat criminals (of which there is never a shortage in the current security climate).

Indeed, keylogging is effective in many industries, with the focus remaining – jealous spouses notwithstanding – almost exclusively on profit potential. Over the past few years, we have seen banks, healthcare, retail, telecommunications and all manner of government departments fall prey to such attacks. Most of the ones we know about have been unsuccessful. Unfortunately they point to a disturbing trend and serve to highlight an inherent lack of preparedness on the part of the victims (the unsuspecting public), the service providers (banks, retailers, educational institutions, government and healthcare organizations) and the experts hired to protect them (traditional anti-malware companies). The fact that there are three parties involved only increases the degree of confusion, impairs accountability and further introduces delays into the mix.

### **A New Strategy**

The nature of history to invariably repeat itself allows us to recognize and anticipate trends. Just like viruses ushered in the introduction of signature-based anti-virus software (back when the number of self-replicating programs was still manageable and the risk was a fraction of what it is today), we have seen similar reactions to spam, spyware and Web-based threats. With the realization that keyloggers are here to stay and their success being often tied to their simplicity (and diversity), anti-keylogger software is seeing a surge in popularity. The awareness charge is led by none other than the large anti-malware firms whose reactive model of detection is likely to allow very large gaps in what is supposed to be a closed loop for data protection. Instead it presents an awkward approach to a problem that can only be addressed by preventative measures.

These preventative measures are based on the idea that a threat whose effectiveness is solidly rooted in simplicity can only have a countermeasure that is based on a similar philosophy. Two models exist that appear to provide an effective answer to the problem: defuse the situation by disconnecting the keylogging mechanism itself or by feeding it scrambled information. Both solutions are elegant and effective, so our real-world analysis was designed to test three available products and separate fact from fiction.

*“a threat whose effectiveness is solidly rooted in simplicity can only have a countermeasure based on a similar philosophy”*

### **Three Solutions, Two Approaches**

All three tools have free versions available for individual, personal use while offering a ‘pro’ version for business use (and more discerning users).

1. **GuardedID** from StrikeForce [v.1.02]  
This Internet Explorer browser add-on neatly embeds itself within the Windows kernel to encrypt and convert each keystroke into letters *a-to-k*.

2. **KeyScrambler** from QFX Software [v.1.2.0]  
Similar in intent and operation this browser add-on combines cross-browser compatibility with a method for elegantly encrypting keyboard input as it travels between the user's fingers and online forms. When active, it shows users the exact stream of scrambled text being produced with each keystroke.
3. **Keylogger Hunter** from Styopkin Software [v.2.11]  
Taking a different approach to fighting keylogger attempts at stealing information, this discrete Windows utility essentially incapacitates most of them by clearing the stream as keys are being pressed.

### **The Analysis**

With the clear goal of testing the products against existing, popular Keyloggers and not each other, we set out to compare their performance and verify their claims. In the process, we followed these simple guiding principles:

1. Anti-keylogger effectiveness was our main goal. The cost and business model of each solution is not material to our analysis, although we have included pricing in our review and have kept the interests of both home and corporate users in mind for test scenarios and in some recommendations.
2. We had no intention to test the tools against every possible keylogger, only a representative sample. We also did not go 'underground' in an attempt to find the latest and the greatest keylogger, pry it out of the hands of script-kiddies and expose it to our trio of purpose-built tools.
3. We ensured that all three programs were tested against the same keyloggers, under the same circumstances, in a typical Windows environment. That environment: Windows XP SP2 (latest patches installed), 512 MB RAM, 100GB HD, partially disabled security software (to allow unimpeded manual installation of these standard keyloggers). Web browsers tested: Internet Explorer 6.0 and 7.0, Firefox (Mozilla) v.2.00. This environment was recreated for each anti-keylogger. Keyloggers were uninstalled between tests.
4. The controlled environment was further verified by enumerating running processes and keylogger traces. For this analysis, it was accomplished using two utilities called Process Analyzer and Hook Analyzer. No effort was made to compare keyloggers among themselves or test their stealth abilities. The original list of keyloggers was initially longer, but due to instabilities on the part of some of these, the final list is composed of the remaining programs. Indeed, the success of malware depends on not only its ability to effectively interface with the Trojan that facilitated its introduction, but it must also benefit from a degree of quality to allow it to work in a wide variety of situations. Needless to say, this balance between built-in intelligence (read: complexity) and the simplicity needed to operate stealthily is another

challenge faced by would-be criminal hackers (albeit one that many appear to meet quite capably).

5. We made no attempt to test program functionality above and beyond keystroke interception (aside from the ability to install and execute in our controlled test environment). This means that screen scraping, screen capture, communication and alerting, capture log encryption and other features were not material in our findings, nor did they have any relation to the stated goal of evaluating the anti-keylogger tools' effectiveness.
6. Finally, we did not make any attempts at targeting these anti-keyloggers themselves in an attempt at exploiting buffer overflows or any other innate vulnerability. In the interest of full disclosure, we observed in one instance cleartext being captured in a log while an anti-keylogger was clearly active, but despite our attempts at recreating that situation, we were not able to reproduce it, so we will not comment on it here (and it is the only recorded issue that we will not comment on). Future editions of this analysis may opt to intentionally create such situations and indeed record the behaviour of these utilities over an extended period of time.

### **The keyloggers**

We selected a variety of keyloggers, all of them publicly available and offering a vast array of surveillance options. Where possible, we used the full version of the tool, or the fully functional keylogger within its evaluation period. The tools we tested against are as follows:

1. Anti-Keylogger Tester from FirewallLeakTester.com/Guillaume Kaddouch
2. Perfect Keylogger from BlazingTools (Regular and Free versions)
3. All-In-One Keylogger v. 2.8 from Relytec
4. Quick Keylogger v.2.1 from WideStep Security Software
5. Keyboard Spectator Pro v.3.3 from ReFog Software

Most of these keyloggers were hook-based. The first one, Anti-Keylogger Tester (AKLT) is a purpose-built tool that uses no operating system hooks (i.e. kernel or driver-based keylogger) and requires no installation, so it was a good complement to our analysis.

Without entering into a discussion of the benefits and drawbacks of keylogging with hook-based vs. kernel-based techniques we can say that these tools basically fall into two categories. As demonstrated by Anti-Keylogger Tester, a potential third type is based on a filter driver. This can be implemented using the installed DirectX drivers that games often use to collect keyboard activity on Microsoft Windows systems. With the clear exclusion of hardware keyloggers that can be plugged directly into the computer's keyboard port, the current selection of keyloggers easily spans over 99% of all keyboard data interception techniques.

### 3. Test Results

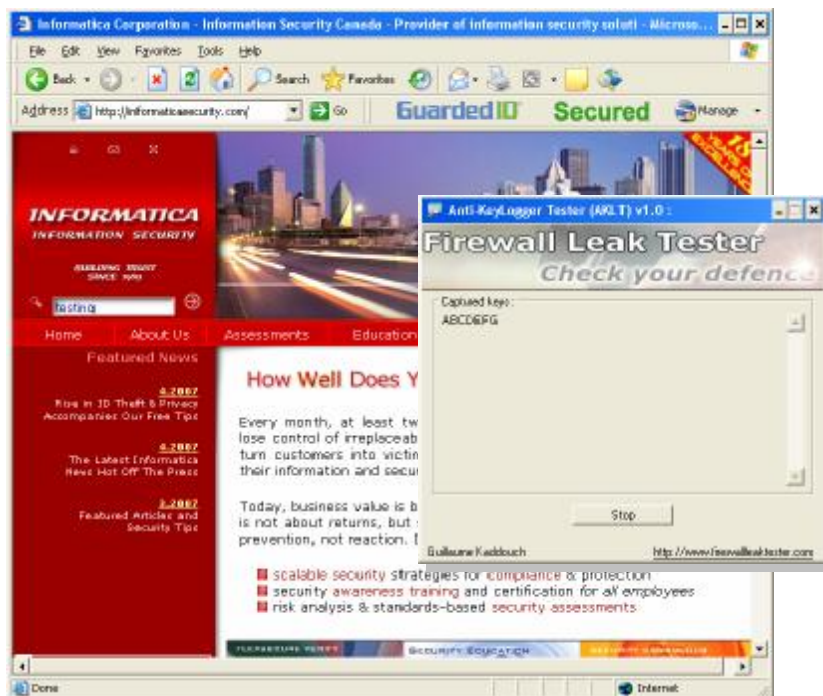
#### GuardedID from Strikeforce (\$29.99 annual license)

This \$29.99 browser add-on (free trial available) was easy to install, activate, use and uninstall against all the keyloggers we tested. When using Internet explorer, it watches keyboard input to Web forms and sequentially replaces the keys pressed with characters from A to K. According to StrikeForce, the program encrypts keystrokes entered and decrypts them as they are placed in Internet Explorer Web forms. The characters

*“GuardedID was easy to install, activate and use against all the keyloggers we tested”*

In the screen capture below, GuardedID shows that it is active (showing the word **Secured** as seen below) in the browser’s toolbar. Having just typed the word “testing” into the page’s search field, the AKLT keylogger simultaneously shows us the keyboard capture that is simply the string “ABCDEFGG”.

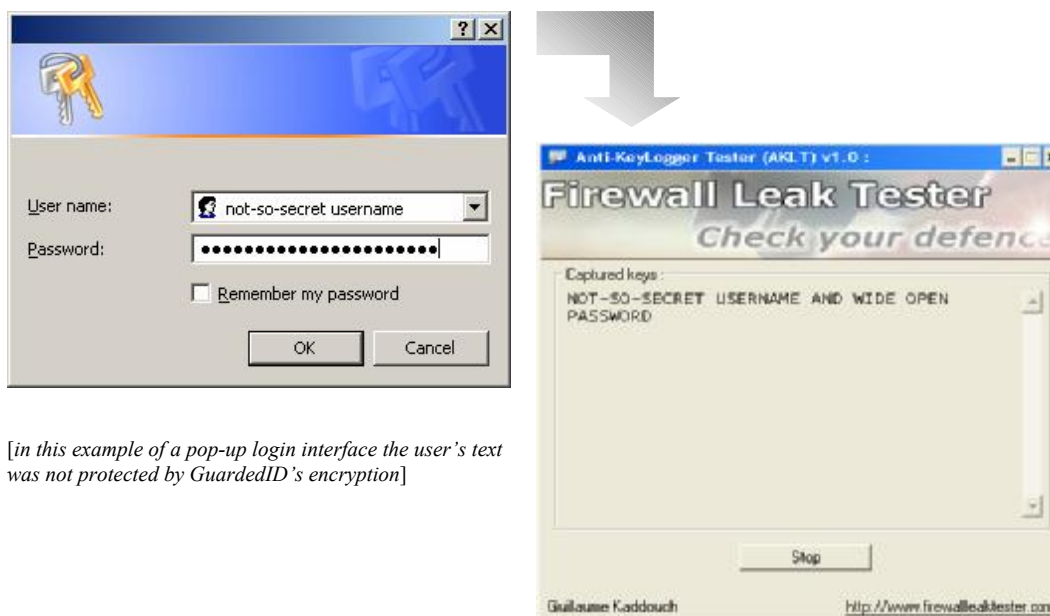
The tool works very well for its intended purpose and has performed consistently well with all the Keyloggers we tested. Much to our satisfaction, one after the other, each key log reported strings containing the letters “abcdefghijk”. The ‘manage’ button on the toolbar allows for easy access to GuardedID’s options.



GuardedID is not currently available for any other Web browser, which may reduce its popularity with a subset of the end-user population in the near term (the company has acknowledged that at least Firefox/Mozilla will be supported

in a future version). This should not impact corporate sales however, as most workstations currently rely on Internet Explorer by default.

It is important to note that, as illustrated below, the tool does not work with pop-up boxes that do not contain HTML forms. Unfortunately, there are numerous pop-up access/login for password-protected Web sites that do not use integrated forms or HTML such as those to online or custom applications that contain ActiveX, Java and Flash. Understanding the difference between what is and isn't supported can lead to confusion on the user's part. Fortunately, GuardedID's CryptoColor feature intuitively changes the form's background color (to a customizable default of *green*) to show when text input is safe from prying eyes.



[in this example of a pop-up login interface the user's text was not protected by GuardedID's encryption]

GuardedID's ability to mask keyboard input does come with some drawbacks. One is the fact that there is a 1:1 relationship between the length of the input and the size of the captured string. This can allow attackers to narrow the criteria used to brute force or guess passwords. It is naturally much easier to guess a password whose length is known, in particular if the attacker manages to correlate this information with any other tidbit that can be gained through social engineering or inference. Adding to this problem is the fact that certain keys were not masked or blocked at all. As illustrated above, keys such as Space, Backspace and Tab were simply left blank in the keylogger's capture and clearly facilitate the task of figuring out the length of login credentials.

Finally, GuardedID's protection is nicely implemented with only the above drawbacks needing to be mentioned. One additional feature is GuardedID's protection against unauthorized deactivation: while it can be set to automatically start with the browser, it cannot be disabled without closing Internet Explorer. Implementers need to be aware of the spectrum of applications and Web sites used in the course of a user's day to anticipate interaction with any custom applications and sites that may not fall into GuardedID's protection envelope.

## KeyScrambler Pro from QFX Software (\$24.99 one time)

We had the opportunity to also test KeyScrambler (free for personal use, \$24.99 for the Pro version) which also uses encryption to protect the confidentiality of text input. The screen capture at right shows that input text bears no relation or pattern as compared with the encrypted text shown in the green bar.

To increase user comfort, KeyScrambler displays this green bar anytime protected text is input on a Web page.

The location of the bar can be changed with one click. The program also displays a dynamic icon in the system tray anytime a browser window is open to indicate that it is functioning. Right-clicking the icon conveniently brings up a small menu with the program's control options.



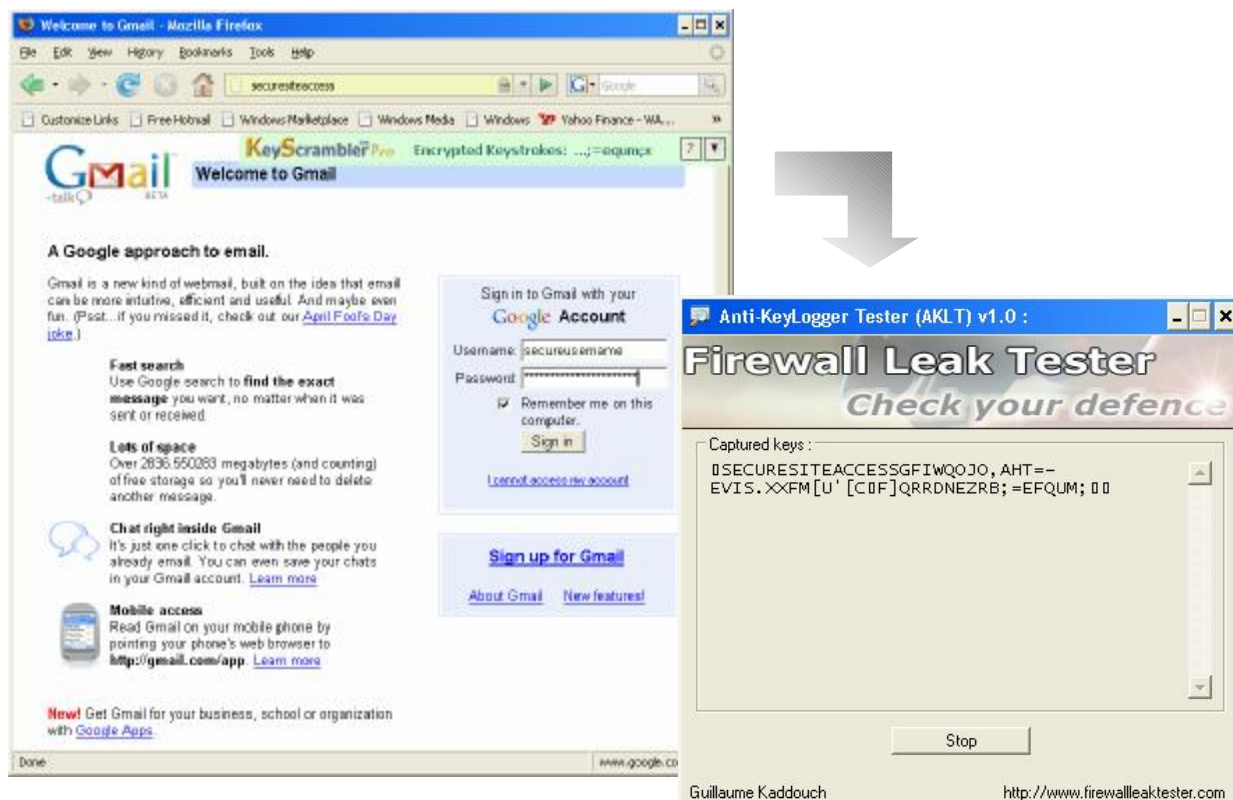
*“KeyScrambler’s coolest feature is by far its ability to encrypt and decrypt keyboard input while showing its work”*

KeyScrambler’s coolest feature is by far its ability to encrypt and de-encrypt keyboard input and to show us what it’s doing in real time. We also enjoyed the program’s support for the popular Firefox browser which is implemented identically and just as capably as for Internet Explorer.

Although Keyscrambler operates as a filter driver within the Windows kernel it does not intercept keyboard data from the same ‘depth’ as GuardedID, but when put to the test, KeyScrambler defeated all our tested keyloggers. Theoretically, and as demonstrated to us by a purpose-built StrikeForce program, it is possible for a keylogger to operate below Key-Scrambler’s watchful eye while being successfully intercepted by GuardedID.

A simple check of keylogger logs showed scrambled text for all Web forms including those implemented with active content. Interestingly, the same issue that we observed with GuardedID is present here: keys such as Space, Backspace and Tab are not encrypted (and simply show up as spaces). A hacker can thus see the visited site (since non-form input is not protected) and the length of the input credentials, likely reducing the brute forcing or guessing effort required.

Bug-wise, we did observe, on very rare occasions (over a one-day testing period) that the de-scrambling function did not instantly work. On one occasion, switching the focus from a browser window to an email window and starting to type produced scrambled text in the email's body. The issue was easily fixed by switching back (i.e. repeating the procedure).



Worth noting in the above captures is the fact that while the username and password were adequately scrambled, the logged output is close, but does not exactly reflect KeyScrambler's version: The green KeyScrambler bar shows the string “;=equm;x” while our handy AKLT real-time keylogger reports “=efqum\_\_”.

Although KeyScrambler's functionality is, like GuardedID's, mostly limited to Web forms and most pop-up windows (i.e. .htaccess) will be unprotected by its encrypted algorithm, it is the only one of the two whose functionality covers ActiveX controls, Java applets and Flash forms. These are increasingly popular methods of creating active content for interactive Web sites and banking sites.

Finally, we were impressed with this browser add-on for multiple reasons:

1. home users can enjoy the free Personal version
2. both home and business users will find that the Pro version carries the lowest price of the three tools we tested
3. KeyScrambler Pro expands the layer of protection to include some of the more interactive sites, in particular those that use Java and Flash interfaces.

## Keylogger Hunter from Styopkin Software (\$39.99 one time)



Another tool that we would recommend for home users is KeyLogger Hunter. This handy memory-resident utility sits in the System tray and works completely invisibly on all keyboard input. The program is available for a free, unlimited trial for personal use. The \$39.99 cost eliminates the nag screen and makes it even less conspicuous.

Although we liked the fact that it worked quietly and effectively in the background, we enjoyed opening the logs of our keyloggers to find them completely empty.



Aside from actually testing it, the only indication that Keylogger Hunter does anything at all is indicated by the presence of its icon (above, right) in the System Tray. Right-clicking the icon presents a few program options.

The effectiveness of the tool is, as clearly stated on its Web page, limited to hook-based keyloggers. As such, it worked very well against the majority of the keyloggers we tested, but failed against Anti-KeyLogger Tester's hook-less approach and its DirectX driver-based technique.

*“we enjoyed opening the logs of our keyloggers to find them completely blank”*

Because Keylogger Hunter is a Windows utility rather than a Web browser add-on, it functions whether or not the user is accessing a bank site or typing secret recipes. The tool's ability to fool hook-based keyloggers does not adversely affect any running programs and we did not observe any significant processor load. The Keylogger Hunter process is clearly labeled and visible in the Task Manager, using less than 5MB of RAM.

Although this elegant and effective utility has defeated the hook-based keyloggers we exposed it to, we believe that it should be used to supplement other anti-malware and in particular anti-keylogger strategies. As such, we tested it in conjunction with each of the other two tools with no adverse effects. The programs do not overlap and are happy to operate simultaneously.

While Keylogger Hunter is unable to protect against driver and kernel-based keyloggers, both GuardedID and KeyScrambler Pro fall short when it comes to protecting keyboard input other than within Web pages, and in particular, in the aforementioned pop-up login boxes. The combination of these techniques provides an effective, if not complete protection strategy.

## 4. Conclusion

While this analysis served to prove that all three products worked almost flawlessly in their intended and well-defined situations, it is important to note that in the real world, a combination of solutions is necessary to combat the keylogger threat.

The increasingly blended approach to malware infection means that only rarely does such software operate independently. Keyloggers don't typically install themselves, they depend on customized Windows installers, Trojans, scripts hidden within the bodies of emails and within Web pages, et cetera.

As such, the prudent approach as dictated by proven security best practices is to adopt *security in depth*. This simply means that in addition to the excellent tools we tested, users and companies should consider adopting the following (see the Resources section of this document for sources of some of these tools):

1. Signature- and pattern-based anti-malware. Traditionally called anti-virus software, these solutions have evolved over the years to recognize many different types of code, including malware that changes as it replicates (polymorphic), spyware and various types of digital content considered suspect
2. Outbound firewalls. This type of host-based access control is essential in the current reality where an underground economy sees infected computers as soldiers in armies of zombies remotely controlled for profit without their owners' authorization or knowledge.
3. Registry protection. Many keyloggers install themselves deep inside Windows and upon reboot, scramble to be one of the first to be loaded in an effort to evade detection. Simple programs and features built into anti-spyware can effectively protect against such exploits (while keeping users informed).
4. Rootkit protection. The rootkit can be the hacker's ideal, multi-functional malware tool whose key feature is to burrow so deep into the operating system's core that nothing can detect it. Numerous specialized programs are becoming available to detect these and in many cases they can supplement similar functionality that is making its debut in the latest versions of signature-based anti-malware.
5. Password managers. At the core of the problem remains the lowly password. A dated but capable access control method when carefully selected and managed, this simple string is the real target of keyloggers and their owners. A good password manager will help select and remember passwords, but most importantly it can help defeat keyloggers by allowing users to load

passwords into memory and pasting them without pressing Ctrl-C (a common trigger mechanism for some advanced keyloggers) or having to type login credentials at all.

Without such measures we can imagine a situation where despite the complementary use of both Keylogger Hunter and either GuardedID or KeyScrambler, a keylogger would succeed in stealing confidential information. This would simply have to be a kernel or driver-based keylogger that would capture everything except content entered into Web forms. The amount of content captured may even be sufficient to provide critical contextual clues to guess the 'scrambled' login credentials whose string length is clearly visible in the capture logs.

The scope and functionality of browser add-ons may seem like a limiting factor, but it is important to note that the most valuable pieces of information remain access credentials and most of these are entered into Web pages. By going where the threat is, GuardedID and KeyScrambler avoid potential conflicts with other software and, perhaps critically, allow corporate environments to tailor their security strategy with greater precision. In effect, these tools can be a very effective solution in environments where employers use surveillance software to monitor computer use simply because they continue to protect user credentials without interfering with the monitoring process outside Web forms. Similarly, Keylogger Hunter can be used in conjunction with kernel-based keyloggers, screen monitoring programs and other methods to provide essential surveillance without compromising confidentiality.

And confidentiality is really what it is all about.

## 5. Resources

1. StrikeForce <http://www.sftnj.com>
2. QFX Software <http://keyscrambler.com>
3. Styopkin Software <http://www.styopkin.com>
4. ZoneLabs <http://www.zonealarm.com>
5. PasswordSafe <http://passwordsafe.sourceforge.net>
6. Spybot S&D <http://www.safer-networking.org>
7. Microsoft RootkitRevealer <http://technet.microsoft.com>
8. Kaspersky Anti-Virus <http://www.kaspersky.com>
9. Grisoft AVG <http://www.grisoft.com/>
10. McAfee <http://www.mcafee.com/>
11. Symantec <http://www.symantec.com/>

## For More Information

Call: (416) 431-9012

Email: [info@InformaticaResearch.com](mailto:info@InformaticaResearch.com)

Visit us on the Web: [www.SecurityandPrivacy.ca](http://www.SecurityandPrivacy.ca)

Informatica Corporation  
67 Yonge Street  
Toronto, M5E 1J8  
Ontario, Canada