Control System Attack Vectors and Examples: Field Site and Corporate Network

Eyal Udassin
C4 Security
Ramat Hasharon 47114, Israel
eyal.udassin@c4-security.com

Abstract: SCADA systems directly influence the lives and wellbeing of all civilians in almost any modernized country. The best site for an attacker to compromise in order to cause maximum damage is the control center. Much like Aikido, an attacker can use your strengths (centralized management of assets, multiple control applications) to his benefit.

In the whitepaper we will show twp possible attack scenarios on the control center:

1. Attack from the field – how to study the weaknesses and exploit the control server software
2. Attack from the corporate network – research and vulnerability assessment of Proficy software to use it to take over the only authorized connection to the control center

As the aim of the whitepaper is to be as realistic as possible, we will demonstrate all three vectors on a very common target – an oil distribution company running GE Fanuc software: Cimplicity 6.1 for their control system and Proficy Information Portal 2.6 for reports.

Keywords: Application Security, Heap Overflow, Historian, Field Site, Fuzzing Protocol Analysis

1 Introduction

The SCADA domain is a high-profile target for terrorist groups aiming their attacks at the civilian population of developed countries. The implication of this from the security aspect is that the attacker is not your ordinary "arbitrary 14 year old internet hacker", but a well-funded, well equipped and highly motivated group of computer and control experts.

According to Dr. Boaz Ganor, director of the Institute for Counter-Terrorism (ICT), the definition of terrorism is "the deliberate use of violence aimed against civilians in order to achieve political ends" [1]. SCADA systems directly influence the lives and well being of all civilians in almost any modernized country, and the best site for an attacker to control in order to cause maximum damage is the control center. Much like Aikido, an attacker can use your strengths (centralized management of assets, multiple control applications) to his benefit.

In the whitepaper we will show two possible attack scenarios on the control center:

1. Attack from the field – how to study the weaknesses and exploit the control server software
2. Attack from the corporate network – research and vulnerability assessment of Proficy software to use it to take over the only connection to the control center.

As the aim of the whitepaper is to be as realistic as possible, we will demonstrate all three vectors on a very common target – an oil distribution company running GE Fanuc software: Cimplicity 6.1 for their control system and Proficy Information Portal for reports.

2 Attack from the Field

2.1 Introduction

The control server functions as a central hub, controlling multiple field device nodes. The SCADA environment faces a unique challenge to physically secure the network itself as the nodes are scattered in a large geographical area. In many cases the field devices are located in unmanned, remote installations, where they are located in locked cabinets and surrounded by fences. Both these physical security measures cannot thwart a determined attacker. The low investment in physical security of these sites is because rarely can a single device cause substantial, system-level damage, as well as cost to secure the large number of physical sites. This reasoning does not take into consideration a now well-known [2] attack vector, which uses the remote device's network connection to attack the control center itself.

The research we conducted in order to implement the above mentioned attack vector was focused on GE-Fanuc's Cimplicity HMI 6.1 [3] with no service packs and with the latest service pack – SP5. Our lab used the demo project which is provided by the trial version of Cimplicity to simulate a pipeline control center, which uses the Modbus protocol to control remote devices.

2.2 Technical Details

Upon starting the Cimplicity project, the following occurs:

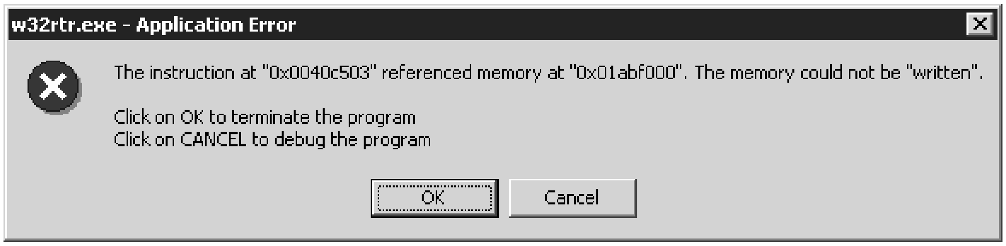
- A Web server starts, listening on two ports. One for HTTP and the other for an XML encapsulated data channel.
- The viewer application launches 2 processes.
- Cimplicity.exe launches w32rtr.exe, which in turn launch pm_mcp.exe.
- Pm_mcp launches no less then 10 child processes.
- W32rtr.exe listens on two TCP ports: 32000 and 32256.

This architecture is usually good news for an attacker, as the probability of implementing such a complex design without bugs is low. On the other hand, such code complexity will prolong the reverse engineering process, should it be required.

Our methodology for revealing vulnerabilities in proprietary applications and protocols relies on an escalation of the expertise level involved in each attempt to reveal security flaws. After learning how the application operates, we first use fuzzers [4], both in-house and open source, which are configured to simulate corrupt data with basic similarity to the original protocol/application format.

Next, we study the protocol, analyze its fields and examine whether a specific message or layer can be used for attacking the system. Last we reverse engineer the client and server applications, the reverse engineering is much more efficient when it's performed after the two previous attempts, as the reverser is now more familiar with both the application and its communication protocol.

In the case of Cimplicity HMI 6.1, we hit home-run on the first strike. After just 10 (!) seconds of running our fuzzer, the application crashed unexpectedly (Figure 1).



w32rtr.exe - Application Error

The instruction at "0x0040c503" referenced memory at "0x01abf000". The memory could not be "written".

Click on OK to terminate the program
Click on CANCEL to debug the program

Figure 1 -W32RTR Crash

Analysis of the crash (using SoftICE [5]) uncovered that the crash was caused due to a heap corruption, and that sending over 2216 bytes to TCP port 32000 will result in this crash. This happens since the program reads 0x1000 bytes from the listening socket, and copies them to a fixed size buffer on the heap which is only 0x8a8 (2216 decimal) bytes long.

The crash would prove to be very annoying to the operator, as cimplicity.exe continues to run, just without w32rtr.exe. In order to return the system to normal behavior the operator must manually terminate the web server, viewer and Cimplicity server processes, in the optimistic scenario where he can identify them quickly enough before any damage occurs.

The heap overflow identified in w32rtr.exe can be used not only to crash Cimplicity, but also to execute arbitrary code on the attacked computer. To summarize David Litchfield's excellent guide to this issue [6], code execution by exploiting heap overflows is performed in the following manner:

> The heap is managed as a doubly linked list. When a heap is first created there are two pointers that point to the first free block, and as elements are added the pointers are updated to point to the previous and next element. As more allocations and frees occur these pointers are continually updated and in this fashion allocated blocks are tracked in a doubly linked list.

When a heap based buffer is overflowed the control information is overwritten so when the buffer (allocated block) is freed and it comes to updating the pointers in the array there's going to be an access violation (as the address was overridden with user input).

By controlling the values of both pointers, the attacker can overwrite the data at any 32-bit address with a 32-bit value of his choosing. This can be used to update the address which will be used by the operating system's exception handler, which will be called with high probability as the heap structure was corrupted. Setting the address to the heap's address will allow the attacker to execute an arbitrary payload.

The research team at C4 constructed a packet which exploits this vulnerability to execute an application on the server. The payload can be easily adapted to open a remote shell for the attacker or to add an administrator with preset credentials. A video showing a successful attack can be obtained C4 upon request.

2.3 Impact

An exploitable vulnerability which results in arbitrary code execution exists in the W32RTR.EXE process. Basic research showed us that this software component is used as a "message router", a component in charge of sending and receiving messages from other elements of the system. As this component provides a mandatory function, it executes in both the Cimplicity HMI server and the HMI-only (operator) computers. This means that both the SCADA server and the HMI workstations are vulnerable to this attack. Should such an attack occur, it would be very difficult to understand why the software crashes, and the source of the attack. Obviously if backup servers and workstations will be activated and operational on the network, they too can be crashed within seconds.

3 Attack from the Corporate Network

3.1 Introduction

The corporate network is a far more permissive network then the SCADA/DCS network. As operations computers are intended for the control system use only, the number of applications running on them is limited, and the connections to external networks are controlled and monitored. The corporate network on the other hand, services many people with different roles, using applications from secretarial work to accounting to performance analysis. The corporate network is constantly connected to the Internet in order to allow email and web access, which in today's business environment is mandatory.

The corporate network is far more susceptible to attacks. Besides the interface with the Internet and the various hazards which it introduces, the users of the network may not have any education regarding information safety and security, and the network hosts external computers and users such as consultants. In addition, the corporate network is not considered mission-critical, so finding a computer virus every time and again is rarely considered as a catastrophic security breach.

The two networks, despite their differences in characteristics, are in many times required to interconnect. This demand arises from the needs of users who are located on the corporate network (executives, performance analysis, internal auditing) to retrieve real-time information from the SCADA system in order to perform their work. For security reasons, it is recommended to allow the corporate network to access only a single computer on the SCADA network, which functions as a reporting server or historian. This connection can be further secured by a firewall permitting only specific application protocols to reach the reporting server.

Attacking the reporting server is the "holy grail" for an attacker who wishes to penetrate the SCADA network from the corporate network [7]. The research we conducted in order to implement this attack vector was focused on GE-Fanuc's Proficy Information Portal 2.6 [8]. Our lab consisted of an "out of the box" installation of Proficy, configured to gather information from Cimplicity HMI 6.1.

3.2 Technical Details

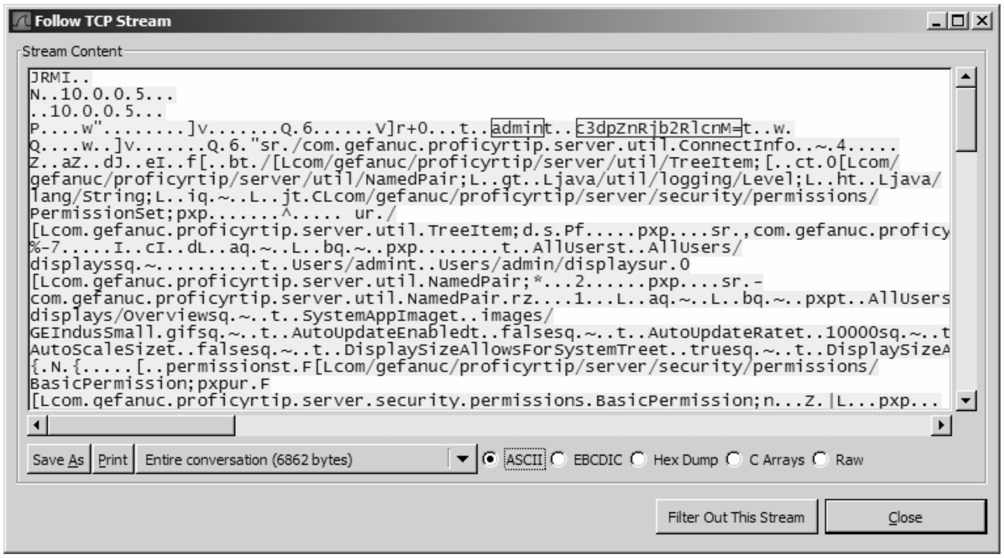
Initial interaction with the server showed that when the user connects to the Proficy web application, hosted on a standard Microsoft IIS server, the user's browser is instructed to download and execute a Java applet. All communication from that point on between the client and the server will be based on the Java RMI protocol, as opposed to the HTTP protocol normally used in web-based applications. The RMI protocol uses a non-standard port, perhaps to avoid easy detection and classification by prying eyes.

In parallel to thorough use of the Applet to access, operate and query all the possible functions that the user interface allows the user, a network sniffer recorded all data communications between the client and the server. A close examination revealed two interesting findings:

1. The packet which was identified as the login packet, showed the username in cleartext and the password in an encoding which resembles Base64 [9].
2. One of the packets which were generated by the user actions included a relative filename and a large chunk of Base64 encoded data.

Finding #1 was quickly analyzed. The Base64 data was indeed the user's password. The implication of this is that an attacker with the ability to place a sniffer on the network or the computer can quickly gain valid credentials to the reporting server. Even more interesting is that the Proficy server can be configured to use Active Directory as its authentication service, so when intercepting the Proficy authentication traffic the attacker gains knowledge of not only how to login to Proficy, but potentially to the entire network. Proficy weakens the security of Active Directory, which uses strong authentication schemes such as Kerberos.

An example authentication packet is illustrated in Figure 2.

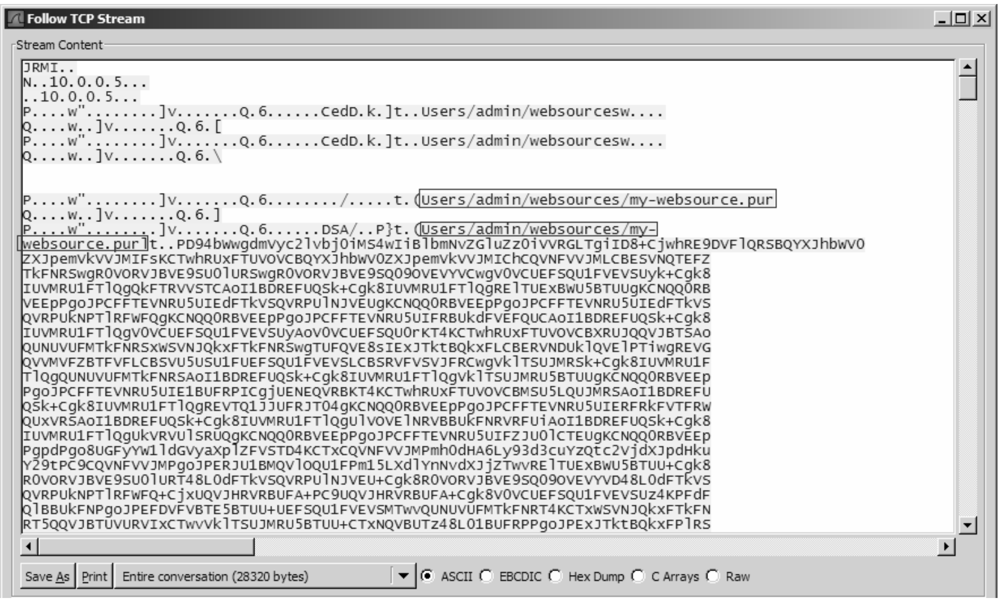


```
JRMI..
N..10.0.0.5...
..10.0.0.5...
P....w"........]v.......Q.6......V]r+0...t..admint..c3dpZnRjb2RlcnM=t..w.
Q....w..]v.......Q.6."sr./com.gefanuc.proficyrtip.server.util.ConnectInfo..~.4.....
Z..aZ..dJ..eI..f[..bt./[Lcom/gefanuc/proficyrtip/server/util/TreeItem;[..ct.0[Lcom/
gefanuc/proficyrtip/server/util/NamedPair;L..gt..Ljava/util/logging/Level;L..ht..Ljava/
lang/String;L..iq.~..L..jt.CLcom/gefanuc/proficyrtip/server/security/permissions/
PermissionSet;pxp.......^..... ur./
[Lcom.gefanuc.proficyrtip.server.util.TreeItem;d.s.Pf.....pxp....sr.,com.gefanuc.proficy
%-7.....I..cI..dL..aq.~..L..bq.~..pxp........t..AllUserst..AllUsers/
displayssq.~..........t..Users/admint..Users/admin/displaysur.0
[Lcom.gefanuc.proficyrtip.server.util.NamedPair;*...2......pxp....sr.-
com.gefanuc.proficyrtip.server.util.NamedPair.rz....1...L..aq.~..L..bq.~..pxpt..AllUsers
displays/Overviewsq.~..t..SystemAppImaget..images/
GEIndusSmall.gifsq.~..t..AutoUpdateEnabledt..falsesq.~..t..AutoUpdateRatet..10000sq.~..t
AutoScaleSizet..falsesq.~..t..DisplaySizeAllowsForSystemTreet..truesq.~..t..DisplaySizeA
{.N.{.....[..permissionst.F[Lcom/gefanuc/proficyrtip/server/security/permissions/
BasicPermission;pxpur.F
[Lcom.gefanuc.proficyrtip.server.security.permissions.BasicPermission;n...Z.|L...pxp...
```

Save As | Print | Entire conversation (6862 bytes) ● ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ○ Raw

Figure 2 – Proficy Vulnerable Login

The second finding was trickier to analyze and exploit. The suspicious packet was sent when a user switched the application to "configuration mode" and selected to add a WebSource under a certain name. When this was done, a file was created on the server with the name and a ".purl" extension. This corresponded to the filename we saw in the packet (Figure 3).



```
JRMI..
N..10.0.0.5...
..10.0.0.5...
P....w"........]v.......Q.6......CedD.k.]t..Users/admin/websourcesw....
Q....w..]v.......Q.6.[
P....w"........]v.......Q.6......CedD.k.]t..Users/admin/websourcesw....
Q....w..]v.......Q.6.\


P....w"........]v.......Q.6......../.....t.(Users/admin/websources/my-websource.pur
Q....w..]v.......Q.6.]
P....w"........]v.......Q.6......DSA/..P}t.(Users/admin/websources/my-
websource.purlt..PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiID8+CjwhRE9DVFlQRSBQYXJhbWV0
ZXJpemVkVVJMIFsKCTwhRUxFTUVOVCBQYXJhbWV0ZXJpemVkVVJMIChCQVNFVVJMLCBESVNQTEFZ
TkFNRSwgR0VORVJBVE9SU0lURSwgR0VORVJBVE9SQ09OVEVYVCwgV0VCUEFSQU1FVEVSUyk+Cgk8
IUVMRU1FTlQgQkFTRVVSTCAoI1BDREFUQSk+Cgk8IUVMRU1FTlQgRElTUExBWU5BTUUgKCNQQ0RB
VEEpPgoJPCFFTEVNRU5UIEdFTkVSQVRPUlNJVEUgKCNQQ0RBVEEpPgoJPCFFTEVNRU5UIEdFTkVS
QVRPUkNPTlRFWFQgKCNQQ0RBVEEpPgoJPCFFTEVNRU5UIFRBUkdFVEFQUCAoI1BDREFUQSk+Cgk8
IUVMRU1FTlQgV0VCUEFSQU1FVEVSUyAoV0VCUEFSQU0rKT4KCTwhRUxFTUVOVCBXRUJQQVJBTSAo
QUNUVUFMTkFNRSxWSVNJQkxFTkFNRSwgTUFQVE8sIExJTktBQkxFLCBERVNDUklQVElPTiwgREVG
QVVMVFZBTFVFLCBSVU5USU1FUEFSQU1FVEVSLCBSRVFVSVJFRCwgVklTSUJMRSk+Cgk8IUVMRU1F
TlQgQUNUVUFMTkFNRSAoI1BDREFUQSk+Cgk8IUVMRU1FTlQgVklTSUJMRU5BTUUgKCNQQ0RBVEEp
PgoJPCFFTEVNRU5UIE1BUFRPICgjUENEQVRBKT4KCTwhRUxFTUVOVCBMSU5LQUJMRSAoI1BDREFU
QSk+Cgk8IUVMRU1FTlQgREVTQ1JJUFRJT04gKCNQQ0RBVEEpPgoJPCFFTEVNRU5UIERFRkFVTFRW
QUxVRSAoI1BDREFUQSk+Cgk8IUVMRU1FTlQgUlVOVElNRVBBUkFNRVRFUiAoI1BDREFUQSk+Cgk8
IUVMRU1FTlQgUkVRVUlSRUQgKCNQQ0RBVEEpPgoJPCFFTEVNRU5UIFZJU0lCTEUgKCNQQ0RBVEEp
PgpdPgo8UGFyYW1ldGVyaXplZFVSTD4KCTxCQVNFVVJMPmh0dHA6Ly93d3cuYzQtc2VjdXJpdHku
Y29tPC9CQVNFVVJMPgoJPERJU1BMQVlOQU1FPm15LXdlYnNvdXJjZTwvRElTUExBWU5BTUU+Cgk8
R0VORVJBVE9SU0lURT48L0dFTkVSQVRPUlNJVEU+Cgk8R0VORVJBVE9SQ09OVEVYVD48L0dFTkVS
QVRPUkNPTlRFWFQ+CjxUQVJHRVRBUFA+PC9UQVJHRVRBUFA+Cgk8V0VCUEFSQU1FVEVSUz4KPFdF
QlBBUkFNPgoJPEFDVFVBTE5BTUU+UEFSQU1FVEVSMTwvQUNUVUFMTkFNRT4KCTxWSVNJQkxFTkFN
RT5QQVJBTUVURVIxCTwvVklTSUJMRU5BTUU+CTxNQVBUTz48L01BUFRPPgoJPExJTktBQkxFPlRS
```

Save As | Print | Entire conversation (28320 bytes) ● ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ○ Raw

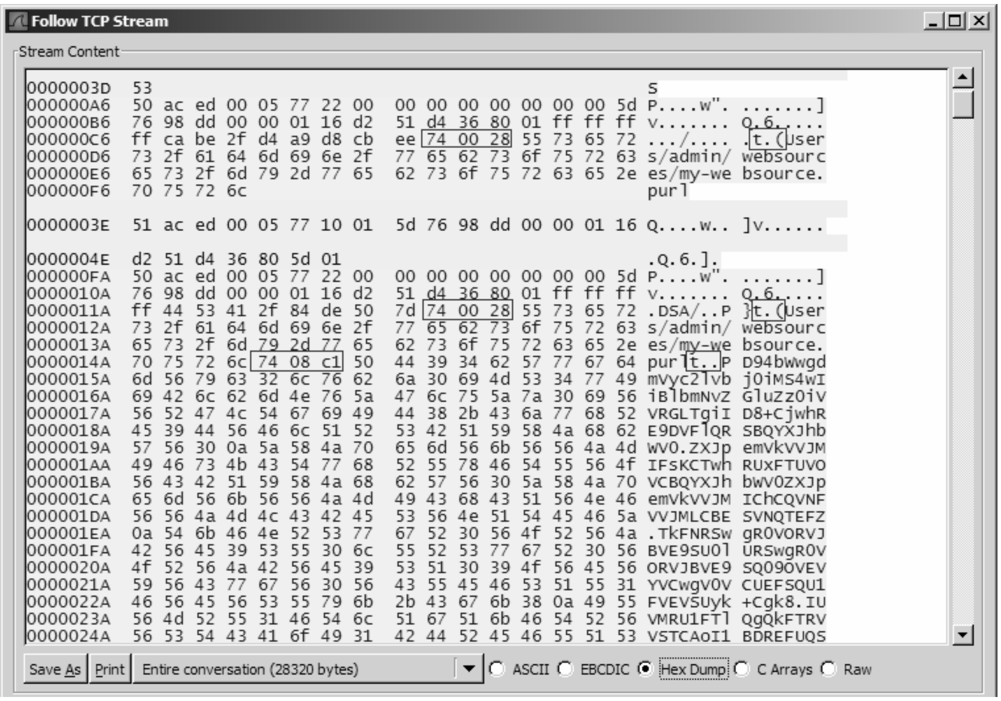
Figure 3 – Control Uploaded Filename

We created a modified packet with the same information, but with a different filename, and sent it to the server. The application on the server created the file we requested. Now we needed to find a way to control the file contents in order to successfully implement this attack vector.

The large data block in the packet which was Base64 encoded indeed included the file contents which the server created on its hard drive. Originally, the file content was XML information of the WebSource the user requested to add to his display. Our attempt to modify this data block to a Base64 version of the string "Hello!" resulted in a failure. This usually means that one or more of the following occurred:

1. The server performs sanity checks on the file contents data
2. There's a CRC check or digital signature in the packet which is no longer valid
3. The packet data includes meta information such as lengths or offsets, which are no longer valid

Analysis of the packet, and other packets which contained Base64 encoded data (including the login packet) showed that the encoded data is always preceded by the a byte with the value 0x74, which represents the letter 't' in ASCII encoding, and two bytes which represent a unsigned integer (from 0 to 65355) indicating the length of the Base64 data block. For example, observe the following packet data (Figure 4).



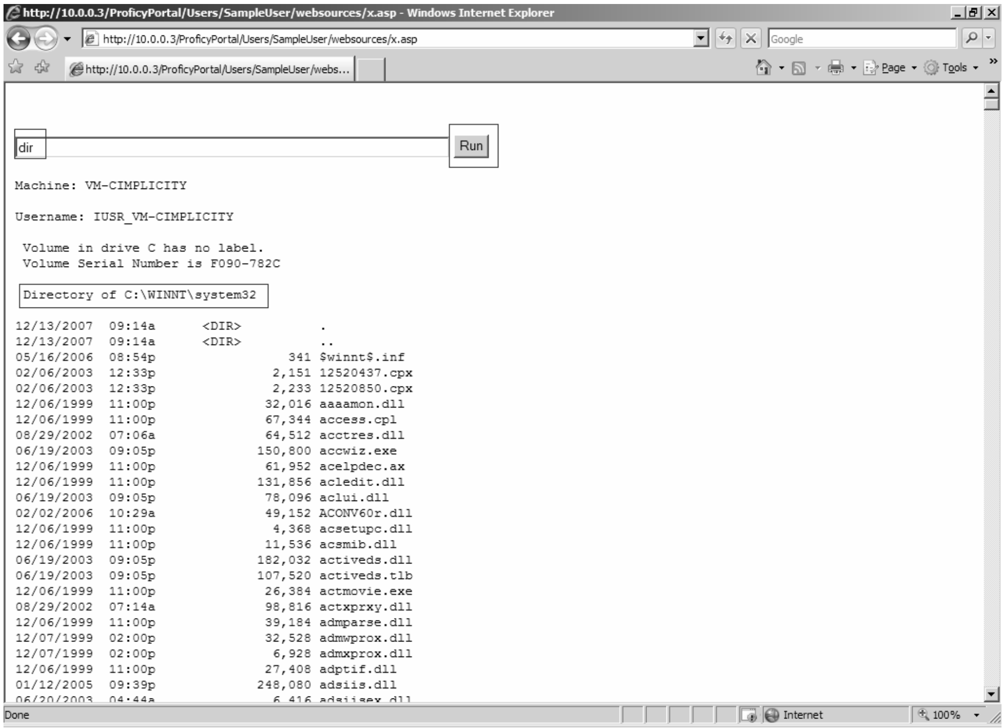
```
0000003D  53                                               S
000000A6  50 ac ed 00 05 77 22 00  00 00 00 00 00 00 00 5d P....w". .......]
000000B6  76 98 dd 00 00 01 16 d2  51 d4 36 80 01 ff ff ff v....... Q.6.....
000000C6  ff ca be 2f d4 a9 d8 cb  ee 74 00 28 55 73 65 72 .../.... .t.(User
000000D6  73 2f 61 64 6d 69 6e 2f  77 65 62 73 6f 75 72 63 s/admin/ websourc
000000E6  65 73 2f 6d 79 2d 77 65  62 73 6f 75 72 63 65 2e es/my-we bsource.
000000F6  70 75 72 6c                                      purl

0000003E  51 ac ed 00 05 77 10 01  5d 76 98 dd 00 00 01 16 Q....w.. ]v......

0000004E  d2 51 d4 36 80 5d 01                             .Q.6.].
000000FA  50 ac ed 00 05 77 22 00  00 00 00 00 00 00 00 5d P....w". .......]
0000010A  76 98 dd 00 00 01 16 d2  51 d4 36 80 01 ff ff ff v....... Q.6.....
0000011A  ff 44 53 41 2f 84 de 50  7d 74 00 28 55 73 65 72 .DSA/..P }t.(User
0000012A  73 2f 61 64 6d 69 6e 2f  77 65 62 73 6f 75 72 63 s/admin/ websourc
0000013A  65 73 2f 6d 79 2d 77 65  62 73 6f 75 72 63 65 2e es/my-we bsource.
0000014A  70 75 72 6c 74 08 c1 50  44 39 34 62 57 77 67 64 purlt..P D94bWwgd
0000015A  6d 56 79 63 32 6c 76 62  6a 30 69 4d 53 34 77 49 mVyc2lvb j0iMS4wI
0000016A  69 42 6c 62 6d 4e 76 5a  47 6c 75 5a 7a 30 69 56 iBlbmNvZ GluZz0iV
0000017A  56 52 47 4c 54 67 69 49  44 38 2b 43 6a 77 68 52 VRGLTgiI D8+CjwhR
0000018A  45 39 44 56 46 6c 51 52  53 42 51 59 58 4a 68 62 E9DVFlQR SBQYXJhb
0000019A  57 56 30 0a 5a 58 4a 70  65 6d 56 6b 56 56 4a 4d WV0.ZXJp emVkVVJM
000001AA  49 46 73 4b 43 54 77 68  52 55 78 46 54 55 56 4f IFsKCTwh RUxFTUVO
000001BA  56 43 42 51 59 58 4a 68  62 57 56 30 5a 58 4a 70 VCBQYXJh bWV0ZXJp
000001CA  65 6d 56 6b 56 56 4a 4d  49 43 68 43 51 56 4e 46 emVkVVJM IChCQVNF
000001DA  56 56 4a 4d 4c 43 42 45  53 56 4e 51 54 45 46 5a VVJMLCBE SVNQTEFZ
000001EA  0a 54 6b 46 4e 52 53 77  67 52 30 56 4f 52 56 4a .TkFNRSw gR0VORVJ
000001FA  42 56 45 39 53 55 30 6c  55 52 53 77 67 52 30 56 BVE9SU0l URSwgR0V
0000020A  4f 52 56 4a 42 56 45 39  53 51 30 39 4f 56 45 56 ORVJBVE9 SQ09OVEV
0000021A  59 56 43 77 67 56 30 56  43 55 45 46 53 51 55 31 YVCwgV0V CUEFSQU1
0000022A  46 56 45 56 53 55 79 6b  2b 43 67 6b 38 0a 49 55 FVEVSUyk +Cgk8.IU
0000023A  56 4d 52 55 31 46 54 6c  51 67 51 6b 46 54 52 56 VMRU1FTl QgQkFTRV
0000024A  56 53 54 43 41 6f 49 31  42 44 52 45 46 55 51 53 VSTCAoI1 BDREFUQS
```

Figure 4 - Text Fields Analysis

Equipped with the knowledge of how to control the filename and its contents, we proceeded to craft a packet that will create an ASP file on the server which will allow us to gain a remote shell. Our initial version of the ASP file had to be modified as the server adds a backslash to the quote and double quote marks. Finally, with a special "Proficy-proof" version of our web backdoor, we crafted a new version of the packet which contains the Base64 encoded version of the backdoor as well as an updated length field to make the packet valid. The backdoor was successfully uploaded and executed by the web browser (Figure 5), allowing us to use the reporting server as a bridge to the SCADA network.


http://10.0.0.3/ProficyPortal/Users/SampleUser/websources/x.asp - Windows Internet Explorer
http://10.0.0.3/ProficyPortal/Users/SampleUser/websources/x.asp
Google
http://10.0.0.3/ProficyPortal/Users/SampleUser/webs...
Page
Tools
dir
Run
Done
Internet
100%

```
Machine: VM-CIMPLICITY

Username: IUSR_VM-CIMPLICITY

 Volume in drive C has no label.
 Volume Serial Number is F090-782C

 Directory of C:\WINNT\system32

12/13/2007  09:14a      <DIR>          .
12/13/2007  09:14a      <DIR>          ..
05/16/2006  08:54p                 341 $winnt$.inf
02/06/2003  12:33p               2,151 12520437.cpx
02/06/2003  12:33p               2,233 12520850.cpx
12/06/1999  11:00p              32,016 aaaamon.dll
12/06/1999  11:00p              67,344 access.cpl
08/29/2002  07:06a              64,512 acctres.dll
06/19/2003  09:05p             150,800 accwiz.exe
12/06/1999  11:00p              61,952 acelpdec.ax
12/06/1999  11:00p             131,856 acledit.dll
06/19/2003  09:05p              78,096 aclui.dll
02/02/2006  10:29a              49,152 ACONV60r.dll
12/06/1999  11:00p               4,368 acsetupc.dll
12/06/1999  11:00p              11,536 acsmib.dll
06/19/2003  09:05p             182,032 activeds.dll
06/19/2003  09:05p             107,520 activeds.tlb
12/06/1999  11:00p              26,384 actmovie.exe
08/29/2002  07:14a              98,816 actxprxy.dll
12/06/1999  11:00p              39,184 admparse.dll
12/07/1999  02:00p              32,528 admwprox.dll
12/07/1999  02:00p               6,928 admxprox.dll
12/06/1999  11:00p              27,408 adptif.dll
01/12/2005  09:39p             248,080 adsiis.dll
06/20/2003  04:44a               6,416 adsiisex.dll
```

Figure 5 - Web Backdoor On Proficy Server

3.3 Impact

The two vulnerabilities in Proficy enable an attacker who gained partial or full control over the corporate network the ability to execute arbitrary code on the Proficy server, which will commonly be located on the SCADA network or a DMZ. This ability is achieved by sniffing the network until the attacker intercepts a user login to the application, which will quickly reveal the credentials of the user to the attacker. Of course a disgruntled insider who is allowed to view data on the Proficy server will already have credentials. As an authenticated user, the attacker can proceed to upload and execute his code on the Proficy server. The attack will succeed even when a well-configured firewall separates the Proficy server from the corporate network, as it is based on the same protocols which are required by Proficy to operate.

4 Conclusion

Our primary research goal was achieved; we demonstrated real, feasible attacks on a control center which generate from two potential attack vectors, the field devices and the corporate network.

Besides the opportunity to get hands-on experience with GE-Fanuc's software products, we also learned how slowly the vendor responds to critical security issues, even if it's threatening the availability of the SCADA server. We can only assume that GE-Fanuc is not an exception, as we've encountered substantial criticism about this issue in regards to other SCADA system vendors.

We would like to emphasize that additional attack vectors on the control center exist, besides the ones specified in this paper. Remote support connections and physical security should also be a primary concern when evaluating the security of the control center and its resilience to attack.

It is our hope that asset owners will realize the threat of implementation flaws of their SCADA software, and demand their vendor to review and audit its application code from a security perspective.

About the Author – Eyal Udassin is the co-founder and a security expert at C4 Security, a penetration testing company which uncovers and mitigates technical security vulnerabilities by attacking the tested system or network. He has performed hundreds of penetration tests against financial, governmental, military and SCADA clients as part of C4's red team. Mr. Udassin specializes in protocol analysis, reverse engineering of binary code, passive and active reconnaissance of computer networks, and exploiting vulnerabilities uncovered in proprietary software. Mr. Udassin is also the lead designer of proprietary software solutions offered by C4 to mitigate existing vulnerabilities in cases where the software vendors are hesitant to fix or simply went bankrupt.

References

[1] "The Relationship Between International and Localized Terrorism", Boaz Ganor, www.jcpa.org/brief/brief004-26.htm

[2] Cyber Security for Utility Operations, Sandia National Laboratories, http://www.sandia.gov/scada/documents.htm

[3] Cimplicity HMI Product Page, http://www.gefanuc.com/as_en/products_solutions/hmi_scada/products/proficy_cimplicity.html

[4] Definition of Fuzzers - http://en.wikipedia.org/wiki/Fuzz_testing

[5] Compuware (previously Numega) SoftICE - http://www.compuware.com/products/numega/index.htm

[6] Windows Heap Overflows, David Litchfield, www.blackhat.com/presentations/win-usa-04/bh-win-04-litchfield/bh-win-04-litchfield.ppt

[7] SCADA System: Evolving Cyber Security Challenges, Symantec, http://www.symantec.com/business/library/article.jsp?aid=scada_systems_evolving_cyber_security_challenges

[8] Proficy Real-Time Information Portal Product Page - http://www.gefanuc.com/as_en/products_solutions/production_management/products/proficy_portal.html

[9] Base64 Encoding RFC, http://tools.ietf.org/html/rfc4648