

[REVS] Backdoor Spotcom Analysis

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-11/0106.html>

From: SecuriTeam (*support_at_securiteam.com*)

Date: 11/27/03

To: list@securiteam.com

Date: 27 Nov 2003 12:09:10 +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>

Backdoor Spotcom Analysis

SUMMARY

Spotcom is a backdoor client application that allows a hacker to control infected machine from the internet. It uses a fake DNS server for communication. The server IP address (218.242.252.211) is hard-coded in the program body. This particular piece of software was installed on machine via a bug in IE. Although it is not limited in this form of distribution, it still needs one as it includes no spreading mechanism in itself. Spotcom installs itself as a Windows NT service by replacing itself with a valid Microsoft service QOS RSVP. Service installation requires system level privileges. If the system is configured properly, this program is not a threat. The program also uses some other NT-specific win32 API calls, which makes it only available on NT/W2K/XP (and maybe later).

DETAILS

The program is distributed in two files:

%sysdir%\msrsvp.exe

%sysdir%\olegui.dll

It may also create one or more files of the following name:

%sysdir%\wins\logX.txt

Securiteam: [REVS] Backdoor Spotcom Analysis

Where X presents a physical drive letter (C, D, and so on).

Uninstallation:

Replace the path %sysdir%\msrsvp.exe with %sysdir%\rsvp.exe in QOS RSVP service and delete files %sysdir%\msrsvp.exe and %sysdir%\olegui.dll. Note that the installation allows taking over any service, not only RSVP.

msrsvp.exe is also capable of creating a service of its own, named as "IIS publishing service" (more on this in chapter 3.).

Defense:

Close direct outbound access from clients (DNS, HTTP, FTP..).

System installation:

The two executable files are packed as one executable using a custom binder application. The binder was named as "web2.exe", it was packed using the UPX executable packer and its size is 36864 bytes. The binder extracts the files msrsvp.exe and olegui.dll and executes msrsvp.exe with argument "replace rsvp", which replaces the executable path in Microsoft QOS RSVP service (for more information of msrsvp.exe arguments, see chapter "msrsvp.exe arguments"). After that, it starts the service (chapter

OK.. What's this? Jarkko guesses it is some sort of remote file fetching implementation. It may even be a buggy http client implementation (server may request a normal "GET /file.exe HTTP/1.0" stuff in authentication string). But what's strange, the client throws the authentication string with two NULL-bytes in front (\0\0string). My apache server was very upset about that. Jarkko was able to fetch files with custom server implementation using netcat. But that sounds very strange, why in earth the author has chosen this instead of normal HTTP? This function may be worth of further examination.

Conclusion:

Jarkko was really impressed about couple of things in this backdoor. First thing was the file system monitoring feature described in chapter "File system monitoring". Using this simple technique, it is possible to spy on every single filesystem operation in system. In effect, this gives the hacker quite good overview what's interesting in this particular system. Another thing that caught my attention was a fair amount of work that the program does for keeping it stealthy – DLL injection and using DNS port for reverse communication channel. The DLL injection is becoming de-facto in Trojan industry and they are now really looking for alternative communication channels instead of normal "direct" connection to client or using reverse channel via HTTP. This is getting scary. What about this type of program that uses real DNS protocol for communication? Registering an anonymous subdomain with full NS records is not a big deal.. Do you allow recursive DNS queries from workstations? Is your network safe? Well, Jarkko hopes that we will never see this kind of super Trojan.

This implementation also suffers from a couple of flaws. First off, it does not implement any kind of code-obscuring techniques. It is all right

Securiteam: [REVS] Backdoor Spotcom Analysis

there, ready to disassemble and read. Or maybe it does, Jarkko is just missing it for that reason. Second thing, there is at least one entry level programming error in the code (IP address handling of `gethostbyname()`, see chapter "Communication protocol"). This is very strange, because the IP address is determined successfully with the very same routines elsewhere in the code (see chapter "File system monitoring" for example). This makes me wonder, is this some development version released in hurry or impatient cut-and-paste? Anyway, the idea and design is still very good.

ADDITIONAL INFORMATION

The original document is available at:

<http://www.klake.org/~jt/malware/spotcom/>
<http://www.klake.org/~jt/malware/spotcom/>.

The information has been provided by <mailto:jt@klake.org> Jarkko Turkulainen.

=====

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.