

RogerWilco: new funny bugs

Source: <http://www.derkeiler.com/Mailing-Lists/Full-Disclosure/2004-03/1571.html>

From: Luigi Auriemma (aluigi_at_altervista.org)

Date: 03/31/04

Date: Wed, 31 Mar 2004 20:11:46 +0000

To: bugtraq@securityfocus.com, bugs@securitytracker.com, news@securiteam.com, full-disclosure@lists.securityfocus.com

#####

Luigi Auriemma

Application: RogerWilco

<http://rogerwilco.gamespy.com>

Versions: – RogerWilco <= 1.4.1.6

– RogerWilco Base Station <= 0.30a

Platforms: Windows, MacOS, Linux and FreeBSD

Bugs: A] Crash with malformed UDP packet

B] "Voices from the deep" bug

C] Privacy problems

D] Annoying attacks

Risk: (not needed)

Exploitation: remote, versus server and client (channel broadcast)

Date: 31 Mar 2004

Author: Luigi Auriemma

e-mail: aluigi@altervista.org

web: <http://aluigi.altervista.org>

#####

1) Introduction

2) Bugs summary

3) Bugs details

4) The Code

5) The Code examples

6) Fix

#####

=====

1) Introduction

=====

RogerWilco is a voice chat application running on Windows and MacOS

Full-Disclosure: RogerWilco: new funny bugs

but are also available dedicated servers (called "Base Stations") for Windows, Linux and FreeBSD x86.

The program is distributed by Gamespy, is no longer supported and is affected by some critical security bugs but is also still used by a lot of people.

RogerWilco is full of security bugs very interesting to study and this time I want to talk about some types of bugs not caused by programming errors but by how the software has been designed.

To be more comprehensible this advisory/paper is divided into 2 sections, one with a quick summary of the vulnerabilities and another with all the details.

#####

=====

2) Bugs summary

=====

----- A] Crash with malformed UDP packet -----

A special crafted UDP packet (big and with some big values in it) sent to the UDP audio port of RogerWilco will immediately crash the server or the client.

----- B] "Voices from the deep" bug -----

Is possible for anyone to talk into a channel without being into it but simply sending the audio stream directly to the server or to a specific client inside the same channel.

The audio stream will be transmitted to anyone in the channel or also only to a specific user or group of users.

Only transmission is possible, not reception.

----- C] Privacy problems -----

Both client and server report a lot of informations, the server for example shows all the IP addresses and port used by clients and clients show the server IP to which they are connected.

----- D] Annoying attacks -----

Full-Disclosure: RogerWilco: new funny bugs

The dedicated server shows the message "nothing read from recv" when someone connects to its port 18009 and disconnects without sending data.

Making a lot of empty connections the server's administrator will be flooded by these messages.

The GUI application refreshes its entire window when a user enters, exits or changes his nickname. If someone changes his nickname infinitely times all the users in the same channel will have some bad effects as the impossibility to take the control of their application.

#####

=====

3) Bugs details

=====

----- A) Crash with malformed UDP packet -----

RogerWilco uses an UDP port for the transmission and the reception of the audio stream.

Each UDP packet is composed by a first byte that is ever 0x0f and then by the name of the channel to which transmitting the sound followed by a NULL byte.

Then are located the informations about what user or users must receive the audio stream and also if must be retransmitted.

The users who must receive the data (forwarded by the server) are listed using fields of 16 bits that contain their IDs (each user receives an ID assigned by the server when he joins).

The last piece of the packet is the audio data block.

A quick example of audio packet is the following:

```
"\x0f" // ever 0x0f
"channel\0" // name of the channel in which trasmitting the stream
"\xff\xff" // this data "should" represent the type of transmission
"\x7f" // as above, I don't have details (not important here)
"\x00" // I don't know its usage (not important here)
"\x01" // number of target IDs (server excluded), max 127
"\x00\x00" // ID 0, it is the server's ID (who must receive the data)
"\x00\x01" // ID 1, the user with ID 1 (who must receive the data)
"data..." // audio stream
```

Now, RogerWilco manages the packet in the following mode: the program arrives to read until the "number of target IDs" and then starts a loop to read all the 16 bits fields after it containing the target IDs.

The following is the piece of code doing that (from RWNED.DLL):

Full-Disclosure: RogerWilco: new funny bugs

```
:100050BF 668B06 mov ax, word ptr [esi]
:100050C2 50 push eax
:100050C3 E81C1D0000 Call 10006DE4 (WSOCK32.ntohs)
:100050C8 8B4D58 mov ecx, dword ptr [ebp+58]
:100050CB 83C602 add esi, 00000002
:100050CE 66890479 mov word ptr [ecx+2*edi], ax
:100050D2 8B442418 mov eax, dword ptr [esp+18]
:100050D6 47 inc edi
:100050D7 3BF8 cmp edi, eax
:100050D9 7CE4 jl 100050BF
```

If an attacker sends a big channel name (as 924 chars) specifying the presence of 127 IDs BUT without adding them to the packet, the program will read from a non allocated memory zone (ESI pointer).

In the dedicated server the crash happens at offset 100050BF of RWNED.DLL while in the GUI program it happens at offset 1000544B of NETWORK.DLL (the vulnerable instructions are the same).

B] "Voices from the deep" bug

RogerWilco is composed by a TCP and an UDP section, the first is used to manage users, nicknames, IDs, accesses and other things while the second is only used for the audio stream.

The nice fact is that is not needed to join a channel (TCP section) to transmit the own audio but is only needed to send the UDP stream to the server that will manage it normally.

This "structure" lets anyone to talk anonymously into any server's channel without being stopped and without limits because limits are managed in the TCP section, so for example we can put our voice in a server also if it is password protected.

The only 2 small and almost unexistent limits I have found are that the data can be only sent and not received and that is needed to know the IDs of the users inside the channel to let the stream to reach them.

RogerWilco supports a maximum of 127 IDs for each sound stream (look the explanation of the previous bug) so if is impossible to enter in the channel of the server to get all the real user IDs (for example because it is protected by an unknown passowrd), exists ever the possibility to use the IDs from 0 to 127 because IDs are sequentials and are ever reused so I "think" is rare to find a server with users having IDs over 127.

C] Privacy problems

Full-Disclosure: RogerWilco: new funny bugs

When an user enters in a channel, the server immediately sends to him all the list of users inside the channel with their IDs (tag 0x0a0f), nicknames (tag 0x0c0f) and moreover their source IP addresses and ports (tag 0x0f0f).

The same happens if we try to enter in a client (exactly as we do with servers because the UDP and TCP ports are EVER opened) in fact we will receive the tag 0x010f showing the IP of the server in which the user is talking.

The result is that if an user is talking in a channel and don't like what another user is saying, he can easily cause damage exactly to him.

D) Annoying attacks

There is not too much to say about these so called "attacks", in fact the message "nothing read from recv" is shown into the dedicated server console when a client connects to the port 18009 and disconnects without sending data.

The port 18009 is something like a mini web server showing the current channels hosted on the dedicated server and some other informations.

Instead more interesting is the problem of the GUI program, in fact the tag 0x100f is used just by the users to change their nicknames while they are talking in a channel.

Changing the own nickname continually will create some visualization effects to the other users because the window of the program will be recreated each time and the users cannot control the program during this boring refresh.

#####

4) The Code

"Testing tool for RogerWilco 0.4" released:

<http://aluigi.altervista.org/poc/wilco.zip>

#####

5) The Code examples

The following are some quick and simple examples of how to test all the problems I have described in this and in the previous advisories for RogerWilco using my proof-of-concept.

Full-Disclosure: RogerWilco: new funny bugs

"server" and "client" are the IP or the hostname of the host we wanna test (as localhost):

A] Crash with malformed UDP packet

```
wilco -10 server  
or  
wilco -10 -p 3783 server
```

B] "Voices from the deep" bug

```
wilco -9 server
```

then we must connect our RogerWilco client to localhost:3780/CHANNEL where CHANNEL is the channel in which we wanna transmit our sound. Then we simply need to use our client normally (for example hitting F12 to talk).

```
wilco -8 -c mychannel server
```

this option will transmit a bad and annoying noise sound to all the users into "mychannel".

C] Privacy problems

```
wilco server  
or  
wilco client  
or  
wilco -p 3783 server  
or  
wilco -p 3783 -c mychannel -n yournickname server
```

The tool will show all the informations received from the server or the client.

We can also use the proxy option that can be used with a RogerWilco client to get the informations in real-time:

```
wilco -x server
```

D] Annoying attacks

Full-Disclosure: RogerWilco: new funny bugs

wilco -6 -p 3783 server
and
wilco -7 server
or
wilco -7 client

#####

=====
6) Fix
=====

RogerWilco is no longer supported.

#####

Luigi Auriemma
<http://alugi.altervista.org>