

# Leaktests as a Measure of Outbound Protection

---

*An Agnitum White Paper*



**OUTPOST 2008 PRODUCT FAMILY: 100% PASS RATE FOR ALL EXISTING LEAK TESTS**

*January 2008*

## Contents

Abstract.....	4
Points to remember.....	4
The firewall as a key security component.....	4
Introduction to the firewall's functionality.....	4
Windows-based systems – the vulnerability within.....	5
Configuring the firewall.....	5
Tools to test the firewall's vigilance.....	7
A look inside Outpost functionality.....	7
Leak tests overview.....	9
Ways information can leak.....	9
Leak technique #1: Filename/identity spoofing.....	9
Leak technique #2: Trusted process launch with parameters.....	9
Leak technique #3: Exploitation of trusted firewall policies.....	10
Leak technique #4: DNS request spoofing.....	10
Leak technique #5: Component injection.....	10
Leak technique #6: Interaction through Windows messages.....	11
Leak technique #7: Process injection.....	11
Leak technique #8: Using DDE and OLE automation to control applications.....	11
Leak technique #9: Deployment of a new network protocol.....	12
Leak technique #10: Process ID obfuscation.....	12
Leak technique #11: Removing interception functions.....	12
The leak tests.....	13
Methodology.....	13
1. Atelier Web Firewall Tester (AWFT).....	14
2. BITStester.....	15
3. Breakout.....	16
4. Coat.....	17
5. Copycat.....	18
6. CPIL v1 (Comodo Parent Injection Leak Test).....	19
7. CPIL Suite.....	20

8. DNSstest ..... 21

9. DNStester ..... 22

10. FireHole ..... 23

11. FPR (Fake Protection Revealer) ..... 24

12. Firewall Leakage Tester ..... 25

13. Ghost..... 26

14. Jumper ..... 27

15. OSfwbypass ..... 28

16. PCAudit v1, PCAudit v2 ..... 29

17. PCFlank Leaktest ..... 30

18. Runner..... 31

19. Surfer ..... 32

20. Thermite..... 33

21. TooLeaky ..... 34

22. WallBreaker..... 35

23. YALTA (Yet Another Leak Test Application) ..... 36

24. ZAbypass ..... 37

Independent test results..... 38

Conclusion ..... 38

Contacts ..... 38

## Abstract

This document details how the Outpost Pro 2008 product family protects against the sophisticated types of data breaches enabled by real-world malware. The focus is on attack techniques and the tools used to test firewalls' outbound strength. It describes how the latest version of Outpost Pro is pitted against these leak tests to measure its effectiveness in preventing unauthorized disclosure or theft of personal information.

The document also provides a brief introduction to firewall's functionality and a short overview of today's threat and vulnerability landscape.

## Points to remember

- All Windows-based systems are vulnerable and require third-party protection
- Firewalls guard your data by controlling what is allowed to be sent and received from the Internet
- Data can be sent indirectly, bypassing standard firewall barriers
- Many techniques exist that can be leveraged to steal user data; these techniques are increasingly adopted by malware authors
- Leak tests are non-destructive testing utilities designed to determine the strength of the firewall's outbound protection
- Leak tests simulate outbound breaches to determine if the firewall can detect and prevent these attempts
- Outpost Firewall Pro and Outpost Security Suite Pro software have passed all existing independent leak tests with flying colors
- Give serious consideration to who you trust to provide your online security

## The firewall as a key security component

Personal data stored on home and small office computers is at risk of being stolen and misused by cybercriminals. If not properly protected, every digital asset is at risk: passwords and login information can be captured by a keylogger, confidential files and documents can be exposed by a Trojan, people's identities can be co-opted into a botnet army and turned against the other Internet users.

Antivirus, antispymware and other signature-based solutions are a necessary component of an overall protection portfolio. However, to fully protect one's computer against the full spectrum of Internet-borne threats, an element of proactive protection is also needed to complement existing defenses and proactively tackle attacks from new and unknown sources.

## Introduction to the firewall's functionality

The firewall's primary task is to safeguard the integrity and confidentiality of your data, controlling what is allowed to be sent over and received from an unsafe network such as the Internet. By serving as a virtual checkpoint for transmitted data, the firewall is able to screen incoming and outgoing traffic, allowing good connections, blocking bad, malicious or incompliant connections, and suspending unknown connections until the user can provide direction as to whether the connection is good or bad. Good and bad connections are classified according to firewall rules, also known as access policies.

The firewall, in contrast to antivirus, doesn't need to identify threats before it stops them – it simply awaits the user's instructions as to whether to allow or block a particular connection initiated by a new program, and then creates rules based on the response. This approach, which is not dependent on a specific threat signature and presumes all connection attempts to be guilty until proven otherwise, can

stymie any Internet-propagating malware before it can land on a PC or harvest vulnerable data from it. This is an example of proactive defense.

A robust firewall must control the transmission of data in both directions – in and out. The firewall that comes with Windows Vista does not have outbound filtering enabled by default, and XP's firewall doesn't have outbound filtering at all, so data can be removed from the confines of the PC unchecked. Failure to keep tabs on outbound connections may lead to private information getting into the wrong hands and being used by the bad guys to commit a crime against you or other innocent netizens. Naturally, this hole is being heavily exploited by malware that's programmed to steal data from PCs or use the compromised computer's resources to harm other users.

The strength of outbound protection in firewalls may vary from product to product. Alongside an authorized program sending data directly on its behalf, a hostile application may attempt to deceive a firewall by sending data indirectly, posing as a trusted program or using the latter's credentials to establish outbound access.

Neither the Windows firewall nor a number of other well-known commercial firewalls are able to deliver total coverage against these advanced techniques, but as for Outpost, that's a different story.

## Windows-based systems – the vulnerability within

Windows XP's design enables programs to freely interact with one another and use each other's credentials to access the Internet. This is based on the premise that most users log in as Administrators, the highest level of privilege available, which enables them to perform any activity they want. Of course, this means that malware, when executed from an Admin account, inherits the same level of access rights as the authorized user and can modify the system in whichever way will benefit its author, including changing registry entries, writing to restricted areas of disk, loading and sharing infected components, hijacking legitimate processes, and interacting inappropriately with Windows' bundled networking services.

With Vista, Microsoft has made significant strides in strengthening system isolation against malicious interactions. By default, Vista lowers the privileges of any process (even if the process is run from an Admin account) to the lowest possible level and operates in this mode. If the process requires higher rights to perform some function, Vista's User Account Control (UAC) activates and displays a message asking the user to approve these higher-level rights. This reinforcement, along with other measures such as restricting processes with lower ID from interacting with higher-privileged processes and IE Restricted Mode have significantly decreased system vulnerability to malware but have not fully eliminated it. The downside of UAC is that it cannot store and recall user responses to legitimate actions such as viewing system properties in the Control Panel, so users must manually interact with the control process every time a potentially damaging action is attempted. Naturally, many users find this irritating and disable the feature or simply click on "allow" every time the message appears. This effectively negates all the positive aspects of UAC. Additionally, a number of advanced inter-process communication techniques, such as interaction through OLE automation, are not monitored by Vista's defense mechanisms, again reinforcing the need for third-party solutions to plug the gap.

## Configuring the firewall

Let's take a look at the types of protection users get with different firewall configurations:

**Scenario 1: Windows Vista firewall with outbound filtration OFF (the default setting for Vista; XP does not have outbound controls)**

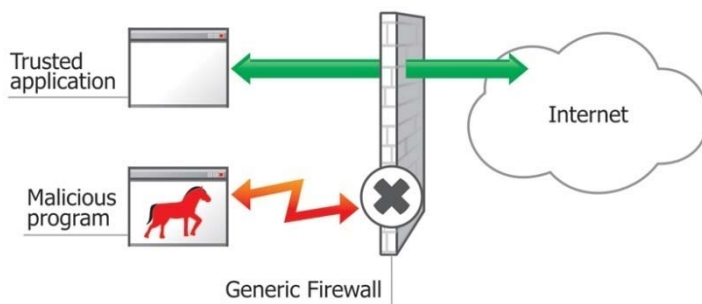


Incoming traffic is known to be legitimate because the firewall controls inbound connections and ensures they are not malicious. Outbound traffic is not controlled at all, exposing the system to outbound threats. Any malware (such as the Trojan horse in this illustration) can transmit data to external sources through this weak firewall configuration.

**END RESULT: firewall is easily penetrated and user data compromised**

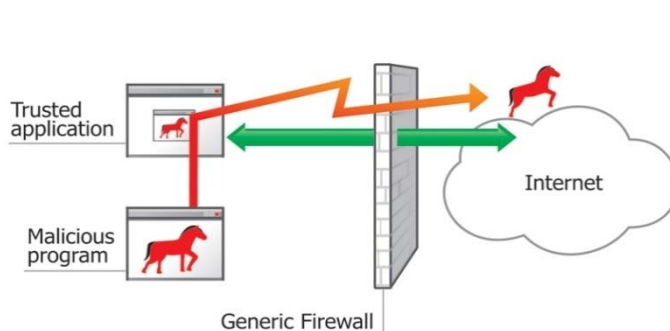
**Scenario 2: Vista or other basic firewall with outbound filtration ON**

a) Malicious programs cannot establish direct outbound connections because the firewall blocks this route



This type of firewall performs basic filtration of outbound traffic, guarding against attempts to directly send data (the firewall checks programs' permissions and blocks noncompliant connections). Malware that is unsuccessful in transmitting data directly because of the firewall's barriers will attempt to use other methods of outbound propagation (see figure b).

b) Malicious program succeeds in transmitting data by hijacking a trusted application and using its permissions to access the network

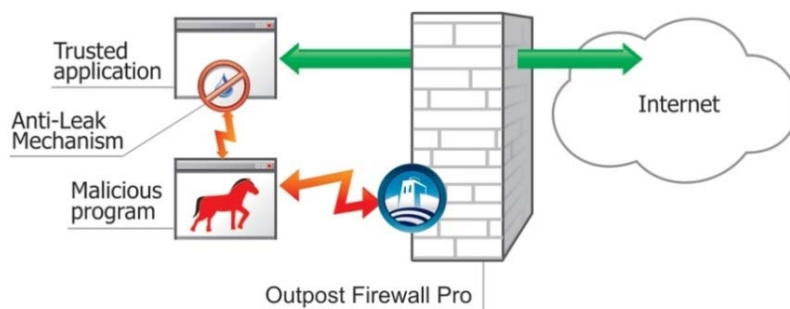


If the direct route is blocked by the firewall, the malicious program will attempt to hijack a trusted application and use its credentials to transmit data off the machine. This method makes the firewall believe a legitimate application is requesting outbound access.

Basic firewalls are unable to detect such activity and as a result render machines vulnerable to this type of indirect data transmission attempt.

**END RESULT: while the firewall foils the direct attack on the first pass, it is unable to prevent hijacked applications from accessing the Internet and transmitting data, compromising the user's privacy.**

### Scenario 3: Outpost Pro or other advanced firewall with bidirectional filtration and monitoring of system activity



Outpost's advanced anti-leak functionality detects and prevents not only attempts by malicious applications to send data off the machine directly, but it's also on guard against illegal or unwanted inter-application activity, ensuring that malware cannot interact with trusted programs and use their credentials to transmit data off the PC indirectly. In combination, direct outbound control and control of applications' activity deliver multi-layered protection against malicious leakage of personal information.

**END RESULT:** malware tries direct outbound transmission and fails, tries indirect transmission by hijacking a normal application and fails again. The user's data is secure.

Later in this document, you'll find more details on the types of techniques used by hackers to try to sneak data past the firewall's outbound defenses and how Outpost Pro defeats each and every one of them.

## Tools to test the firewall's vigilance

As mentioned earlier, firewalls should be able to monitor programs for outbound activity. Even if a program tries to masquerade as another application that's been pre-configured with the firewall as "trusted", a competent firewall should be able to detect such application hijacking and prevent data from being transmitted off the PC in this way.

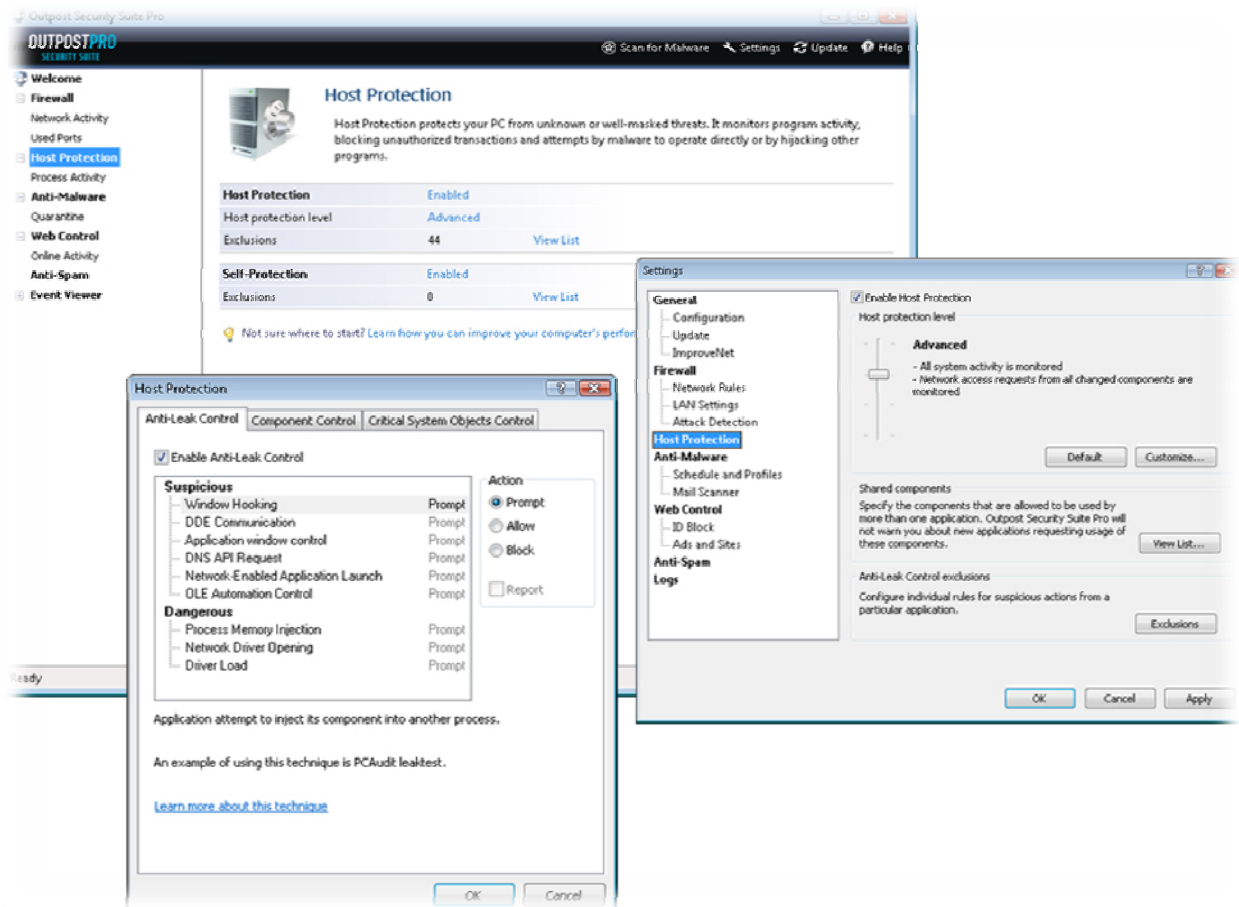
The information security community has developed special tools to test firewalls' ability to recognize unauthorized program interactivity and prevent malware from connecting to the network using legitimate programs' rights. Called "leak tests", these software tools simulate attempts by malware to send data "home", so that users can see how their firewalls might respond to such a threat.

Leak tests use a variety of techniques and mechanisms to test firewalls' ability to prevent unauthorized outbound data transmissions; they are legitimate utilities that send only user-permitted information to isolated test locations and cannot damage the system. For this reason, some people say that leak tests are not real-world situations and so are only proof-of-concept lab examples. However, because the techniques they use can and have been used by actual malware programs, they serve as a valuable indicator of a firewall's preparedness to deal with real-world outbound attacks.

## A look inside Outpost functionality

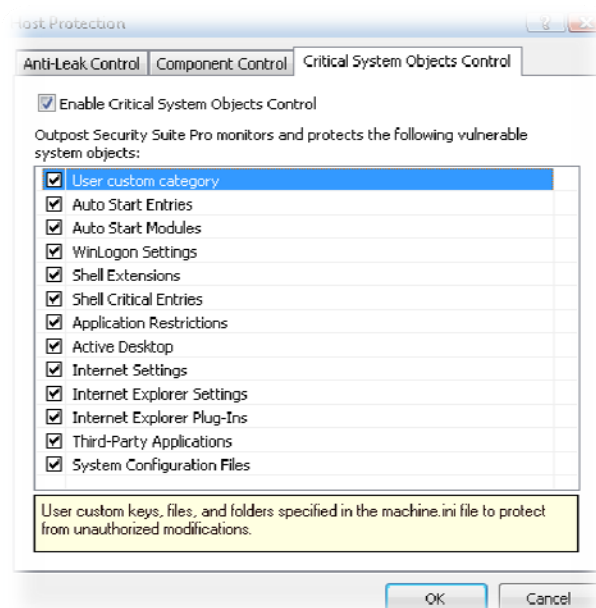
Outpost Pro delivers the ultimate in inbound and outbound connection protection. Its Host Protection module defends against all currently-known data theft techniques.

This module is accessed through the main interface by going to the Host Protection tab. From Outpost's settings menu, you can adjust the level of Host Protection monitoring by moving the sensitivity slider up and down.



The Host Protection module is at the heart of Outpost's proactive protection

In Outpost Pro 2008, a new tab with advanced options has been added to offer additional system protection.



Advanced monitoring properties in the 2008 version of Outpost

## Leak tests overview

Leak tests serve as a common benchmarking tool to determine a firewall's effectiveness in safeguarding personal information against unauthorized disclosure. A wealth of information on leak tests can be found at [Firewall Leak Tester](#) and [Matousec Transparent Security](#). The number of leak tests is growing all the time, as are their proficiency and sophistication in the use of new techniques. Firewall software developers including Agnitum continuously test their products against currently-available leak tests in much the same way that a new car's handling and performance is tested under controlled conditions before it is shipped out to dealerships.

Although passing all current leak tests is not a 100-percent guarantee that your firewall is bulletproof (there are no 100% guarantees in security), it is a good indication that the firewall can withstand determined attempts to steal data.

Before we get into the testing, let's briefly review the ways in which hackers might attempt to steal information from your PC.

## Ways information can leak

Below is an overview of all known techniques used in today's leak tests.

### Leak technique #1: Filename/identity spoofing

How it works:	<p>The technique involves changing the name of an executable file such as <i>eviltrojan1.exe</i> to a common, usually trusted application such as <i>AdobeUpdater.exe</i>, running the file under a fake name and later attempting to send data past the firewall using the trusted application's access credentials.</p> <p>A more advanced method to spoof the identity of a program involves changing its parameters such as image base, Windows title, local placement, and other identifiers to correspond to those belonging to a legitimate process.</p>
Impact:	<i>Light to moderate</i>
Implication:	Firewalls that do not implement proper authentication of network-accessing processes will be defeated by this technique. To block the technique from succeeding, the firewall must identify an application according to its unique signature as opposed to a simple filename association.
What's the risk:	Trojans using this technique can steal personal data and send it to whomever they choose.

### Leak technique #2: Trusted process launch with parameters

How it works:	<p>The technique involves one process launching another process, usually a trusted network-enabled application, with additional startup parameters. These parameters constitute a threat because they change the way the 'child' process behaves. For instance, a command-line script ordering the child process to execute a certain command could be embedded within those parameters and used in an attack.</p> <p>In practice, malware could steal your bank account details by starting a trusted program like <i>iexplore.exe</i> with a preconfigured URL in the address bar containing your credit card authorization code, and send this information over the Internet by exploiting the child process's privileges with the firewall.</p>
Impact:	<i>Light</i>

Implication:	The technique defeats firewalls that do not check a parent process's permission to launch a child process with extra parameters. The technique requires that the firewall alerts the user whenever an unknown process initiates a trusted network-enabled program with modified default startup parameters.
What's the risk:	A specially-crafted browser command could transmit your information to an unauthorized third party.

### Leak technique #3: Exploitation of trusted firewall policies

How it works:	The technique emulates a malicious application attempting to send network data over the trusted ports that are typically opened by the firewall to common Windows applications, components and services. For example, NetBIOS port 139 is generally allowed for File and Printer sharing over a LAN.
Impact:	<i>Light</i>
Implication:	The technique will get past firewalls that do not properly associate existing rules with requesting applications but instead blindly grant access to any requesting executable exhibiting typical network activity.
What's the risk:	If the firewall fails, your personal data will leak over standard communication ports.

### Leak technique #4: DNS request spoofing

How it works:	<p>The technique involves sending a spoofed DNS request that may contain extra data beyond what is normally allowable in a DNS entry. Sensitive user information could be extracted from a PC, inserted into the body of a DNS request, and successfully transferred inside legitimate traffic.</p> <p>A DNS request is a common way for an Internet-requesting program to find the Internet host it wants to connect to. The DNS server translates the hostname, for example <code>www.google.com</code>, to the numerical IP address used in networking, for example <code>209.85.135.99</code>.</p>
Impact:	<i>Moderate</i>
Implication:	Windows provides an interface for common network operations such as DNS, DHCP or NetBIOS activity. It is vital to control which program has permission to initiate DNS requests and whether those requests are legitimate. Firewalls that do not properly control DNS queries will fail to stop this technique.
What's the risk:	Lack of control over outbound connection permission requests could cause data to be transmitted to the outside world using common means such as DNS requests.

### Leak technique #5: Component injection

How it works:	The technique involves embedding a component belonging to one process into the memory space of another process. This component, usually a DLL (Dynamic Link Library), begins to operate in the context of the second process, presenting itself to the firewall as the authentic second process.
Impact:	<i>Moderate</i>
Implication:	Essentially, this technique works by injecting a hostile DLL into the memory of a

	trusted process and then accessing the Internet using the latter's credentials. In order for the firewall to successfully deter the DLL injection attacks, it should be able to monitor and control which components are attached to the main process.
What's the risk:	The injected DLL could be any malicious program that steals user data. Any unchallenged Trojan exploiting this technique would result in a loss of valuable assets.

### Leak technique #6: Interaction through Windows messages

How it works:	Windows provides multiple ways for installed applications to interact. One such way is through Windows messages, a mechanism that enables one application to control the behavior of another application by adding input to the target process's window.
Impact:	<i>Moderate</i>
Implication:	A hostile program can communicate with a trusted application and control its activity through Windows commands. A firewall should be able to detect illegal interaction attempts and alert the user.
What's the risk:	Private data is at risk if the firewall fails to block illegal data transfers resulting from the illegitimate use of Windows messages.

### Leak technique #7: Process injection

How it works:	Similar to component injection, the technique injects an entire process into the memory space of another process. This enables the injected process to become part of the compromised process, acting on its behalf.
Impact:	<i>Severe</i>
Implication:	Due to imperfections in the isolation of a target process, an alien process can append itself to the target process and introduce new, usually hostile, functionality. This event is sometimes referred to as process hijacking. This is a very advanced tampering technique which is hard to stop because it requires that a firewall safeguard the applications' integrity and proactively prevents local intrusion.
What's the risk:	A Trojan with a capability to hijack another process can transfer personal data if the firewall doesn't stop it from taking over another application.

### Leak technique #8: Using DDE and OLE automation to control applications

How it works:	Another, newer, way for applications to communicate is via the COM (Component Object Model) interface, which enables applications to share common data and update that data in real-time. However, this also enables one program to issue commands to the other and control its behavior. Parts of a bigger COM interface are DDE and OLE automation techniques that can be leveraged with malicious intent to control the activity of a trusted program by the intruding application.
Impact:	<i>Severe</i>
Implication:	Although OLE automation and DDE are extensively used for normal application activity, it's important for the firewall to control these techniques to prevent a

	legitimate program from being manipulated by a malicious intruding application.
What's the risk:	Trojans are actively using these mechanisms as additional attack vectors to harm users' data.

### Leak technique #9: Deployment of a new network protocol

How it works:	<p>Windows programs use the standard TCP/IP protocol that comes as part of the operating system to send and receive data. Firewalls should filter data exchanged through this protocol.</p> <p>In an attempt to bypass firewall's filters, a malicious program might inject a proprietary driver that adds another communication protocol to the system and then route malicious traffic through this new protocol.</p>
Impact:	<i>Severe</i>
Implication:	A firewall must guard against the unauthorized inclusion of a new protocol driver and alert the user whenever data is being communicated in unexpected ways that might indicate Trojan activity.
What's the risk:	Personal data can leak over a new connection channel if that channel is not monitored by the firewall.

### Leak technique #10: Process ID obfuscation

How it works:	Windows assigns each running process a special identifying number, the PID or Process Identifier. Firewalls use this number, along with other variables, to identify running applications and associate them with existing access policies in the current session. Malware can attempt to hide its presence by constantly changing PIDs by opening, closing and reopening its payload. An ineffective firewall may lose track of the original process after it has changed its PIDs several hundred times in rapid succession and as a result let malicious traffic through because it no longer recognizes the running processes.
Impact:	<i>Severe</i>
Implication:	The firewall must register and track changing PIDs for any active program, no matter how many times they change. Alternatively, it can maintain process recognition based on its own technologies and not rely on data supplied by the OS.
What's the risk:	Using this technique, a Trojan can steal data and transmit it to hackers while remaining invisible to an ineffective firewall.

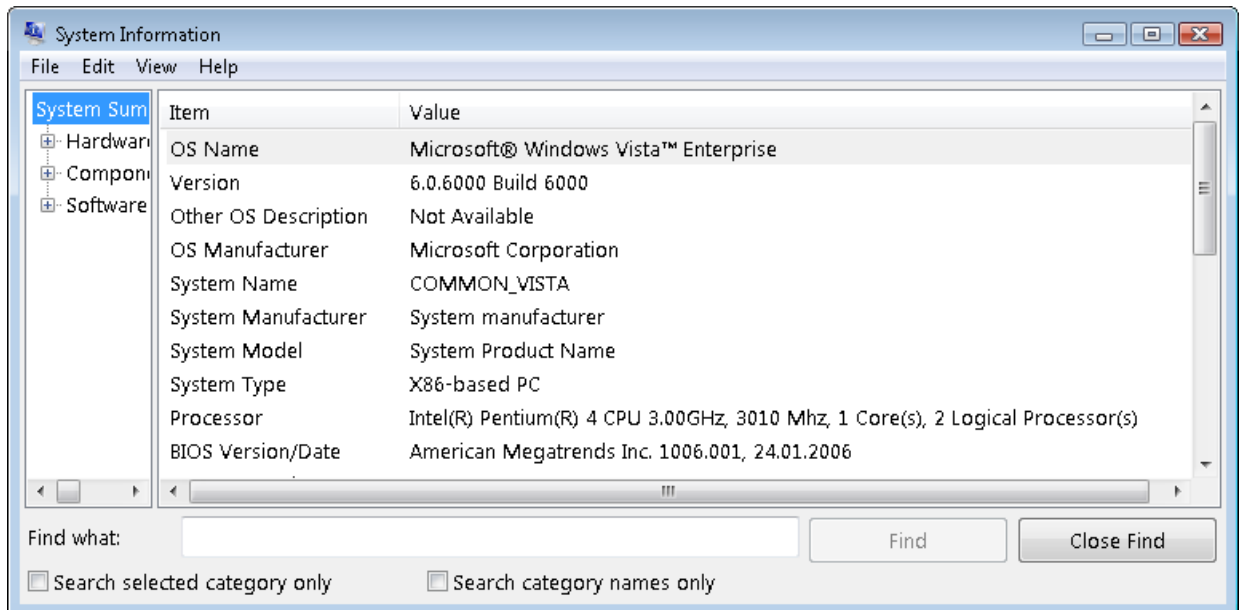
### Leak technique #11: Removing interception functions

How it works:	The majority of security software programs use special "hooks" to intercept the interactive operations of an application pending user approval of the process. These hooks can be dismantled by malware to prevent firewalls from protecting against illegal program activity. The process of removing these protective hooks is called unhooking, a technique which now represents one of the most advanced techniques to interfere with the normal functioning of security programs.
Impact:	<i>Severe</i>

Implication:	If the hooks can be undone, the firewall protection is effectively no longer operational.
What's the risk:	By employing advanced unhooking techniques, Trojans can disarm protection and act freely on the user's system to do whatever harm they wish.

## The leak tests

Currently, there are twenty-four separate leak test programs and Outpost has been tested against all of them. Our test system has the following default configuration: Windows Vista 32-bit Enterprise Edition with all current security updates installed, as shown below.



### Test system configuration properties

Now we're going to put each leak test under the microscope, one at a time, and describe how Outpost Pro software reacts to them. The leak tests are arranged in alphabetical order, and a short commentary accompanies the results of each test.

## Methodology

First, we attempt to run each leak test under a non-Administrator account. If the credentials are insufficient, we manually elevate the test's rights to Administrator level with the "Run as Administrator" command. If the test cannot be started in this environment, we change the testing platform to Windows XP and test it there. Each leak test is first run on an unprotected computer to ensure the test itself is operational.

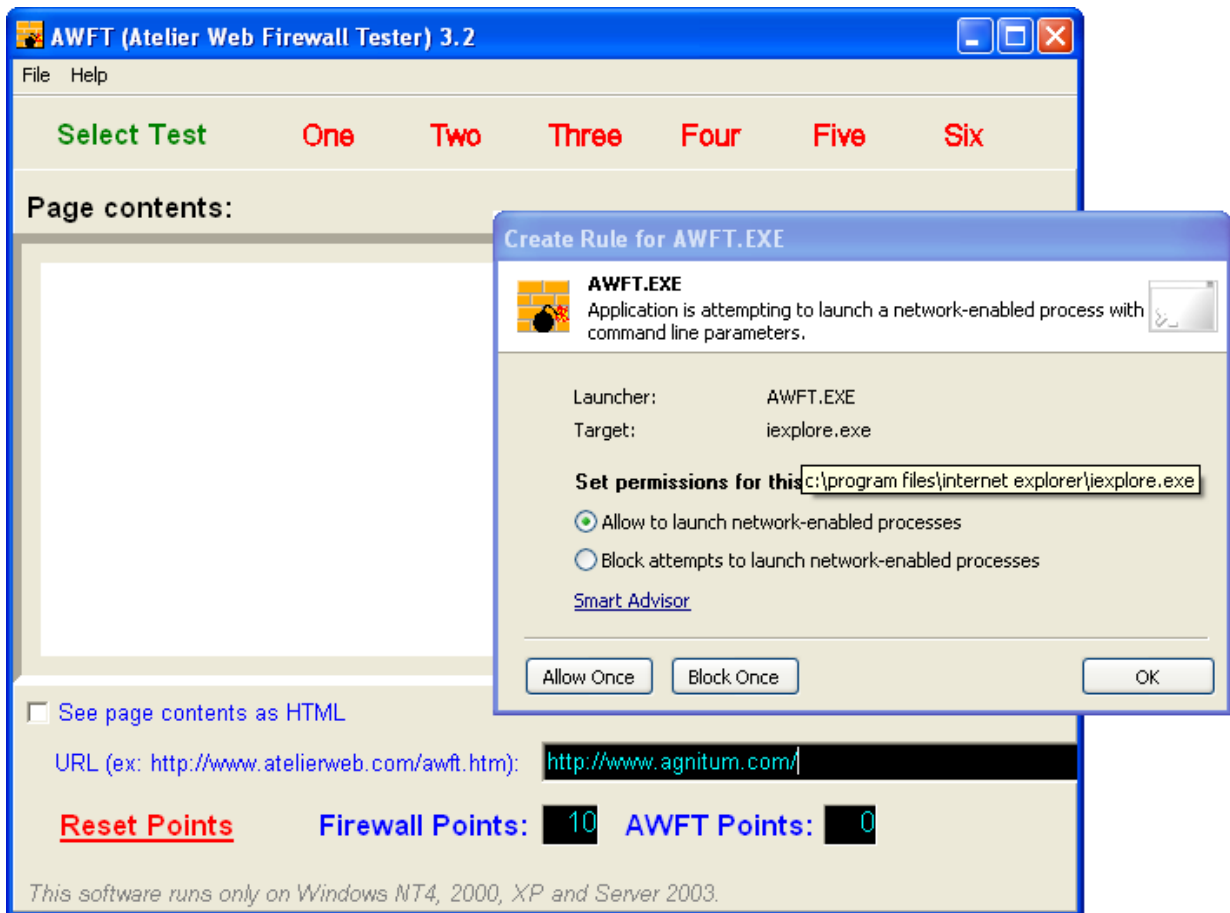
## 1. Atelier Web Firewall Tester (AWFT)

Homepage: <http://www.atelierweb.com/awft>

Direct download link: [Test1](#)

**Penetration techniques used:** Component injection; Process injection; Trusted process launch with parameters.

The AWFT test comprises a suite of six tests, combining multiple techniques. The test assigns a maximum score of 10 points to a totally impenetrable firewall. Outpost scores a perfect 10 by defeating every technique:



### **Implication:**

Outpost prevents a variety of penetration techniques, including those used in the AWFT leak test, to deliver truly proactive protection against unknown threats.



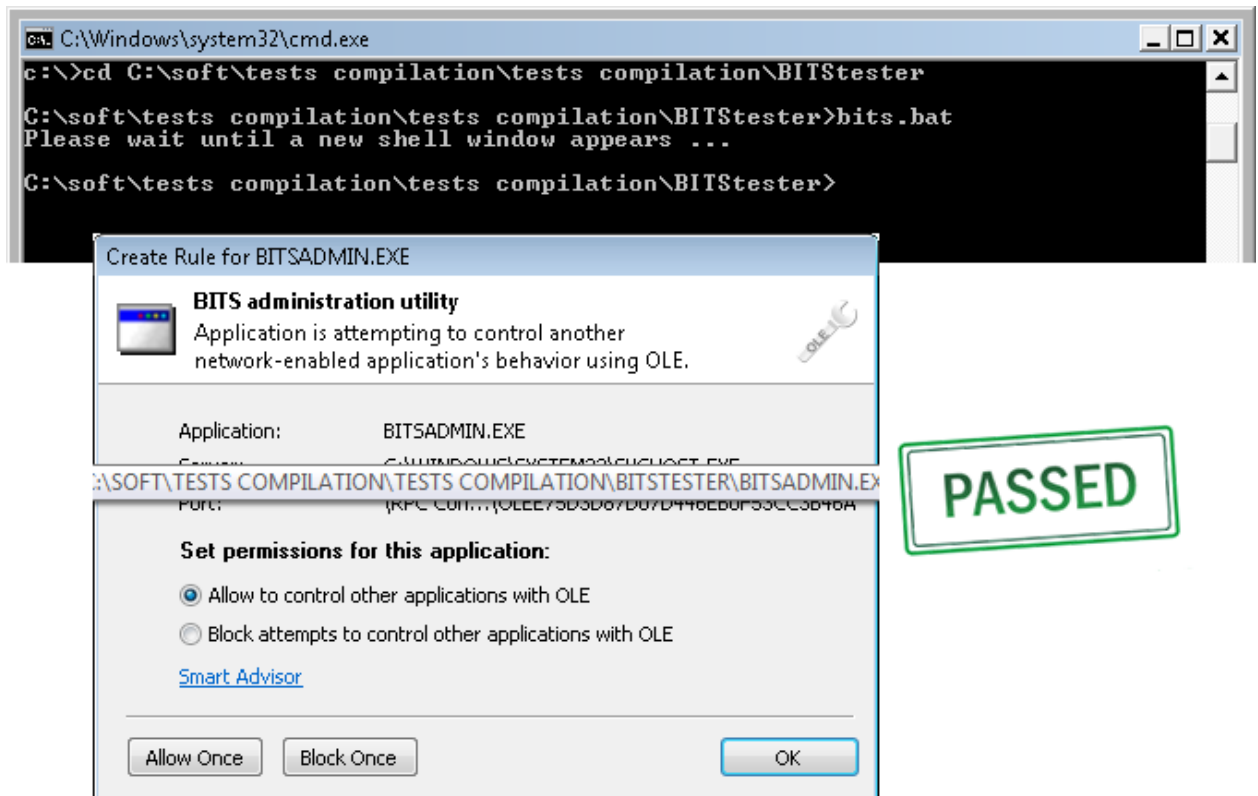
## 2. BITStester

**Homepage:** <none>

**Direct download link:** [Test1](#)

**Penetration techniques used:** Exploitation of trusted firewall policies; use of DDE and OLE automation to control applications.

BITStester starts the Windows Background Intelligent Transfer Service (BITS) and uses an administrative utility – BITSadmin.exe - to control the service's activity through OLE automation. It then directs the service to download a file from an Internet site. BITStester automates the procedure with a batch script. Here's how Outpost responds:



### **Implication:**

Not only does Outpost prevent communication with a trusted program through OLE, it also ensures the existing firewall rules are not misused by malware exhibiting normal network activity.

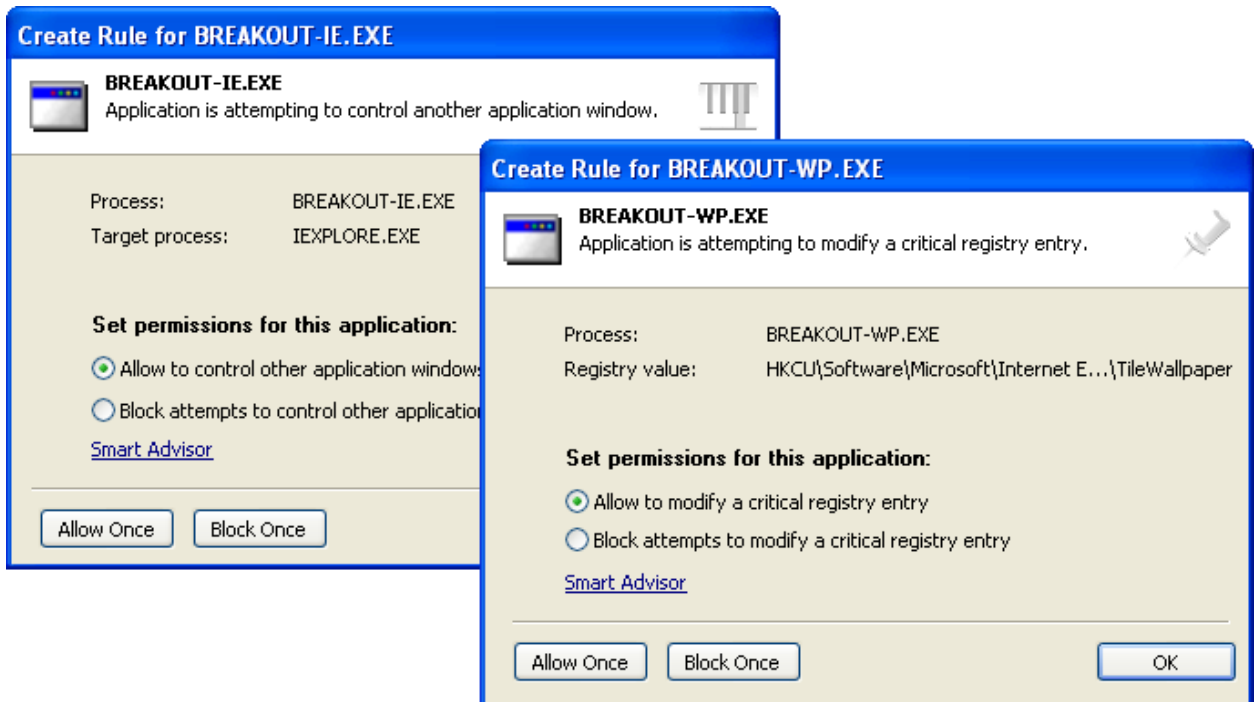
### 3. Breakout

Homepage: <http://www.dingens.org>

Direct download link: [Test1](#) (Internet Explorer); [Test1](#) (Firefox); [Test2](#) (Windows Wallpaper)

Penetration techniques used: Interaction through Windows messages.

The Breakout program consists of three tests; the first two are similar but use different browsers, while the third creates an HTML page and attempts to set it as interactive desktop wallpaper. Test 1 uses Windows messages to make the browser access a predefined site; Test 2 involves modifying the registry to enable the wallpaper change. We can see that Outpost detects this trick by displaying the following prompts:



**Implication:**

Neither unauthorized registry modification nor unauthorized program interaction is possible with Outpost Pro 2008.

**PASSED**

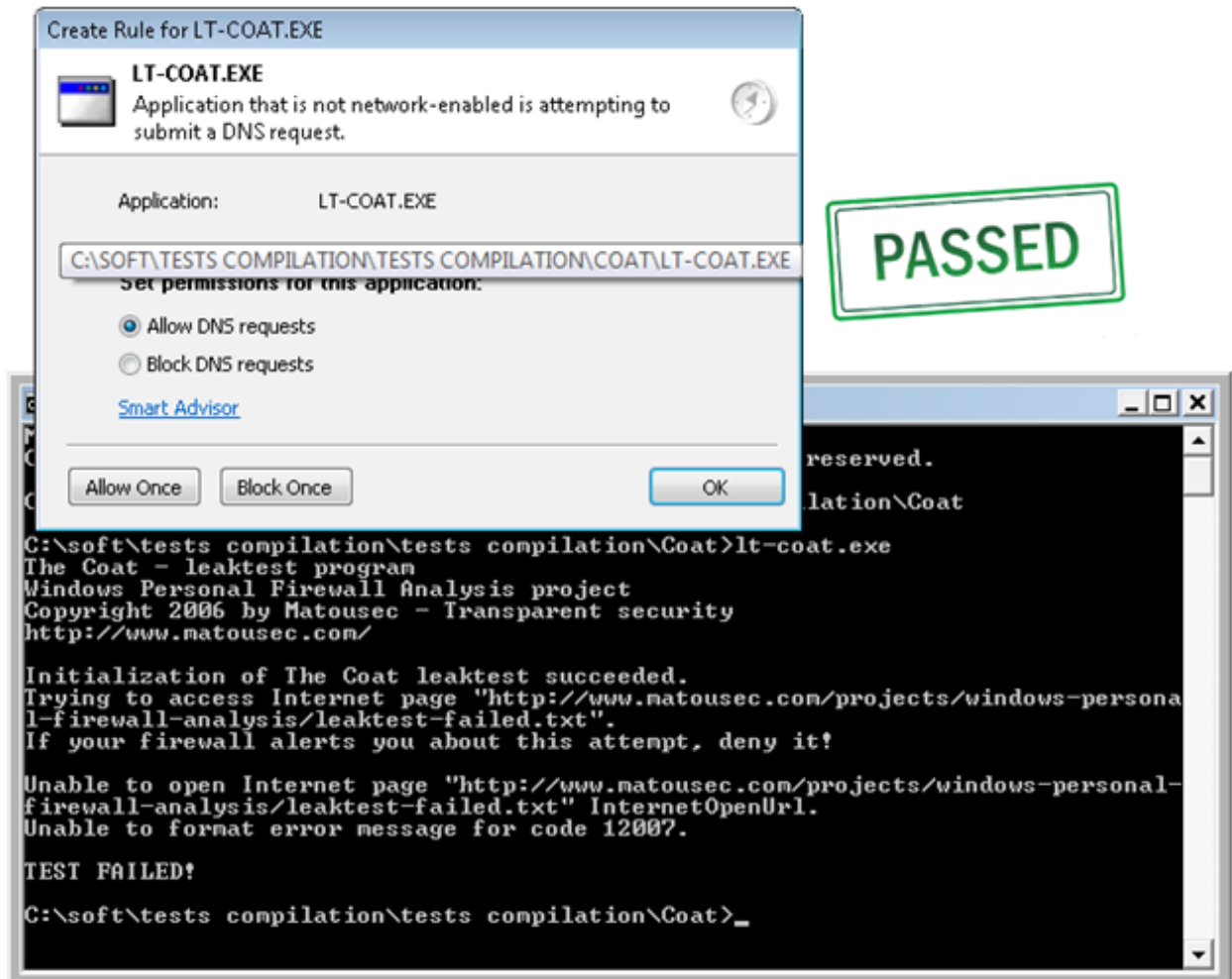
## 4. Coat

Homepage: <http://www.matousec.com>

Direct download link: [Test1](#)

Penetration techniques used: Filename/identity spoofing.

Coat is a very complex and demanding test, created by renowned security research group Matousec Transparent Security. Coat acts by patching its own memory in order to impersonate a trusted application and then carry out an attack by submitting a network query. Here's how Outpost reacts:



**PASSED**

### **Implication:**

Some firewalls have a serious flaw because they rely on Ring 3 data which can be tampered with by malicious processes. Outpost maintains its own internal registry of running processes rather than relying on the vulnerable and sometimes misleading information supplied by the OS.

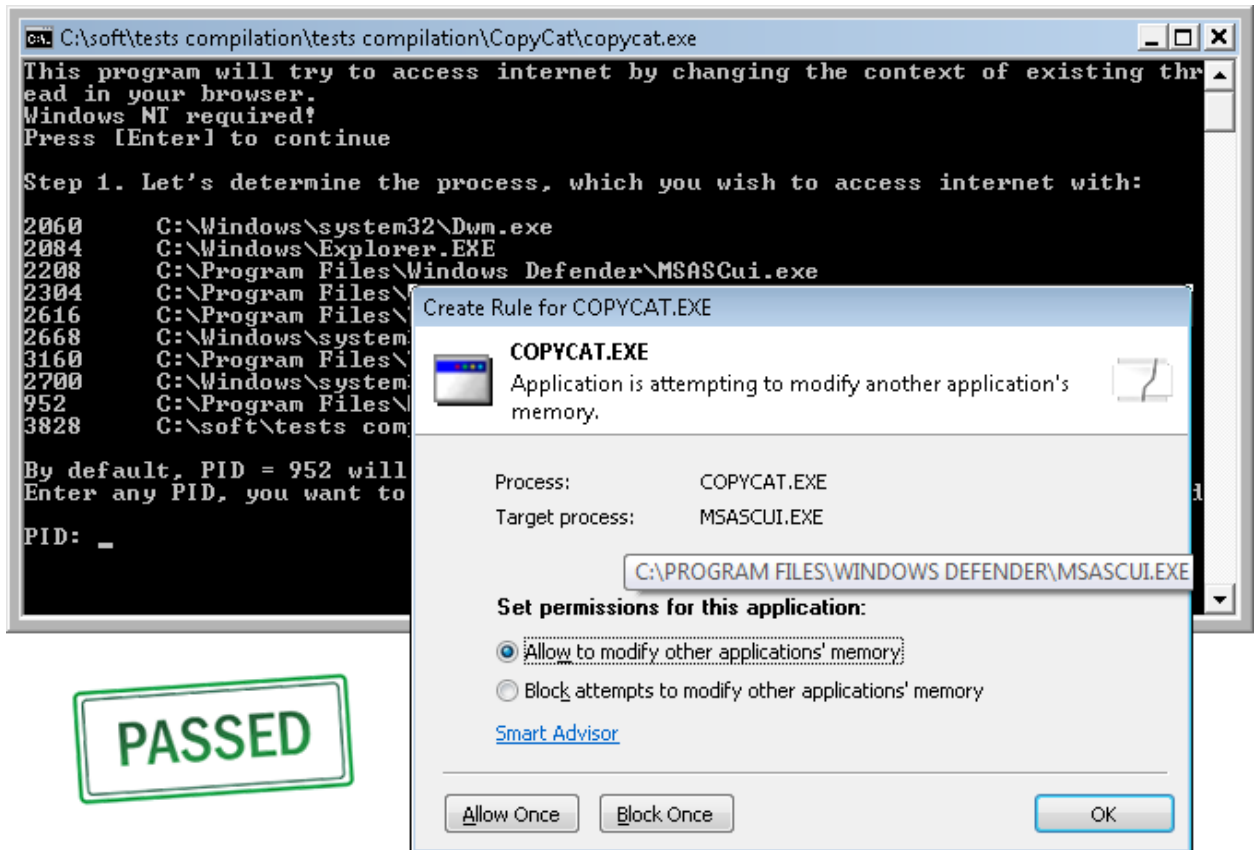
## 5. Copycat

Homepage: <none>

Direct download link: [Test1](#)

Penetration techniques used: Component injection.

Copycat is a classic test that injects a foreign DLL file into the memory of any running process and then directs the compromised process to access a special URL.



### Implication:

After the test completes successfully on an unprotected machine, a \*.txt file appears on disk C detailing intrusion. This does not happen on an Outpost-protected machine, because Outpost monitors memory integrity for unlawful injection.

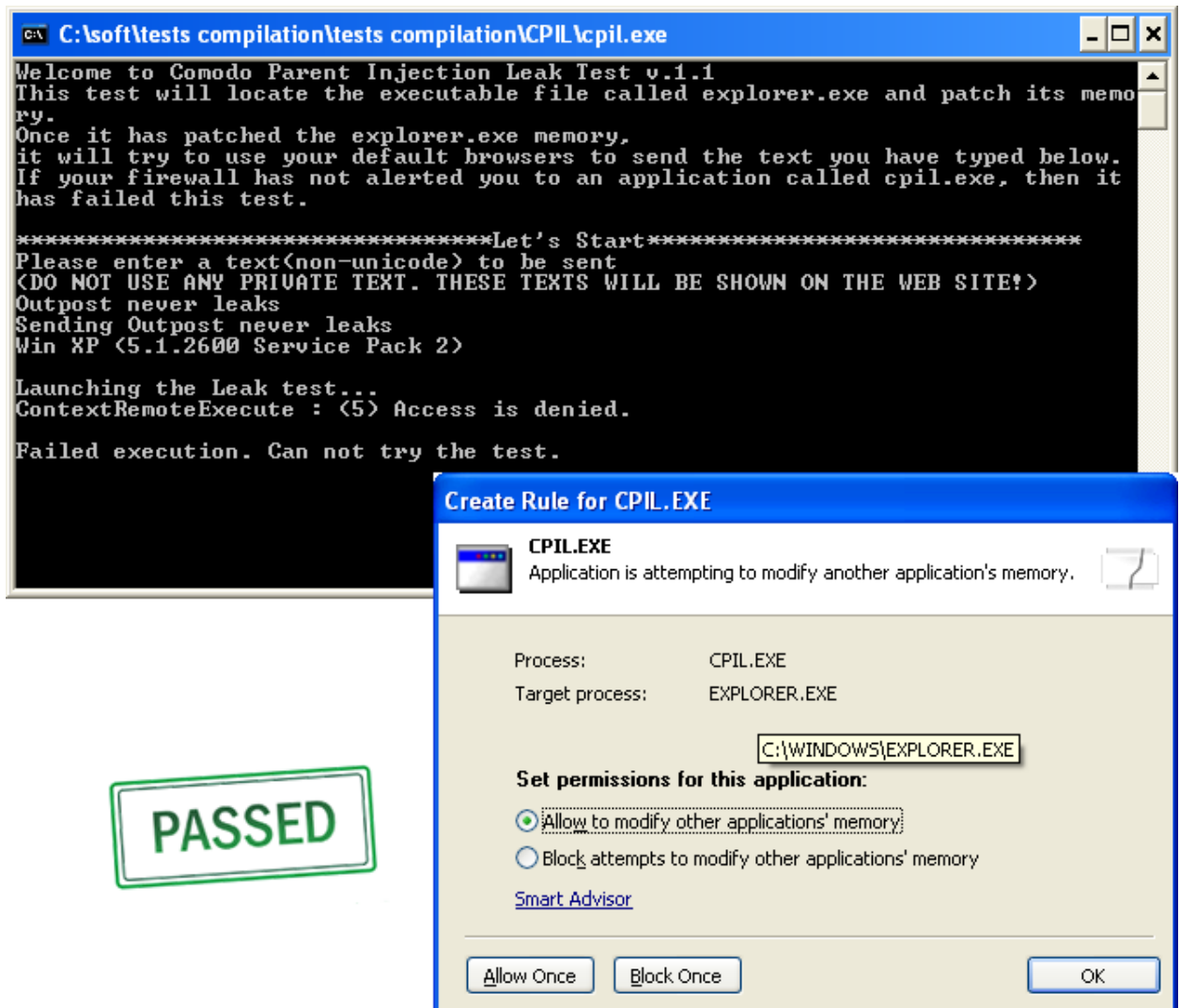
## 6. CPIL v1 (Comodo Parent Injection Leak Test)

**Homepage:** <discontinued, migrated into CPIL Suite (see next page)>

**Direct download link:** [Test1](#)

**Penetration technique used:** Component injection.

Another classic injection test, CPIL injects a DLL file into Windows Explorer, modifies its memory, and attempts to connect to the Internet using the compromised program's permissions to establish outbound connections.



**Implication:**

Outpost prevents tampering with a trusted program's memory.

## 7. CPIL Suite

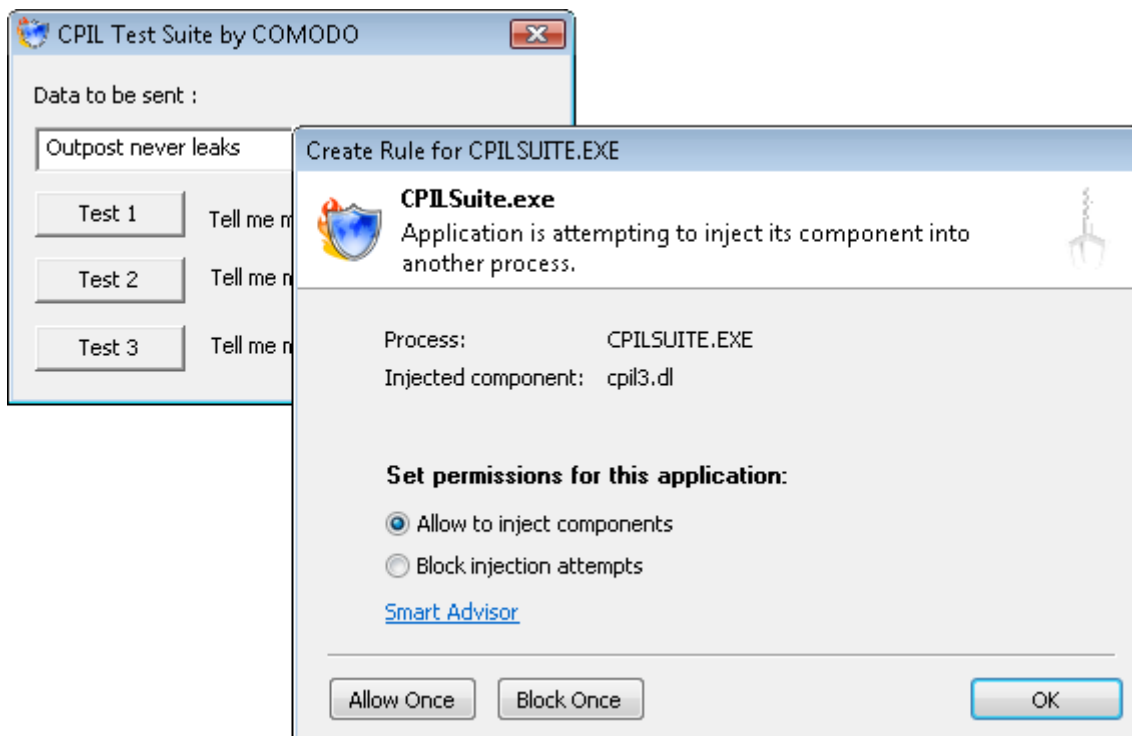
Homepage: <http://personalfirewall.comodo.com/cpiltest.html>

Direct download link: [Test1](#)

**Penetration technique used:** Component injection; Trusted process launch with parameters; Using DDE and OLE automation to control applications; Removing interception functions.

The CPIL suite comprises three separate tests designed to check a firewall's outbound protection.

The first test is the most crucial, as it modifies kernel memory and disables firewall hooks before launching Internet Explorer with modified parameters. The second test patches explore.exe in memory by loading a DLL that attempts to send data through IE by launching it with command line parameters. The final test patches Windows Explorer and through DDE communication instructs the web browser to access a test location.



**PASSED**

### ***Implication:***

CPIL Suite is a collection of advanced techniques that have been used in the real world to steal passwords and other valuable information. By controlling operations of the test samples, Outpost ensures these mechanisms cannot be used by actual malware and so passes all three tests.

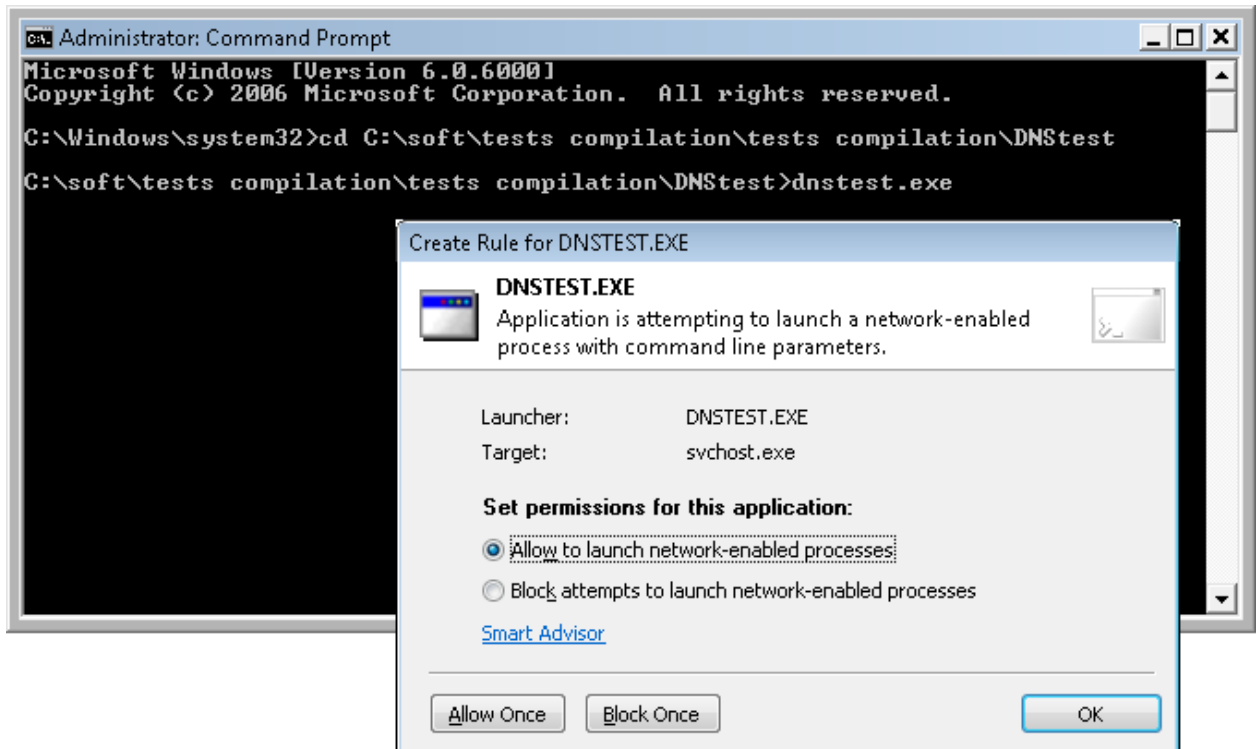
## 8. DNStest

Homepage: <http://www.klake.org/~jt/dnshell>

Direct download link: [Test1](#)

**Penetration technique used:** Component injection; Trusted process launch with parameters.

DNStest attempts to infect the *svchost.exe* process (Generic Host Process for Win32 Services) that accommodates various Windows services, including the DNS client. Once the DNS client has been accessed by a foreign DDL, it is directed to connect to an Internet site. The test is designed to defeat firewalls that don't control injections.



### ***Implication:***

Outpost protects against injection attempts, and also controls whether the trusted application that has been modified is allowed to access the Internet.

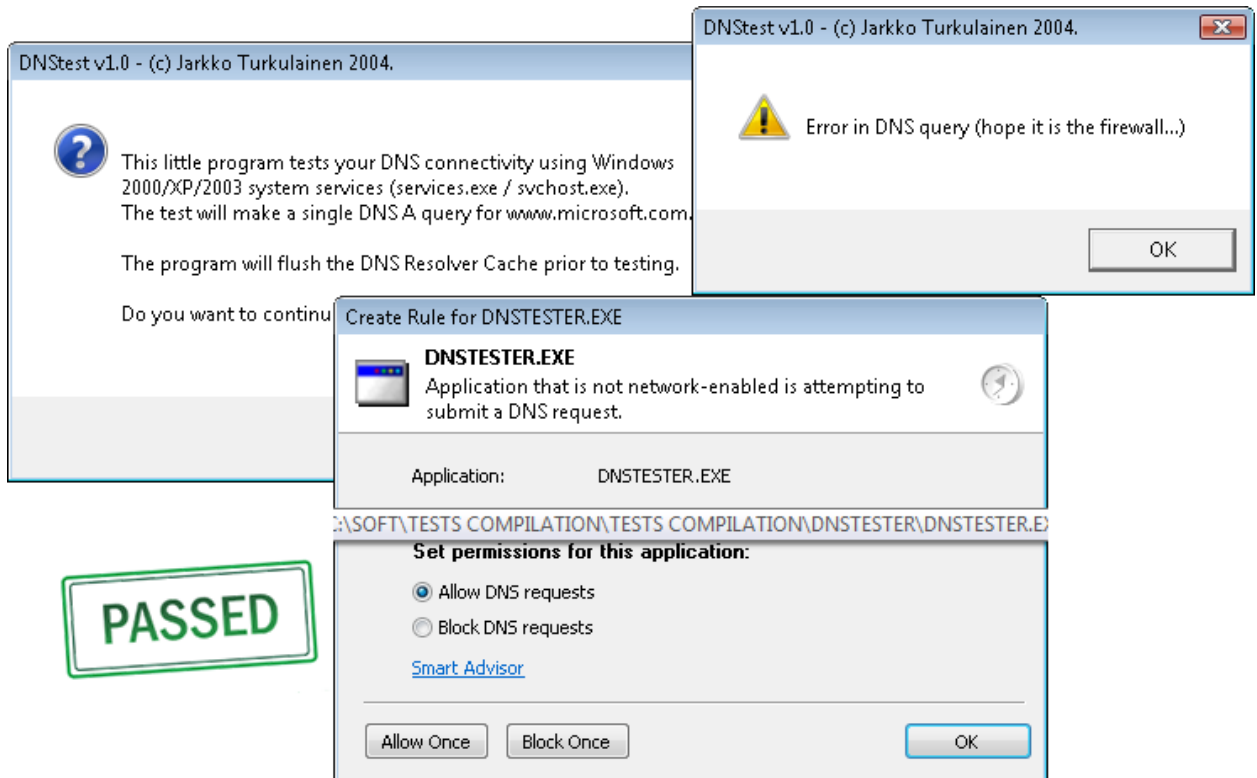
## 9. DNStester

Homepage: <http://www.klake.org/~jt/dnshell>

Direct download link: [Test1](#)

Penetration technique used: DNS request spoofing.

The name similarity with test #8 is not coincidental; both tests were developed by the same security researcher. This test attempts to submit a DNS request; Outpost is ready to tackle this potential breach by reacting like this:



### **Implication:**

DNS queries are perfectly normal for legitimate Internet-connecting applications. However, once a DNS query is initiated with nefarious purposes or which is simply unnecessary, the firewall should restrict such queries.

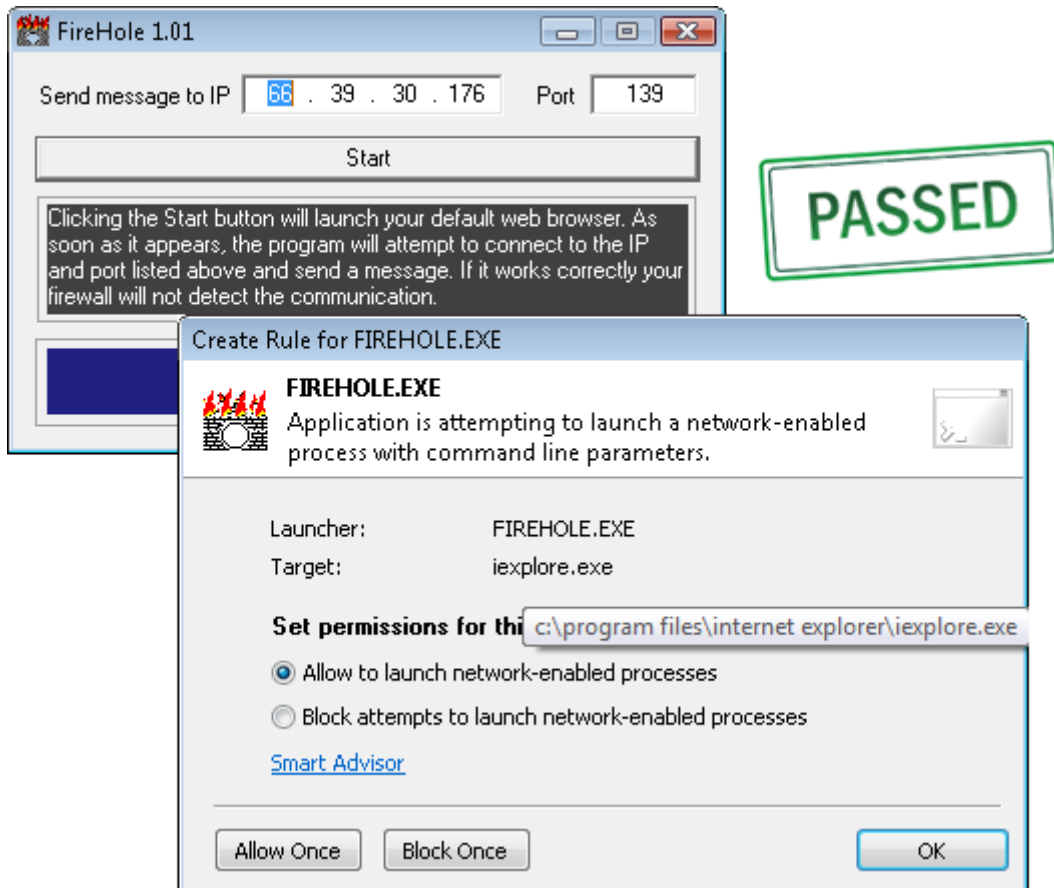
## 10. FireHole

Homepage: <http://keir.net/firehole.html>

Direct download link: [Test1](#)

**Penetration technique used:** Trusted process launch with parameters; Component injection.

FireHole injects its component, fire.dll, into the memory of Internet Explorer. Using the infected browser, the program attempts to access a user-specified Internet host via an arbitrary port. Here's how Outpost responds:



***Implication:***

Outpost monitors the integrity of Internet-enabled programs by restricting unlawful interaction with a foreign component and does not permit unknown programs to run a trusted program with modified parameters.



## 12. Firewall Leakage Tester

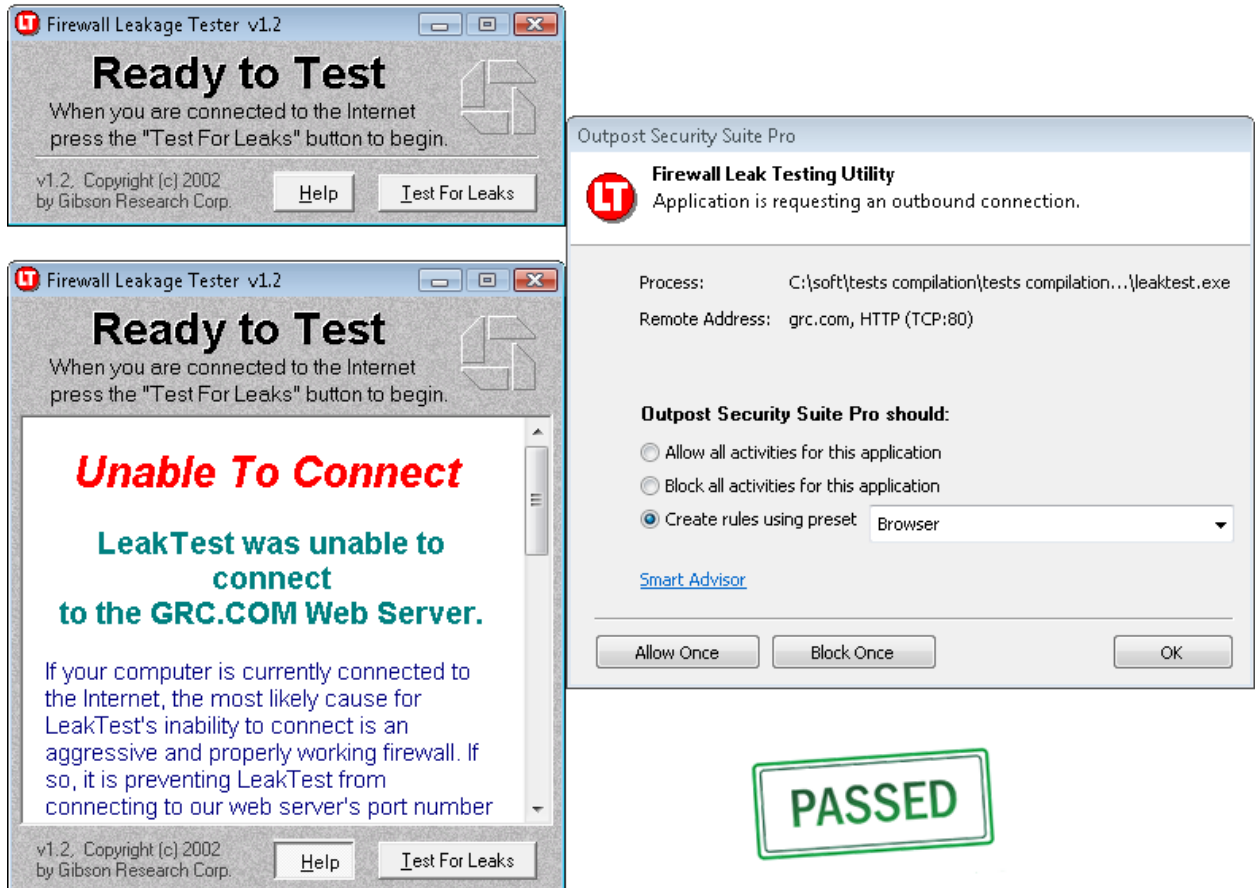
Homepage: [http://www.grc.com/lt/leak\\_test.htm](http://www.grc.com/lt/leak_test.htm)

Direct download link: [http://www.firewallleak\\_tester.com/leaks/leak\\_test1.2.exe](http://www.firewallleak_tester.com/leaks/leak_test1.2.exe)

Penetration technique used: Filename/identity spoofing.

This test renames itself to iexplore.exe and, under this name, attempts to transmit data off the PC.

Outpost stops this attempt and displays an alert:



### **Implication:**

Outpost looks beyond the program's name to its unique identifiers or fingerprints. It uses strong authentication based on vendors' digital certificates, SHA256, MD5 and other robust algorithms that eliminate spoofing.

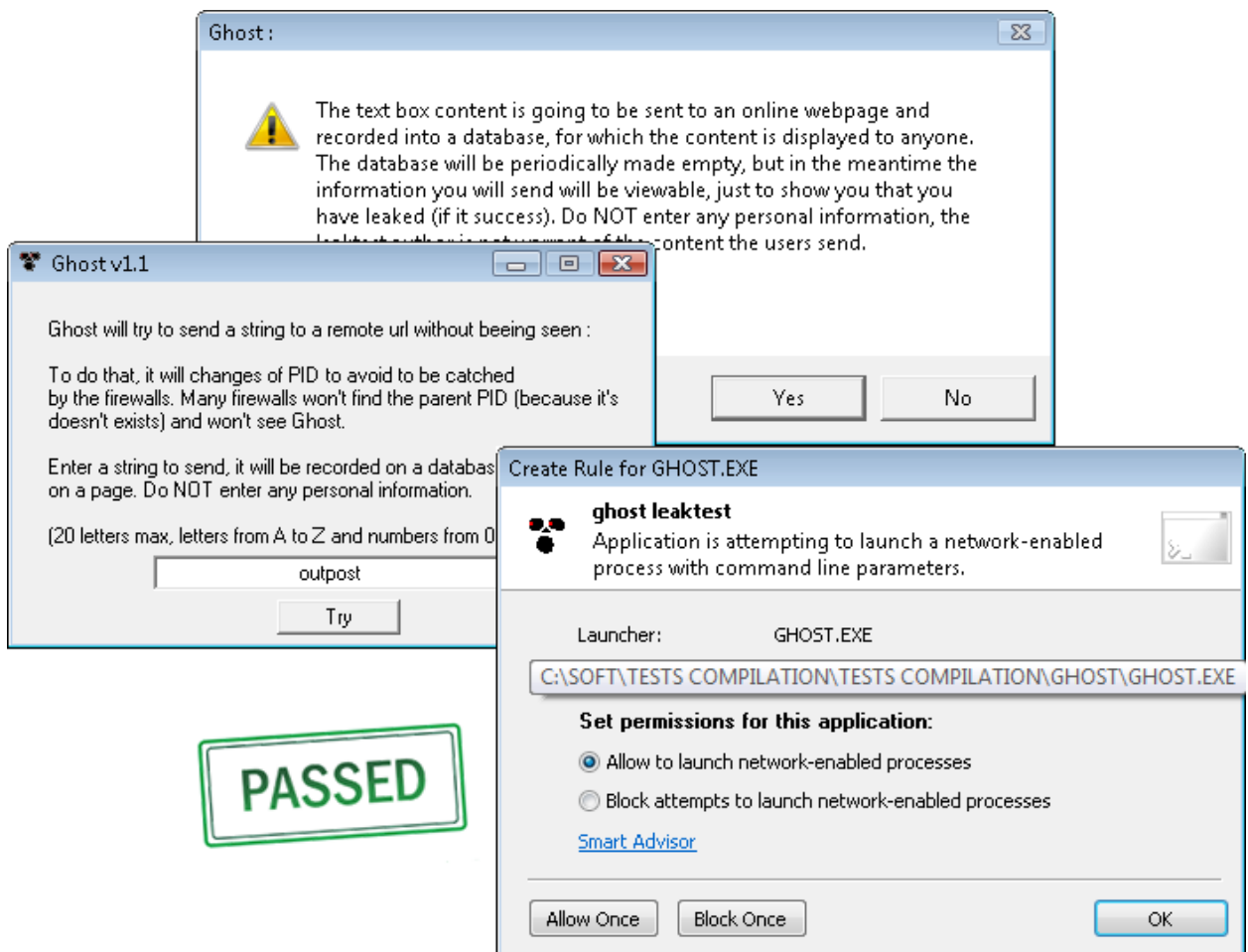
### 13. Ghost

Homepage: <http://www.firewallleaktester.com>

Direct download link: [Test1](#)

**Penetration technique used:** Process ID obfuscation; Trusted process launch with parameters.

Ghost attempts to obscure its presence on a system by constantly changing its PID. This is done by starting, closing and restarting its processes. Many firewalls are unable to withstand such a barrage of constantly changing identifiers for a certain process and simply let the processes proceed unchecked. Here's what a user would see in an Outpost situation:



**Implication:**

Outpost tracks active processes and promptly verifies their permissions for outbound access. By watching program activity and registering running processes in real time, it cannot be tricked by these constant PID changes.

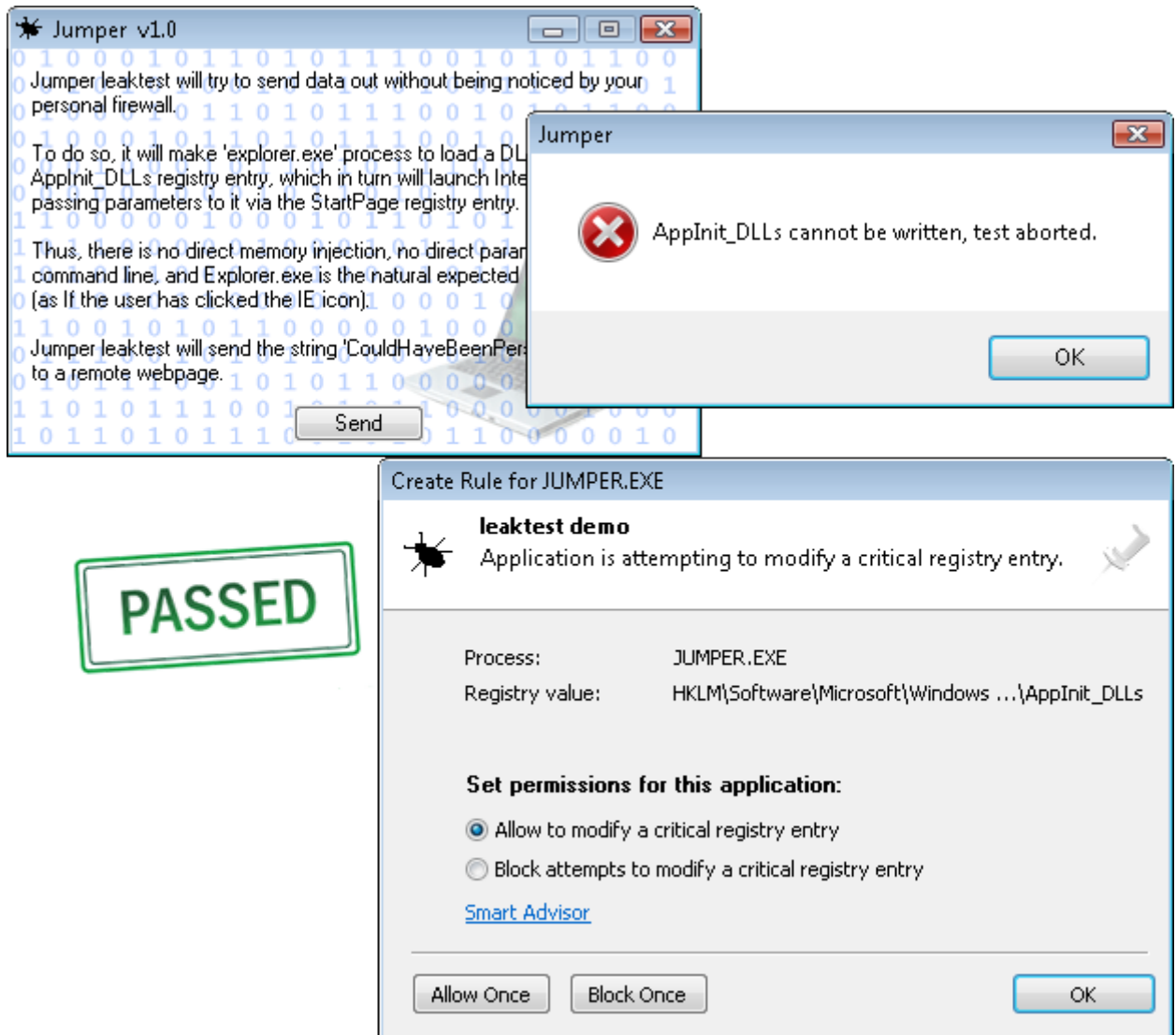
## 14. Jumper

Homepage: <http://www.firewallleaktester.com>

Direct download link: [Test1](#)

**Penetration technique used:** Component injection.

Jumper, along with a number of other leak tests, was created by the well-known proactive security researcher Guillaume Kaddouch (gkweb). The test works by closing a currently-running instance of Windows Explorer (explorer.exe), and modifying its startup parameter in the registry to load a certain DLL the next time it starts. It then runs the modified Explorer with an embedded foreign DLL which directs it to access a testing website. Outpost alerts to this attempt with the following notification:



### **Implication:**

By allowing modifications of the registry to be made only by legitimate applications, Outpost prevents malicious attempts to modify the way a normal application is started.

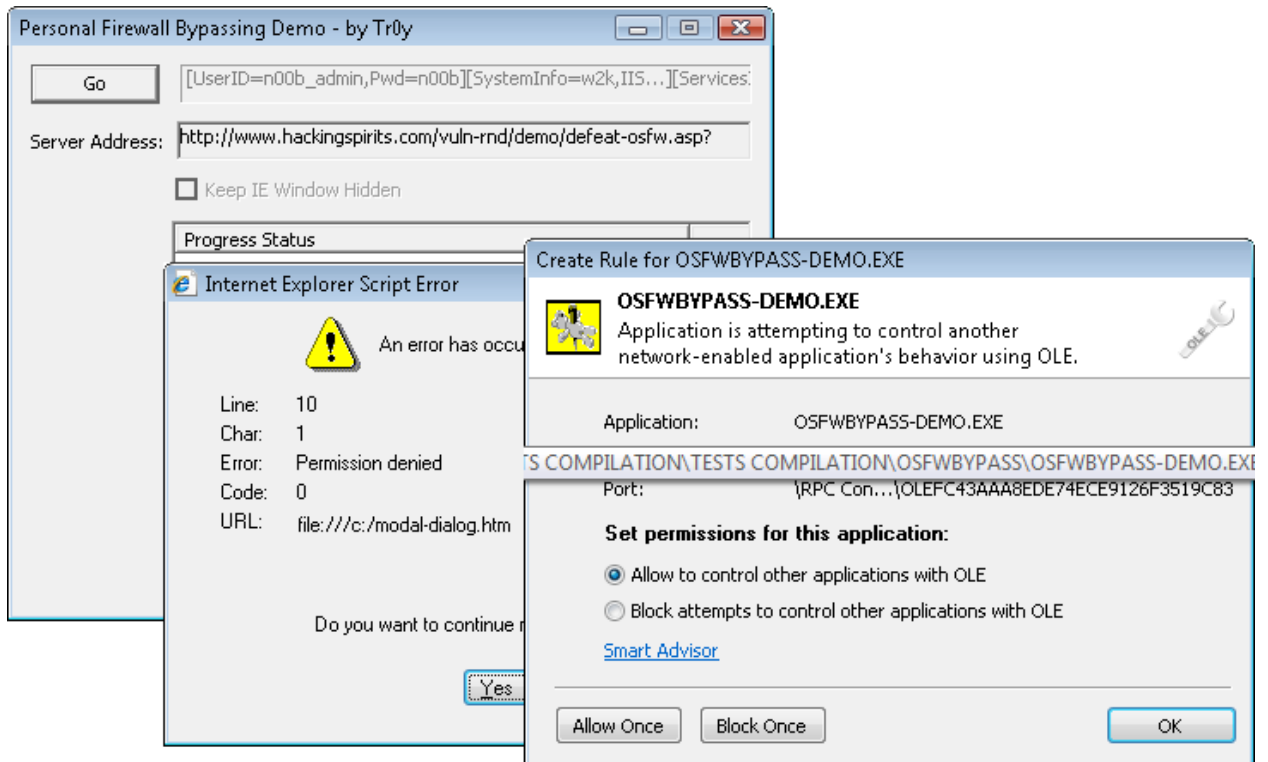
## 15. OSfwbypass

Homepage: <http://www.hackingspirits.com>

Direct download link: [Test1](#)

Penetration technique used: OLE automation.

Using OLE automation, OSfwbypass tries to load an HTML page containing JavaScript code into Internet Explorer. This code then redirects Internet Explorer to access a special web page. Outpost will alert to this activity by bringing up the following window:



**PASSED**

### **Implication:**

OLE automation is a very effective way for applications to interact, but it must be monitored by a proactive security tool to ensure it is being used for benign purposes only.

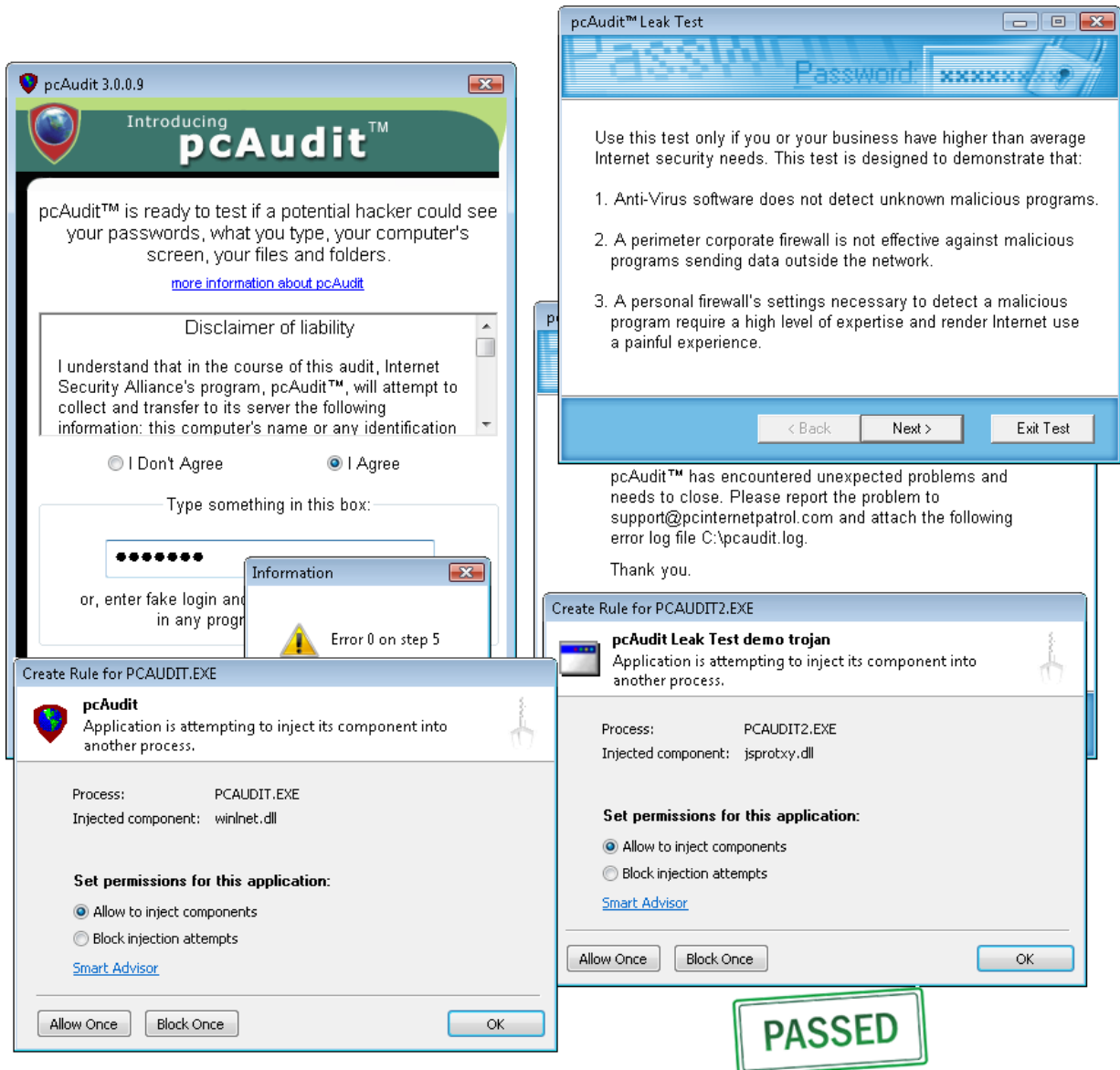
## 16. PCAudit v1, PCAudit v2

Homepage: <http://www.pcindernetpatrol.com/pcaudit>

Direct download link: [Test1](#); [Test2](#)

Penetration technique used: DLL injection.

Both the first and second versions of the PCAudit leak test inject a DLL file into the memory of a trusted application (Windows Explorer) and direct it to access a specified web server. Although the test doesn't work correctly on Vista or XP, Outpost still detected it and proactively stopped unauthorized operations:



### **Implication:**

PCAudit was one of the first leak tests this century but, despite its age, the techniques PCAudit uses still need to be proactively controlled.

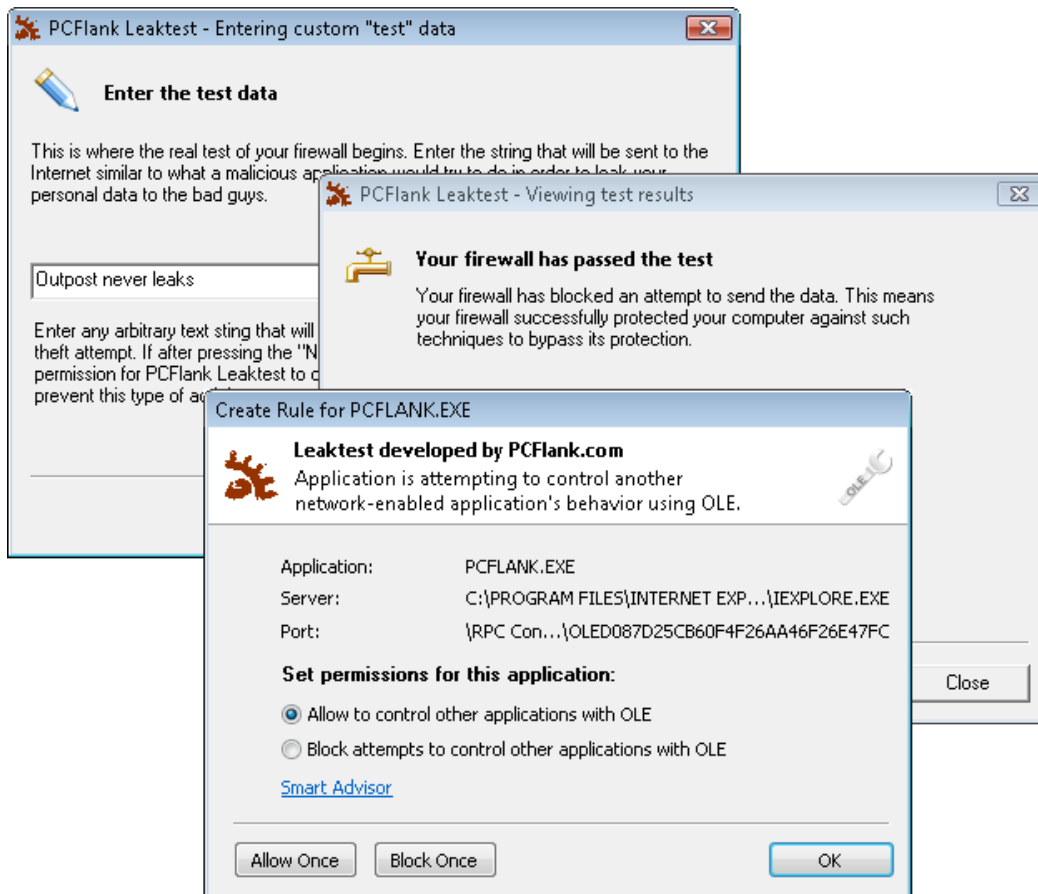
## 17. PCFlank Leaktest

Homepage: <http://www.pcflank.com/pcflankleaktest.htm>

Direct download link: [Test1](#)

Penetration technique used: OLE automation.

The PCFlank leak test accesses Internet Explorer through OLE automation and directs it to send user input to PCFlank's web server. If the test is passed successfully, no user entries will be shown on the page that opens after the test completes. Conversely, a page containing user input will be shown if the firewall fails the test.



### Implication:

The PCFlank leak test pioneered the use of OLE automation as a test for communications with a trusted application. Outpost has been able to detect this technique since before the test first appeared.

**PASSED**

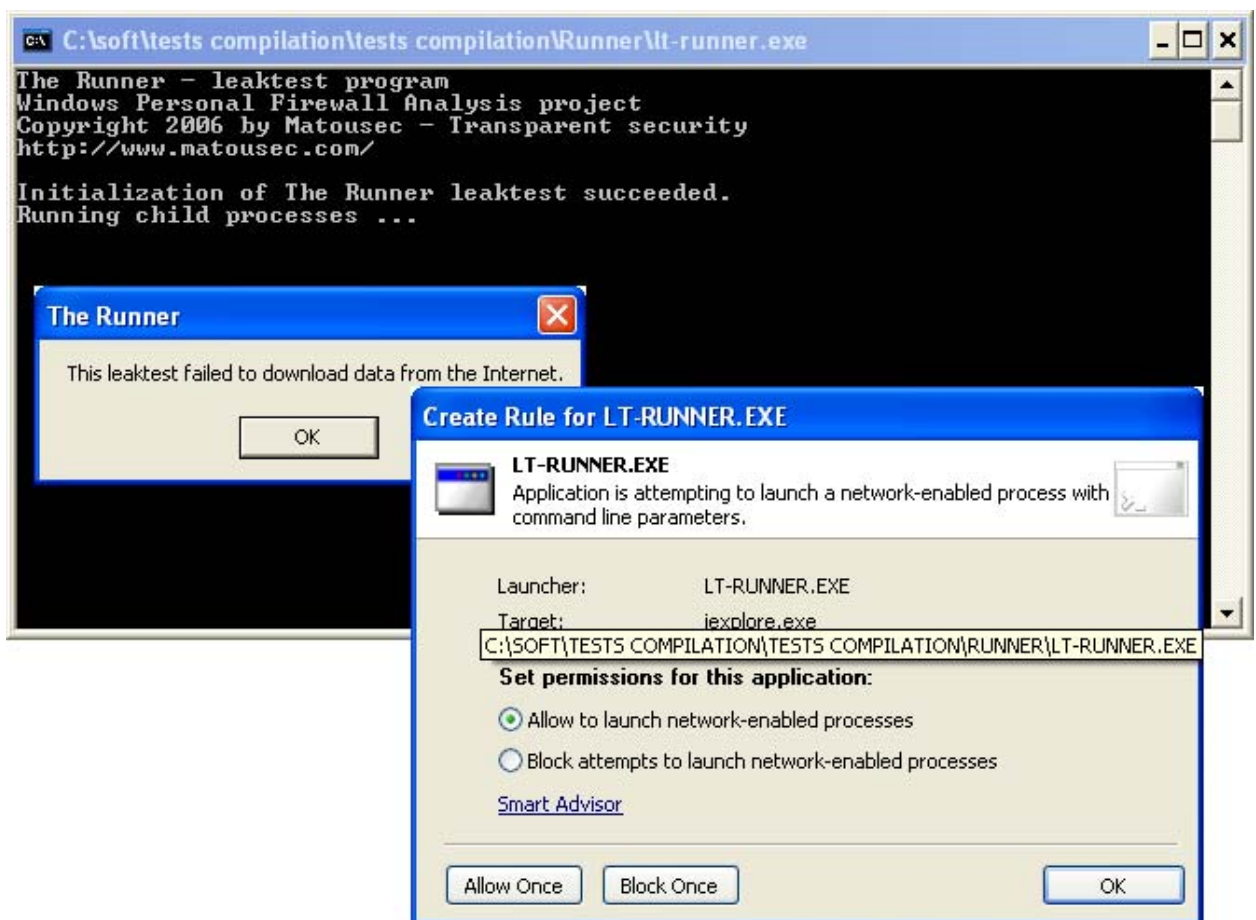
## 18. Runner

Homepage: <http://www.matousec.com>

Direct download link: [Test1](#)

**Penetration technique used:** Filename/identity spoofing.

The Runner leak test locates an executable file belonging to the default web browser, gives it a different name, copies itself to the location of the default web browser and assumes its name. Then it runs itself under a forged name, renames itself to a different name, restores the browser back to its original location, and runs a copy of itself to see if the firewall is able to spot all these “fuzzing” activities. Outpost can detect which file is legitimate, and brings up the following screen alerting a user to suspicious behavior:



### **Implication:**

Although Runner is a very advanced and demanding test, Outpost watches when a program tries to impersonate a legitimate application and blocks this action unless the user authorizes it.



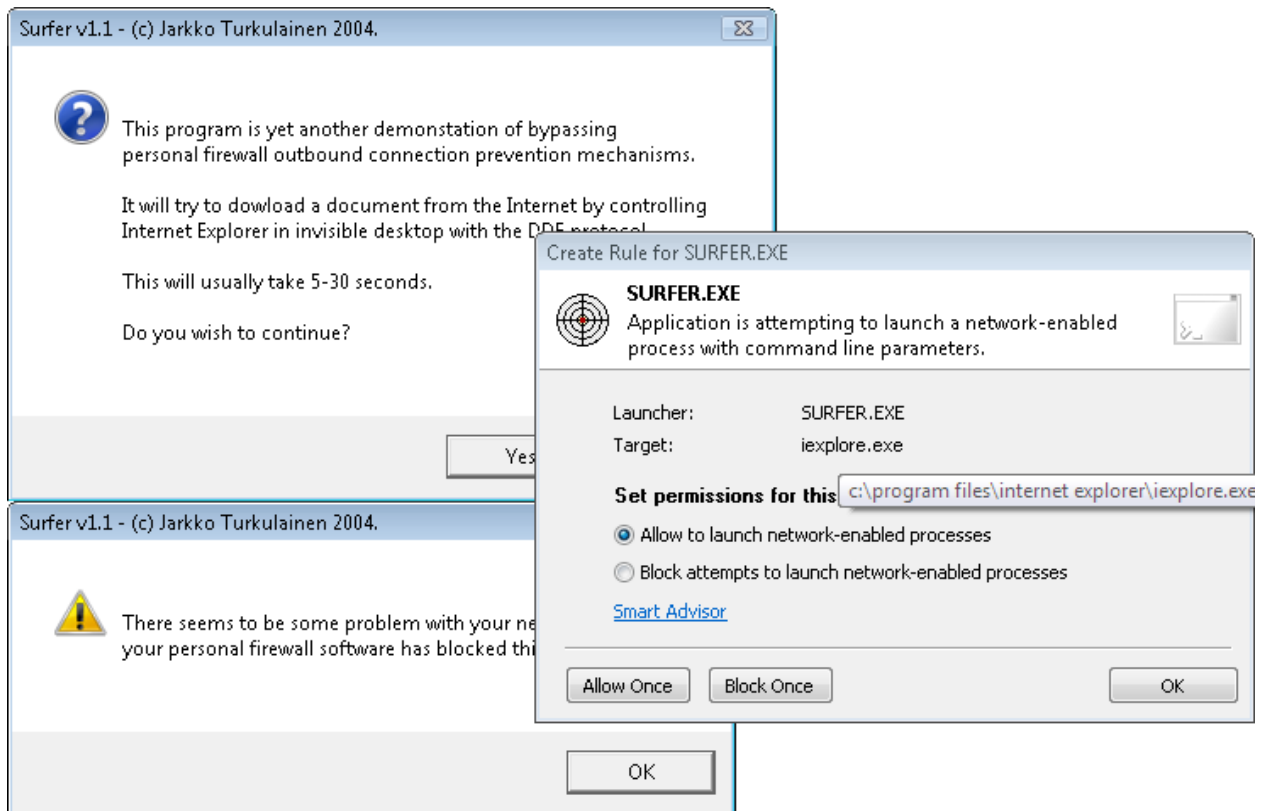
## 19. Surfer

**Homepage:** <none>

**Direct download link:** [Test1](#)

**Penetration technique used:** Using DDE and OLE automation to control applications; Trusted process launch with parameters

Surfer creates a hidden desktop and runs Internet Explorer on it. Then, with the use of DDE, it controls the behavior of the desktop and transfers data to a remote location. Outpost alerts to this activity with the following prompt:



**PASSED**

**Implication:**

DDE is widely used by legitimate applications to interact with each other, but its use for questionable purposes should be detected by the firewall. Outpost permits Internet-active applications to interact with each other only if this activity is authorized by the user.

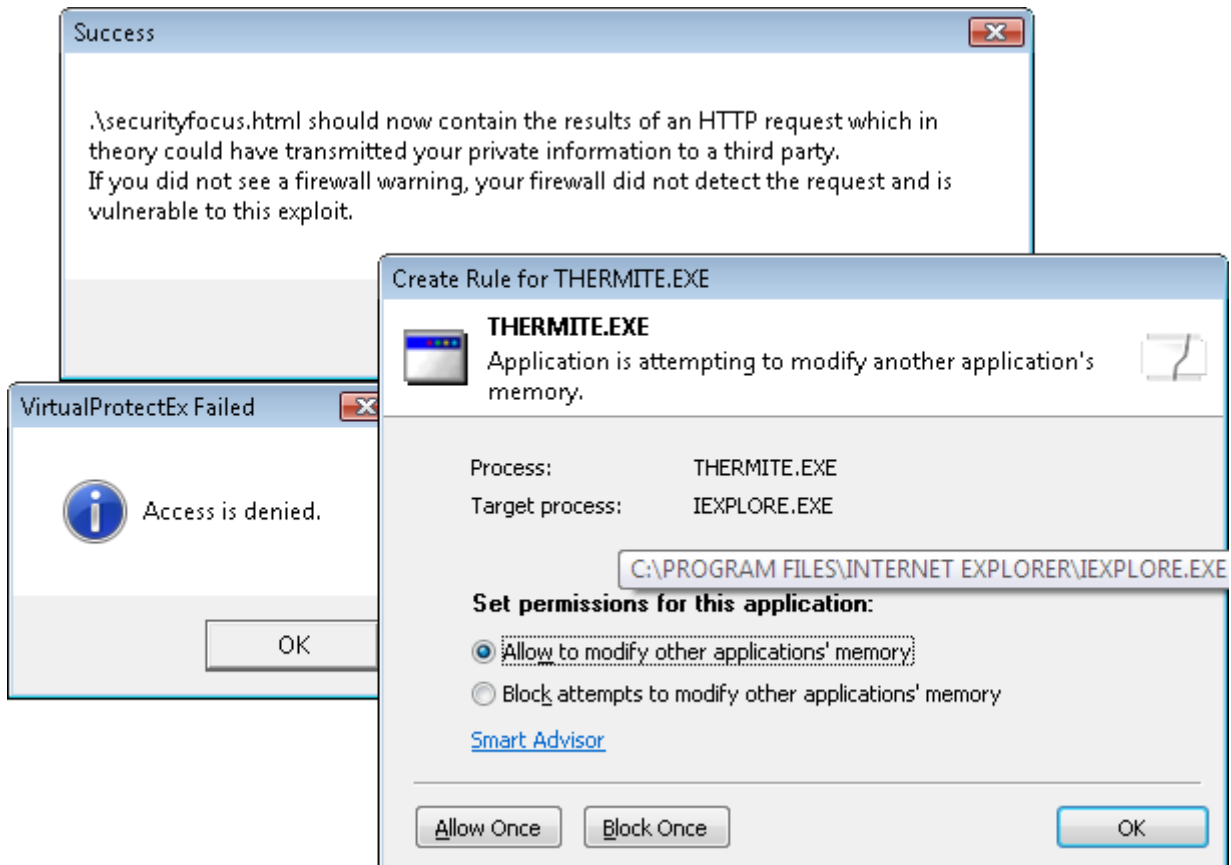
## 20. Thermite

Homepage: <none>

Direct download link: [Test1](#)

Penetration technique used: Process injection

Unlike component injection, where a foreign DLL is implanted into a legitimate application, the entire hostile process is injected into the memory of a running trusted application. Thermite acts by injecting its entire content into Internet Explorer, creating a new thread on it, and instructing the compromised browser to access a specified web site. Outpost detects this:



### **Implication:**

Application hijacking through direct process injection can be used by malware to manipulate a legitimate application and use its rights to send sensitive data to a third party.



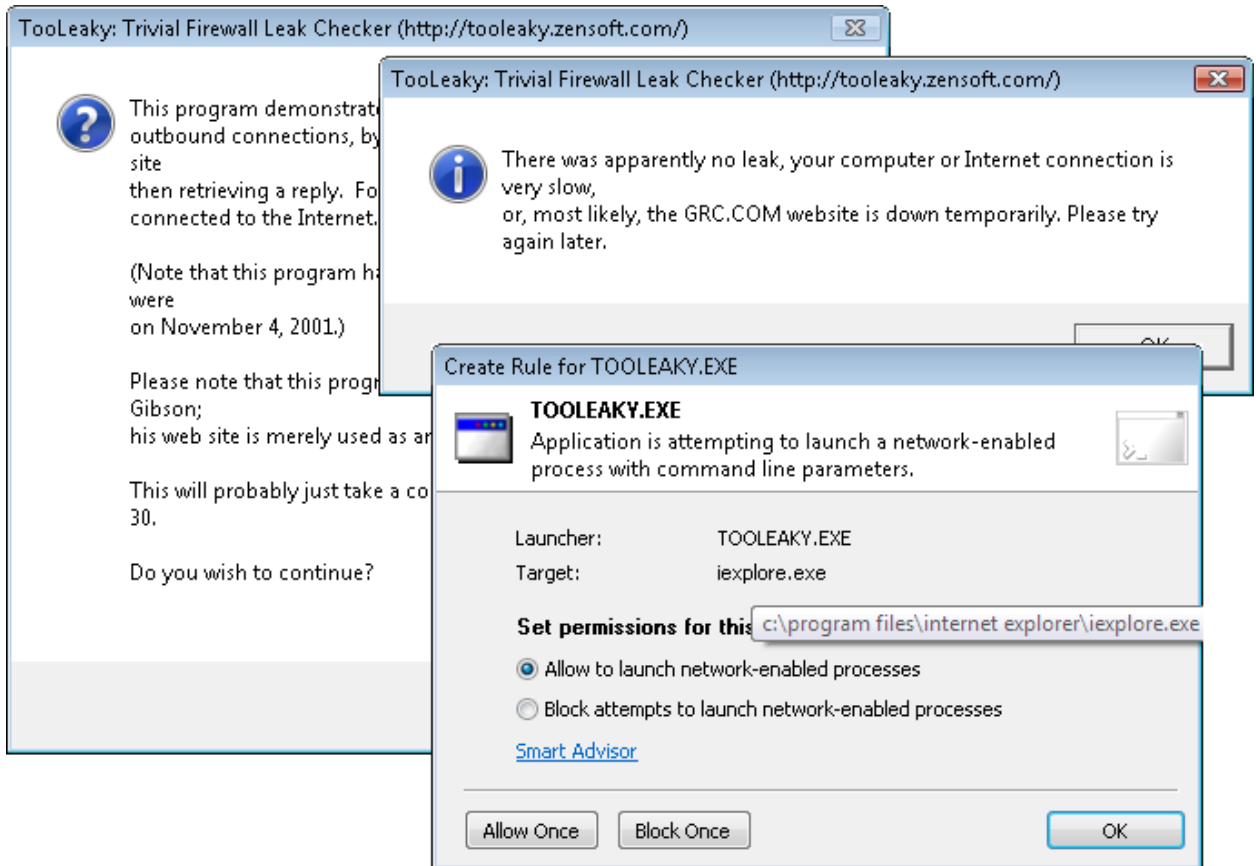
## 21. TooLeaky

Homepage: <http://tooleaky.zensoft.com>

Direct download link: [Test1](#)

**Penetration technique used:** Trusted process launch with parameters

The test launches Internet Explorer in a hidden window with a specific URL parameter. The test did not run correctly on the test machine, but Outpost still intervened and promptly stopped this attempt:



**Implication:**

Outpost controls when a program launches another program with additional parameters that may constitute a threat.



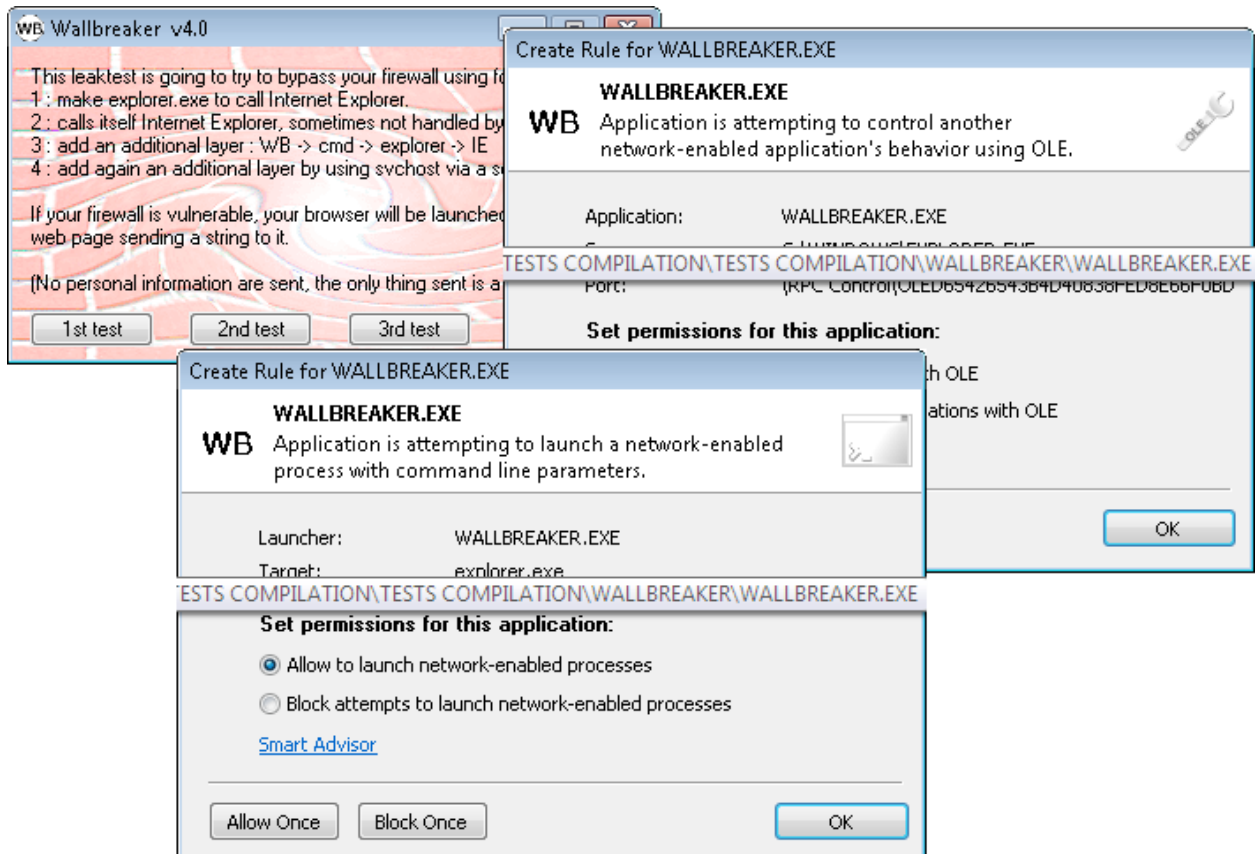
## 22. WallBreaker

Homepage: <http://www.firewallleaktester.com>

Direct download link: [Test1](#)

**Penetration technique used:** Trusted process launch with parameters; Using DDE and OLE automation to control applications.

The test consists of four subtests designed to verify the firewall's performance. Tests 1, 3 and 4 work by loading the default web browser (Internet Explorer) through various DDE commands and ordering it to access a test location. Test 2 works by simply loading Internet Explorer with additional parameters. Here's how Outpost responds:



**Implication:**

Outpost Firewall Pro detects all attempts by the WallBreaker test to get past the firewall, effectively protecting users against these types of interaction techniques.



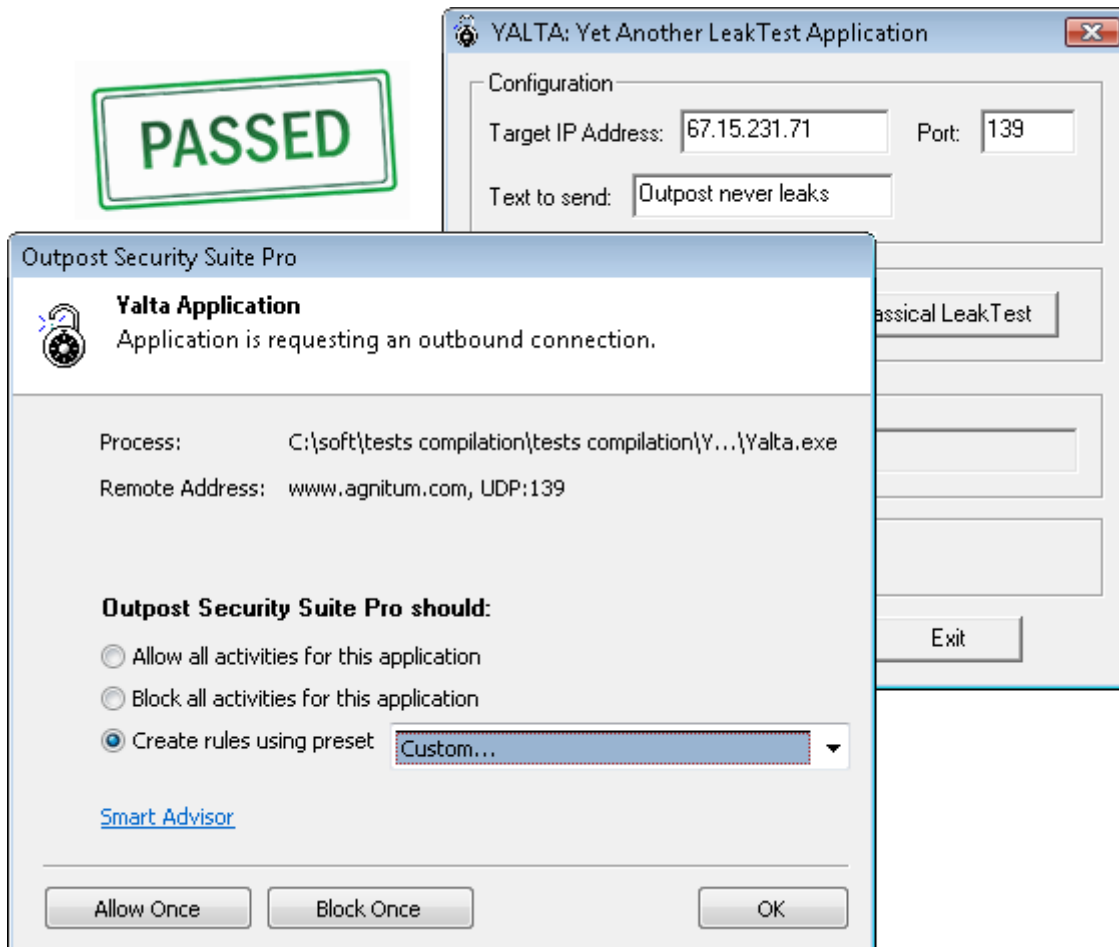
### 23. YALTA (Yet Another Leak Test Application)

Homepage: [http://www.soft4ever.com/security\\_test/En](http://www.soft4ever.com/security_test/En)

Direct download link: [Test1](#)

**Penetration technique used:** Trusted process launch with parameters; Deployment of a new network protocol.

The first test simply tries to impersonate a trusted program and access the network with special parameters that users can designate on their own. The second test involves injecting a new protocol driver into the system and redirecting all traffic to this new channel that is intended to bypass firewall filters. This second test works only on Windows 95 and Windows Me, and failed to start on our test system. Here's how Outpost responds to the first test:



**Implication:**

Outpost checks whether a trusted application can be started with extra parameters that differ from the default.

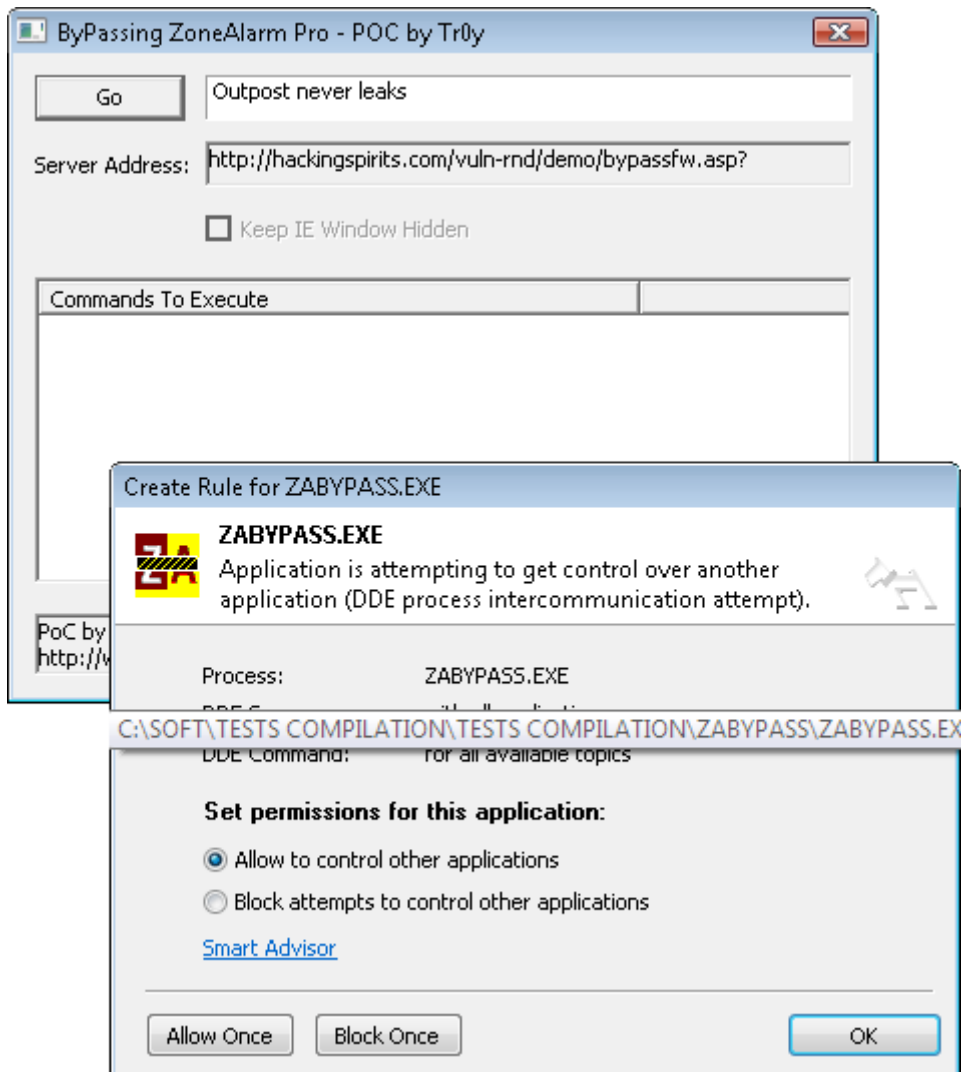
## 24. ZAbypass

Homepage: <http://www.hackingspirits.com>

Direct download link: [Test1](#)

**Penetration technique used:** Using DDE and OLE automation to control applications.

The test communicates with the trusted Internet Explorer browser and directs it to access a specified location through DDE communication. Here's how Outpost reacts:



**Implication:**

DDE communications are monitored by Outpost so that no unauthorized activity can take place.

## Independent test results

Now we've completed the in-house tests, let's take a look at some results obtained by independent testing organizations.

The pioneer in the field of leak testing is the [firewallleaktester.com](http://firewallleaktester.com) website which, despite some outdated data, still provides a wealth of information on firewalls and tools that test their performance. The site is an excellent starting point to learn about leak tests and to track firewalls' past performance.

Currently, one of the most pre-eminent sources of firewall leak testing information is Matousec Transparent Security, which extensively tests firewalls for outbound security effectiveness and provides current ratings for all major firewalls. The company is also the original author of a couple of the most advanced leak tests noted in the previous section.

The most [recent round of testing at Matousec](#)<sup>1</sup> awarded Outpost Pro a perfect 100 percent score when pitted against all currently-known leak tests. This result corroborates our own internal analysis and serves as further proof of Outpost's ability to counter unknown threats, including those represented by 0-day malware. 0-day threats cannot be detected by conventional malware scanners, so there is a growing need to protect computers with proactive defense that can stop these dangers from entering, activating and operating on a PC.

Other well-known firewalls did not fare so well. While this does not mean that lower-scoring products are not serious about outbound protection; however it does indicate that their focus is more on preserving ease of use than providing the maximum level of outbound protection. To ease the burden on the end user while providing the same exceptional level of outbound security, Outpost's ImproveNet and automatic application of access configuration shift the burden of responsibility for decision-making relating to firewall prompt responses largely to the experts at Agnitum and away from the user.

Users can therefore be assured that Outpost not only delivers superior proactive protection against unknown threats but is also sufficiently easy to use that even inexperienced PC users will benefit from that protection right out of the box.

## Conclusion

The tests documented in this paper are clearly not simply abstract lab experiments that cannot be applied in real life. Outpost's ability to pass all leak tests is further evidence of efficacy that users should take into consideration when evaluating available firewall protection.

## Contacts

Agnitum Ltd.

Bolshoy Sampsonievskiy 60, Liter "A"

St.Petersburg, Russia, 194044

Web: [www.agnitum.com](http://www.agnitum.com) Email: [pr@agnitum.com](mailto:pr@agnitum.com)

© 2007 Agnitum Ltd. All rights reserved. Agnitum® and Outpost Firewall Pro™ are trademarks or registered trademarks of Agnitum Ltd. Permission to copy all or part of this work is granted, provided that the copies are not made or distributed for sale, and that the copyright notice and this notice are retained.

---

<sup>1</sup> Results accurate as of December 20, 2007