Remote Control and Reporting using SCADA

Michael Humphries

CNT 4101 Software Project in Computer Networks

Dr. Zalewski

Florida Gulf Coast University

12/09/2011

1. Introduction

SCADA (Supervisory Control and Data Acquisition) is a control system applied prominently in an industrial setting for equipment monitoring. It is a complex setup of computer systems that allow a remote user to control and monitor industrial infrastructures and facility based processes. SCADA systems can be found in industries such as telecommunications, water and waste control, energy, oil and gas refining and transportation. A typical SCADA setup consists of a central SCADA control host workstation, multiple Remote Terminal Units (RTU) and a local PC workstation (Fig. 1). The RTU's are collecting the analog data and converting these data to a digital format for the supervisory system, as well as are executing commands sent by a remote SCADA host.

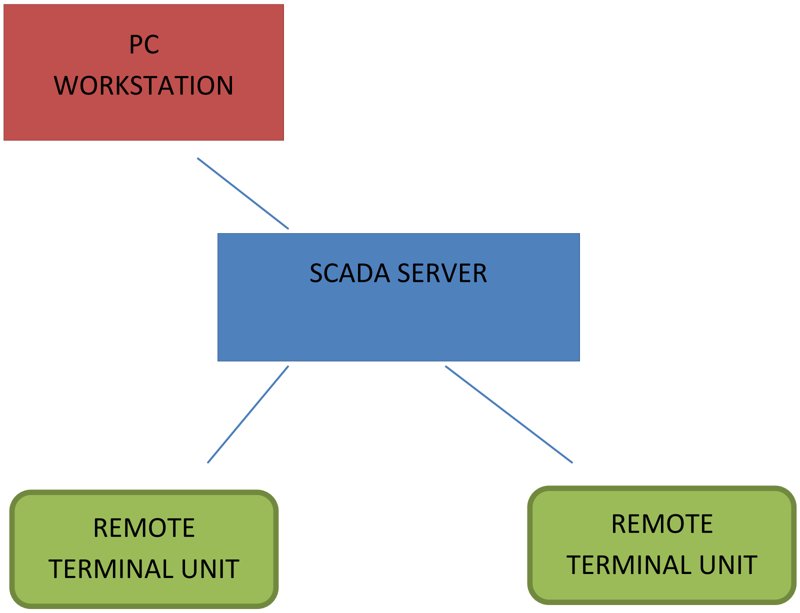


Fig. 1 Simple SCADA Setup

In the CS lab, currently there is only one Remote Terminal Unit and one SCADA control unit server. This is the minimum requirement for a SCADA control system. We do indeed have multiple remote and local workstations as to conduct test on the system. These workstations are within the same local area network and outside.

This project is a continuation of a previous project, for which documentation exists [2].

1.1 In-depth Hardware Component Description

A SCADA server, also called "Supervisory Station" is the heart of the system. The main operation of the station is to maintain communications with the field equipment such as the RTUs. This job is done with multiple components.

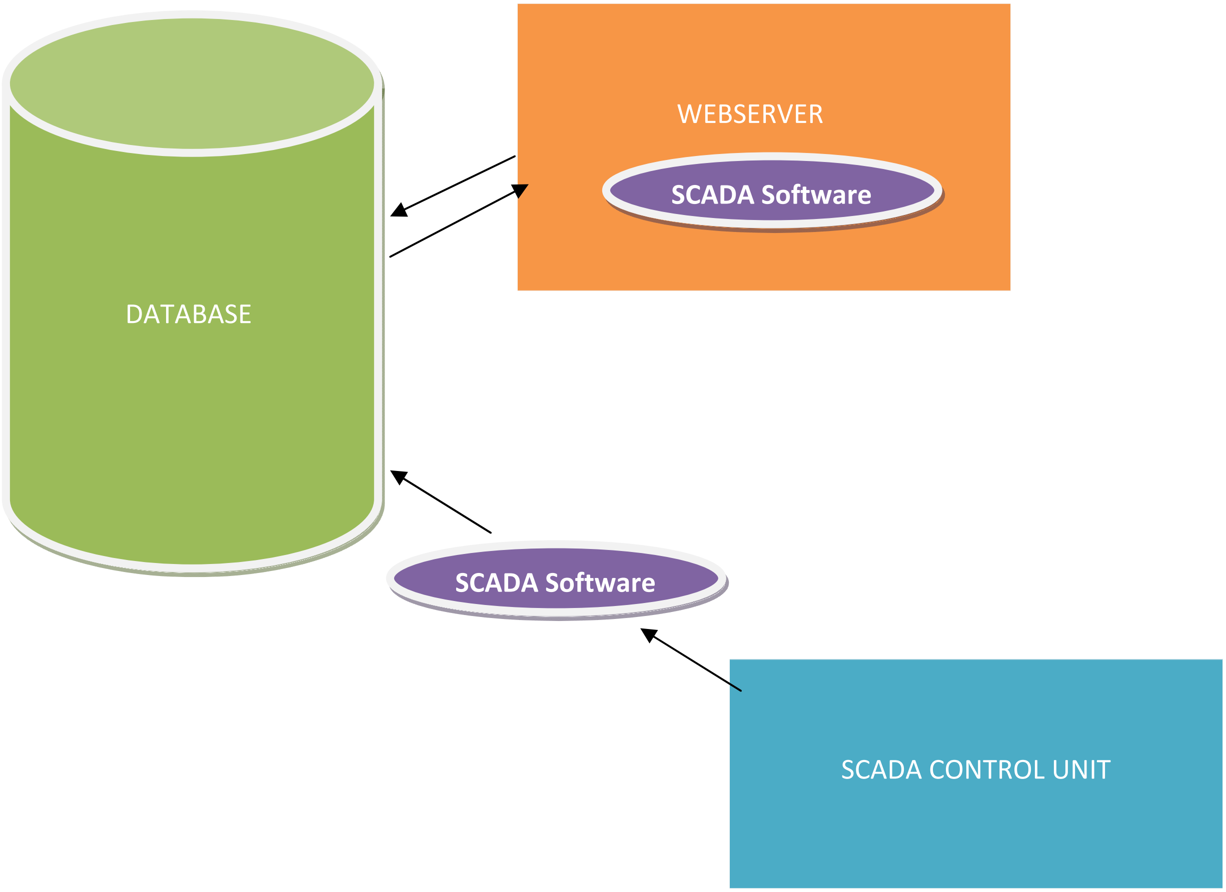


Fig. 2 Simple SCADA Server Supervisory Station Software Setup

Fig. 2, show the basic setup of the Supervisory Station in a SCADA control system. The first unit, "SCADA Control Unit" receives digital data from and sends digital signals to remote devices such as RTUs. This unit is labeled as "SCADA Control Unit" in Fig. 2. This unit then sends its received data to a database. SCADA software is used to interpret the data from the RTUs and to send these data to the database. Next, the database strikes up communication with a webserver whose goal is to receive data from the database and create a dynamic website hosted on a port (in our case: port 80). The webserver is tagged "Webserver" in the diagram displayed in Fig. 2. The SCADA software installed on the web server will send queries to the database, with which the database will respond with the corresponding requested data. The software will then make the data available for any client who has access to the same network as the webserver to view the data on a browser using port 80 and IP address or domain name.

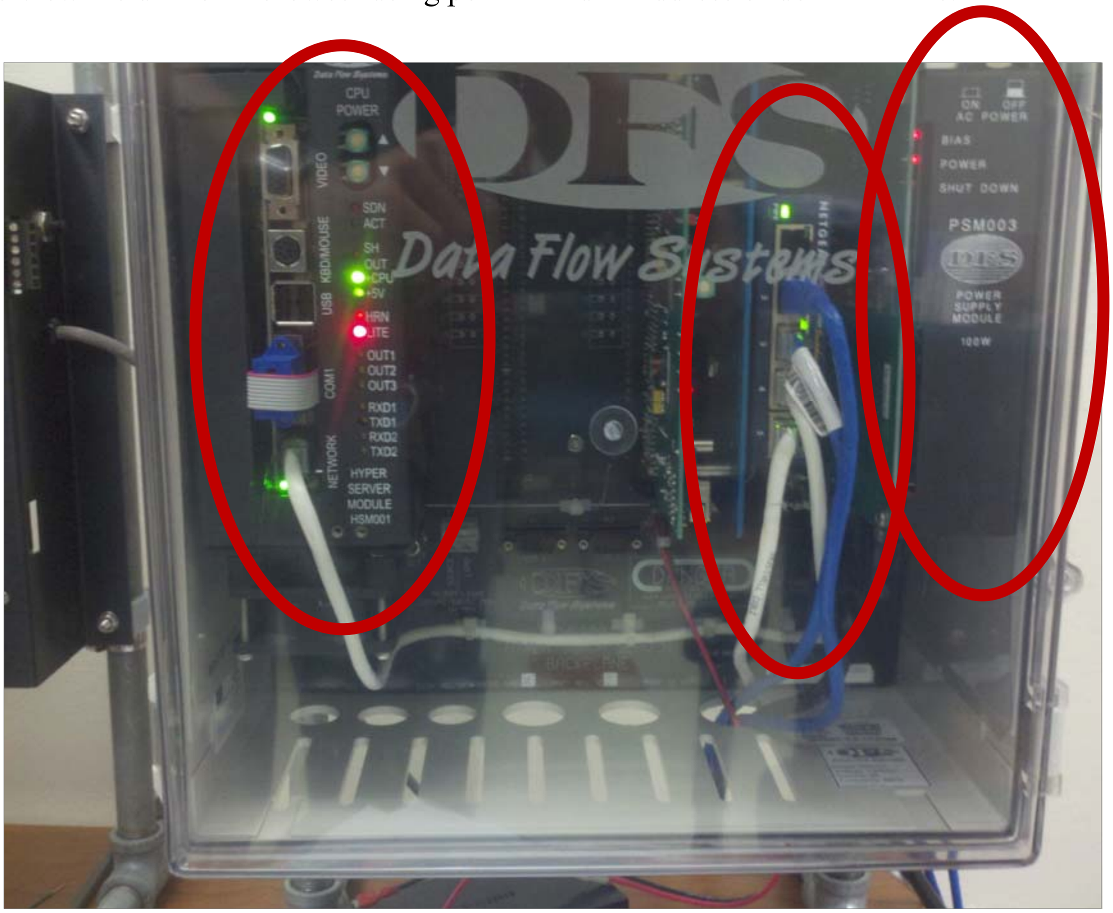


Fig. 3 The Supervisory Station at Florida Gulf Coast University

Fig. 3 shows the Supervisory Station in the CS lab at Florida Gulf Coast University. The first highlighted area on the left in Fig. 3 is the computer system where the webserver from Fig. 2 runs. This device's IP address is 69.88.163.30. Also this computer system holds the database and interprets the received data with the SCADA software. Moving on to the center circle in Fig. 3, this is a network switch. The network has a 69.88.163.1 network IP address. All the RTUs used in this setup use Ethernet as their source of communication. This Netgear Network Switch connects all the RTU’s. It also receives a connection from the local area network infrastructure of the University. Finally, the final red circle is the power supply. This has a simple On and Off switch.

RTUs are used to connect directly to physical equipment at the other end. These devices are connected to the Supervisory Station. The RTUs will take analog input and signals and then convert them into digital representations and send them to the Supervisory Station. RTUs will also receive data coming from the other direction. The Supervisory Station may tell a RTU to perform a function with the physical measure. In this case, the RTU will convert the digital signal to an analog signal, which the physical equipment can understand. Common examples of what RTUs can measure are: pressure, flow, voltage, current or open/closed status. By converting the digital signals received from the Supervisory Station the RTU can perform action such as closing a switch or a valve, or setting the speed of a pump.

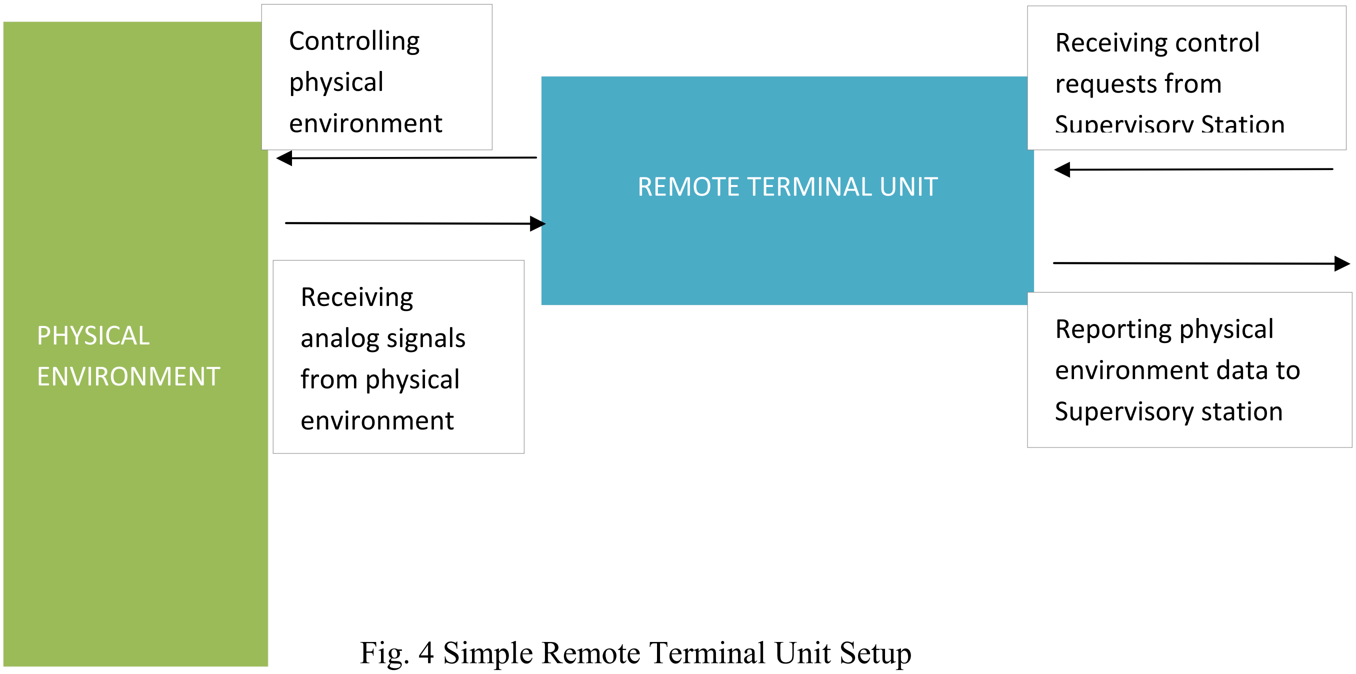

Controlling physical environment
Receiving control requests from Supervisory Station
REMOTE TERMINAL UNIT
PHYSICAL ENVIRONMENT
Receiving analog signals from physical environment
Reporting physical environment data to Supervisory station

Fig. 4 Simple Remote Terminal Unit Setup

Fig. 4 shows how a RTU interacts between the physical equipment and Supervisory Stations. Arrows to the right are data being sent to the Supervisory Station, while arrows on the left are data being interpreted from physical environment. In Fig. 5, shows a simple RTU setup at Florida Gulf Coast University. In our example we are using a control box that is controlling some kind of pump device that is not setup. The physical environment in this example is the human reaction when a person flips a switch.

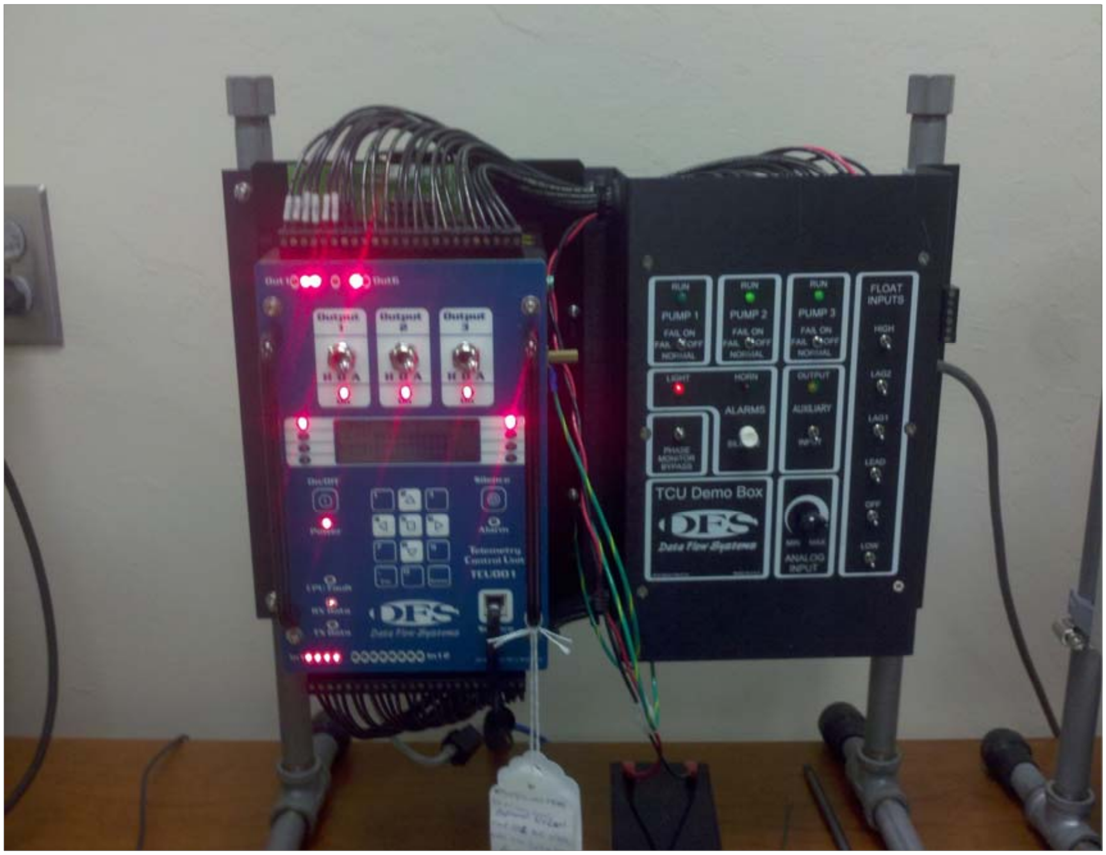


Fig. 5 Remote Terminal Unit in Florida Gulf Coast University CS lab

2. Definition of the Problem

The SCADA technology was created to provide for an efficient and accessible control system. Unfortunately, this has caused huge security risks to be allowed in to the system. Many governmental and critical infrastructure facilities are run by a SCADA backbone, thus the interest in SCADA security is increased in recent years. Thus, the goal of the project is to analyze the security state of the current SCADA setup.

3. Prospective Solution

3.1 Hardware Level Analysis

As with any SCADA systems, the setup in the CS lab is that of many devices each with their own level of security. Therefore, it makes sense to analyze the security implementations of each device in the simple SCADA setup. This will also include nonphysical devices such as the Computer Science Department local network and the Internet itself. While this is a hardware analysis, current running software will also be part of the report.

3.2 Software to Use

Because of the protection provided by the extreme Networks router (see section 4.1), most of the software security analysis is conducted within the local area network. The two computer terminals on this network are: the workstation and the SCADA server.

Fortunately, there are many options for security analysis on Linux/Apache servers. For this project netstat, Metasploit and Wireshark are used. However, because of the inability to log into the Red Hat server with a root account, it is impossible to install software on that machine. Netstat is a command line application that displays network connections for both incoming and outgoing. This also includes routing tables and many other network interface statistics. Netstat is preinstalled on Red Hat, thus this testing is done on the SCADA server. In the next step, Metasploit is used on the Windows XP workstation to pose a threat against the Red Hat SCADA server. The Metasploit project is an open source computer security project that focuses on penetration testing and vulnerabilities scanning [5]. Finally, Wireshark is used to scan the packets being sent from the workstation to the server, and the ones that return. This will tell, if the packets are being encrypted as they are sent or if HTTPS is used. Wireshark is a packet analyzer that is used often for network troubleshooting and security analysis [6].

4. Implementation

4.1 Hardware Level Analysis Results

The SCADA system setup at Florida Gulf Coast University involves using Red Hat and Apache for the webserver. The workstation is running fully patched Windows XP with no Anti-Virus installed. In between the work station and the SCADA server is a Netgear network switch with no hardware firewall. The computers are located on a special Computer Science network that is monitored and controlled with Extreme Networks hardware/software firewall. This provides industrial strength protection from Internet outsiders getting into the SCADA network.

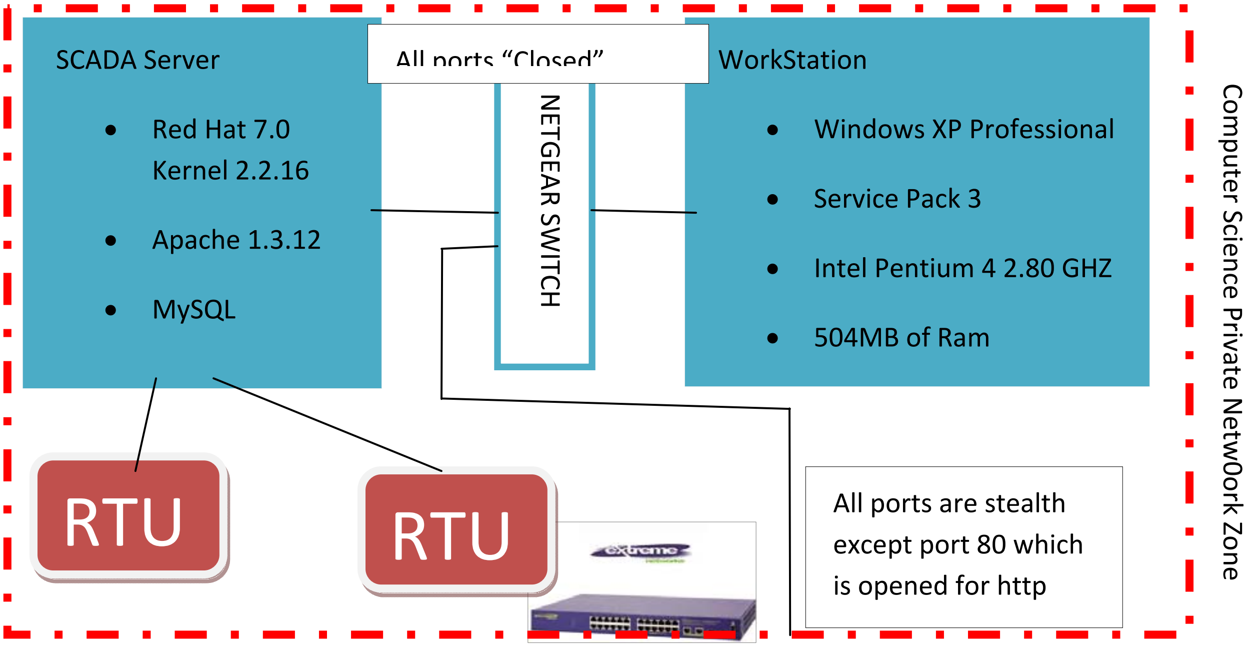


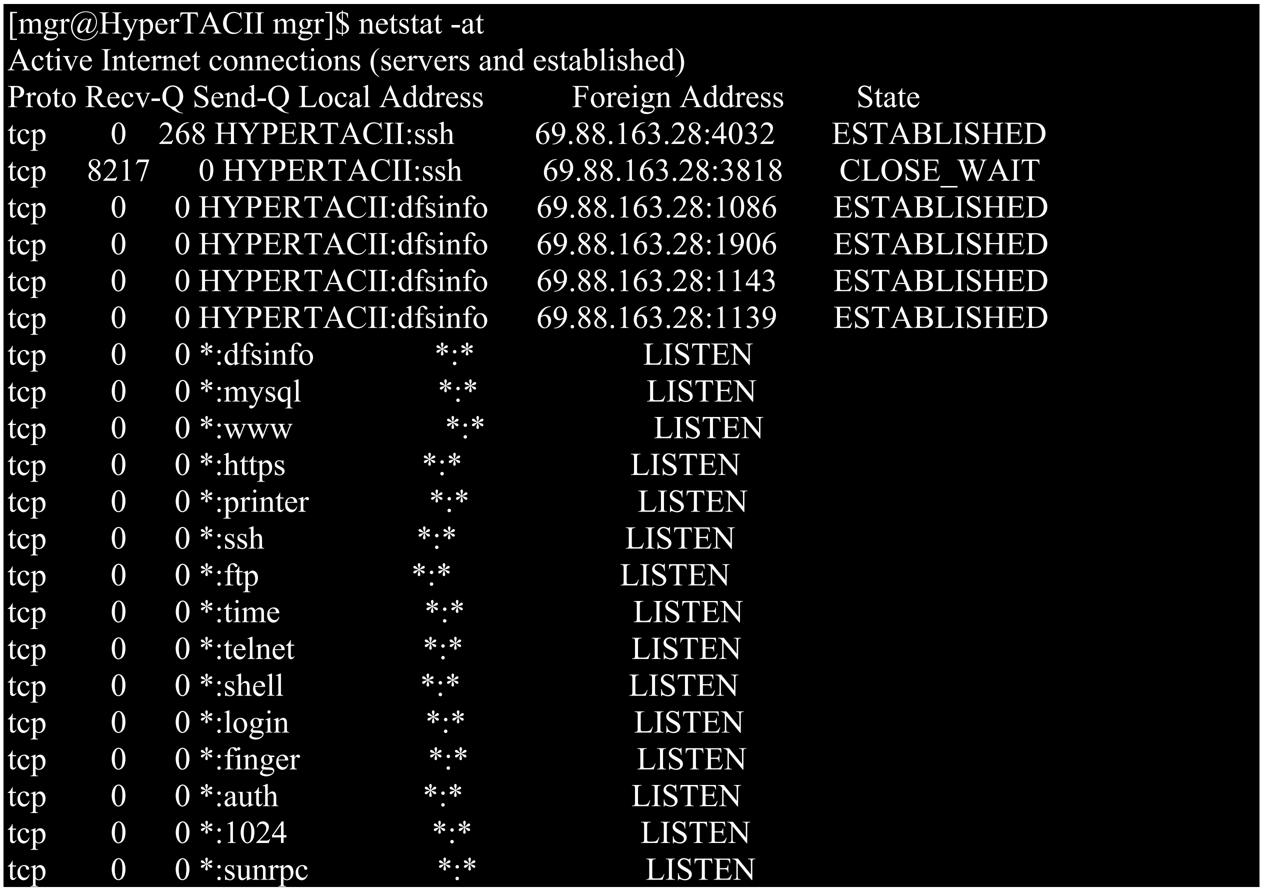
Fig. 6 Network Setup

Figure 6 is a diagram of the current network setup of our sample SCADA system. The two squares boxes represent our computer systems. More details are written in each box. The SCADA server is connected to the RTU's without any kind of security protection. In between the SCADA server and the workstation, there is a Netgear network switch. This switch has all of its networks ports set to "Closed". This means that the ports can be opened if the correct security requirements are met. This allows for SSH and HTTP communication to occur, or whatever else is needed between the SCADA server and the workstation. This could be a security risk if a hacker physical plugs into the switch. He/she could then communicate with the SCADA server and workstation. This is however, a physical layer of security. The Netgear switch also connects to the Extreme Networks switch that “houses” the Computer Science Department network zone. This switch only allows for web traffic using HTTP on port 80. All other ports are set to stealth. This means other ports are not available for user outside of the network. Having only port 80 open proved to be very difficult, because many sites that use HTTPS or provide downloads through FTP were unavailable to me. Sites that used HTTPS or FTP for downloading software use port 443 or 21. This requires transporting software that is needed to install by USB. Nevertheless, the protection provided by the extreme networks router limits most security risks to the local network level. Thus, the test is conducted from the workstation PC and the server itself. Even though, no security system is 100% unhackable, however the protection provided by the extreme networks router is that of an industrial level.

4.2 Software Level Analysis

4.2.1 Netstat Results

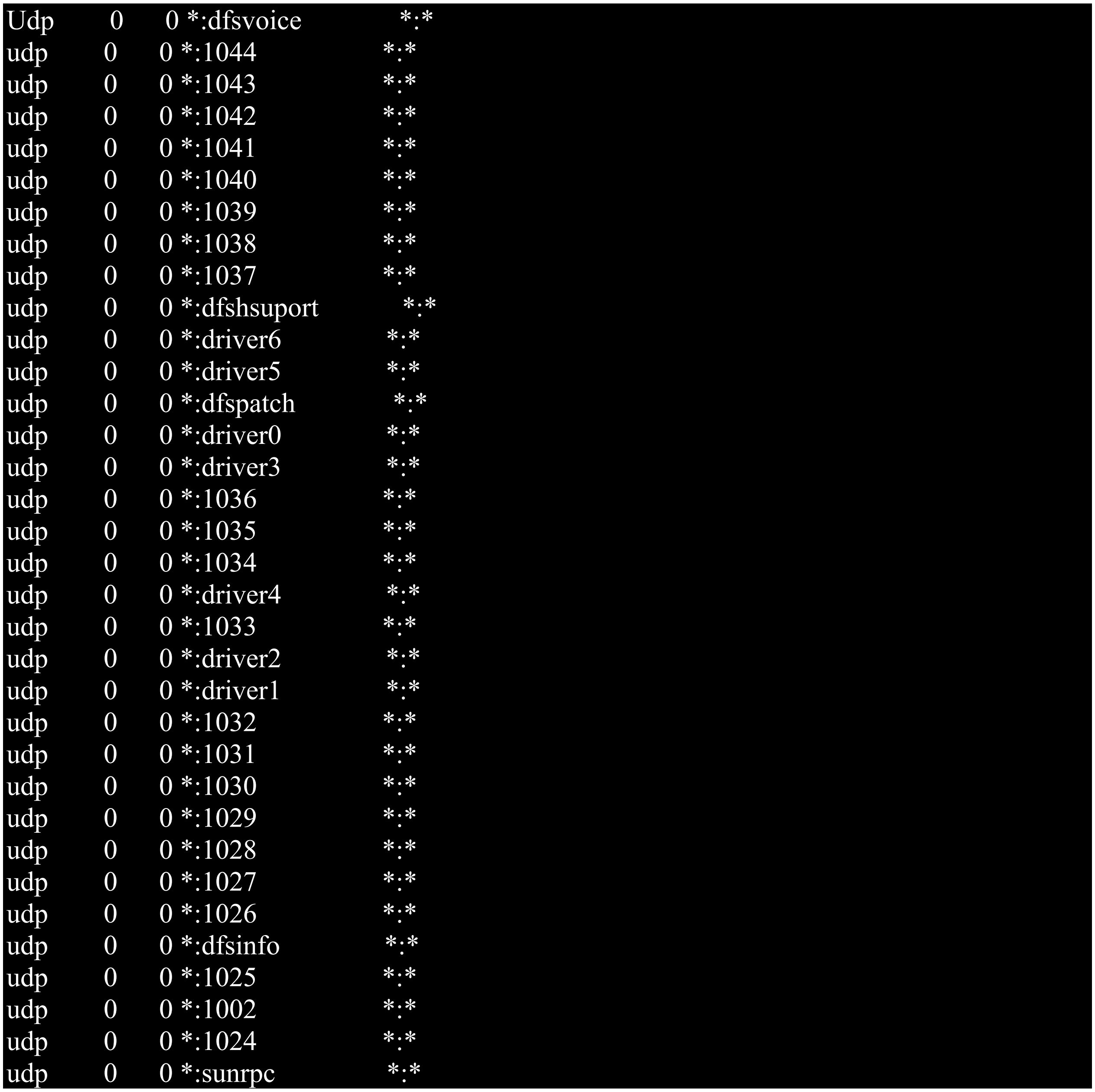
These netstats tests were conducted using an SSH connection via Putty on the Windows XP workstation connected to the Red Hat SCADA. The first test is using “netstat –at”, this displays all of the active network connections going through the primary network device. The results of this test are shown in Figure 7. One can see that there are several connections running when the test was conducted. The first few are connections coming from the Windows XP workstation. The SSH connection are the first two, while the other four are the actual SCADA website hosting. The connections following the first 5 are loopback connections. Loopback means that the ports are listening waiting for an active connection to be created. This is normal for a standard workstation setup, but for a server only required ports needed for the designed operation should be in the listening state. For this setup, that would be SSH and port 80 for HTTP access, all other ports should be closed and not listening.



```
[mgr@HyperTACII mgr]$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp        0    268 HYPERTACII:ssh         69.88.163.28:4032      ESTABLISHED
tcp     8217      0 HYPERTACII:ssh         69.88.163.28:3818      CLOSE_WAIT
tcp        0      0 HYPERTACII:dfsinfo     69.88.163.28:1086      ESTABLISHED
tcp        0      0 HYPERTACII:dfsinfo     69.88.163.28:1906      ESTABLISHED
tcp        0      0 HYPERTACII:dfsinfo     69.88.163.28:1143      ESTABLISHED
tcp        0      0 HYPERTACII:dfsinfo     69.88.163.28:1139      ESTABLISHED
tcp        0      0 *:dfsinfo              *:*                    LISTEN
tcp        0      0 *:mysql                *:*                    LISTEN
tcp        0      0 *:www                  *:*                    LISTEN
tcp        0      0 *:https                *:*                    LISTEN
tcp        0      0 *:printer              *:*                    LISTEN
tcp        0      0 *:ssh                  *:*                    LISTEN
tcp        0      0 *:ftp                  *:*                    LISTEN
tcp        0      0 *:time                 *:*                    LISTEN
tcp        0      0 *:telnet               *:*                    LISTEN
tcp        0      0 *:shell                *:*                    LISTEN
tcp        0      0 *:login                *:*                    LISTEN
tcp        0      0 *:finger               *:*                    LISTEN
tcp        0      0 *:auth                 *:*                    LISTEN
tcp        0      0 *:1024                 *:*                    LISTEN
tcp        0      0 *:sunrpc               *:*                    LISTEN
```

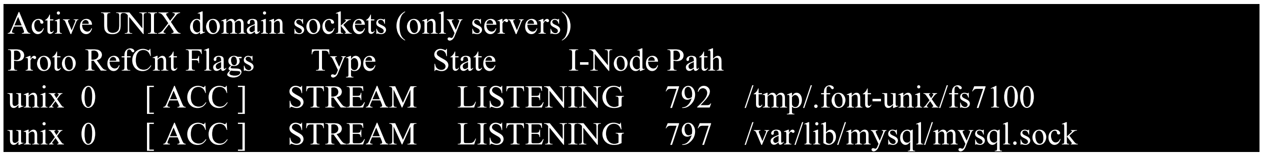
Fig. 7 Active TCP Internet Connections

Next step is to run “netstat –l”. This test displays active network connections on the UDP network communication. For a server station, the number of ports that are listening for a connection should be at a bare minimum, maintaining only necessary connection. The results of this test are shown in Figure 8. Again we are faced with the problem that the server seems to be configured for a workspace environment rather than a locked down server. There are too many networks ports that are in the listening state. The dfs services should be in the listening sate, however there are many that are not required for the intended purpose of the SCADA server.



```
Udp      0      0 *:dfsvoice              *:*
udp     0     0 *:1044                 *:*
udp     0     0 *:1043                 *:*
udp     0     0 *:1042                 *:*
udp     0     0 *:1041                 *:*
udp     0     0 *:1040                 *:*
udp     0     0 *:1039                 *:*
udp     0     0 *:1038                 *:*
udp     0     0 *:1037                 *:*
udp     0     0 *:dfshsuport             *:*
udp     0     0 *:driver6               *:*
udp     0     0 *:driver5               *:*
udp     0     0 *:dfspatch               *:*
udp     0     0 *:driver0               *:*
udp     0     0 *:driver3               *:*
udp     0     0 *:1036                 *:*
udp     0     0 *:1035                 *:*
udp     0     0 *:1034                 *:*
udp     0     0 *:driver4               *:*
udp     0     0 *:1033                 *:*
udp     0     0 *:driver2               *:*
udp     0     0 *:driver1               *:*
udp     0     0 *:1032                 *:*
udp     0     0 *:1031                 *:*
udp     0     0 *:1030                 *:*
udp     0     0 *:1029                 *:*
udp     0     0 *:1028                 *:*
udp     0     0 *:1027                 *:*
udp     0     0 *:1026                 *:*
udp     0     0 *:dfsinfo               *:*
udp     0     0 *:1025                 *:*
udp     0     0 *:1002                 *:*
udp     0     0 *:1024                 *:*
udp     0     0 *:sunrpc                *:*
```

Figure 8 Active UDP Network Connections

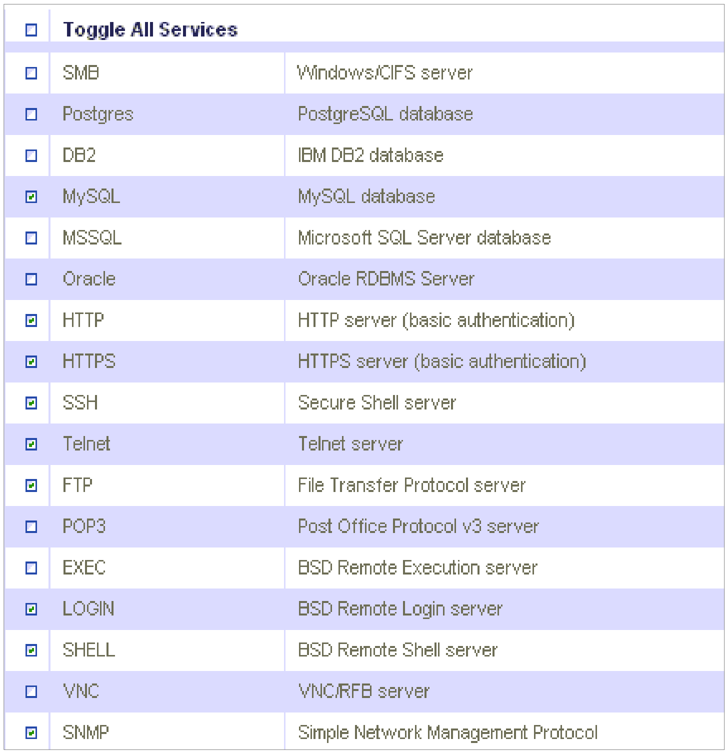


```
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node Path
unix  0      [ ACC ]     STREAM     LISTENING     792    /tmp/.font-unix/fs7100
unix  0      [ ACC ]     STREAM     LISTENING     797    /var/lib/mysql/mysql.sock
```

Figure 9 Active UNIX Domain Sockets

Figure 9 displays the result of “netstat –xl”. This command displays the Active UNIX domain sockets. Active domain sockets are also known as inter-process communication or IPC. This is the data communications endpoint for exchanging data between processes executing within the same host operating system. This analysis proves to give a secure state result. The fs7100 process is required for xserver, which is the graphical interface required for terminal. FS stands for font server. Next, there is a mysql socket. The SCADA webserver uses a MySQL database, so this socket connection is required. No unnecessary socket processes are running.

4.2.2 Metasploit results

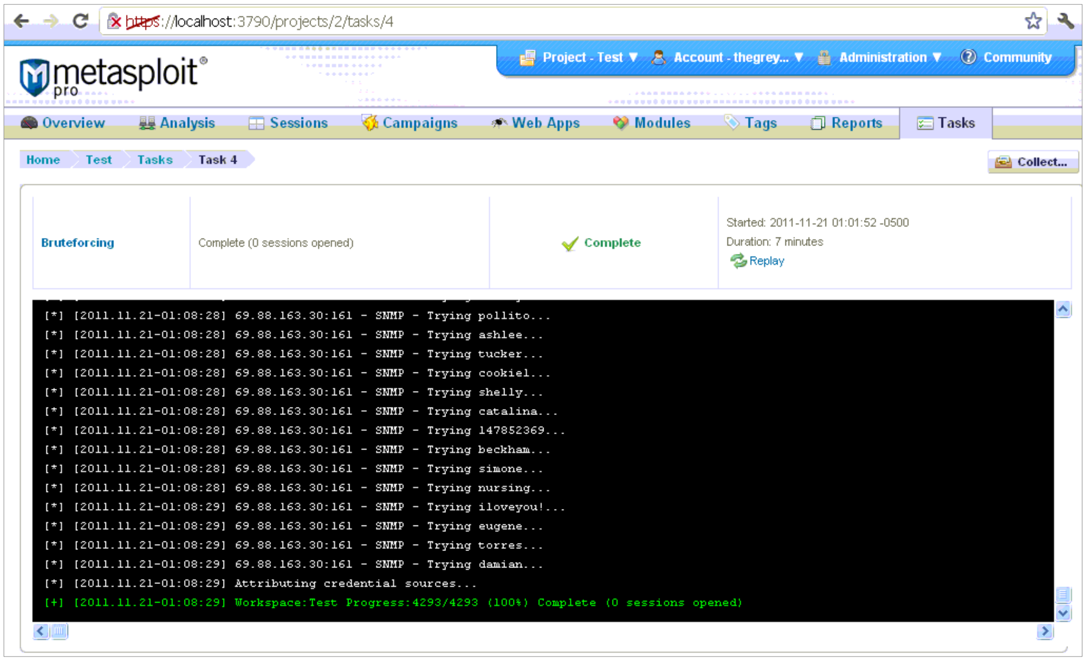


□	**Toggle All Services**	
□	SMB	Windows/CIFS server
□	Postgres	PostgreSQL database
□	DB2	IBM DB2 database
☑	MySQL	MySQL database
□	MSSQL	Microsoft SQL Server database
□	Oracle	Oracle RDBMS Server
☑	HTTP	HTTP server (basic authentication)
☑	HTTPS	HTTPS server (basic authentication)
☑	SSH	Secure Shell server
☑	Telnet	Telnet server
☑	FTP	File Transfer Protocol server
□	POP3	Post Office Protocol v3 server
□	EXEC	BSD Remote Execution server
☑	LOGIN	BSD Remote Login server
☑	SHELL	BSD Remote Shell server
□	VNC	VNC/RFB server
☑	SNMP	Simple Network Management Protocol

Figure 10 Metasploit Bruteforce Configurations

The next step in the SCADA security testing is to use Metasploit. To recap, Metasploit is an open source computer security project that provides extensive analysis on the security of a computer. A penetrability test is a method of evaluating the security of a computer system by simulating an attack from malicious outsiders. The first test with Metasploit is related to the brute-force attack. Brute-force involves systematically checking all possible keys until the correct key is found. Figure 10 shows the configuration that ran with the brute-force attack. All the known processes to be running on the server have been selected, which this includes

MySQL, HTTP, HTTPS, SSH, Telnet, FTP, LOGIN, SHELL, and SNMP. This means, the brute-force attack will attempt to crack the password of all those services. The attack attempts many different/common passwords, trying to login into the individual network services.

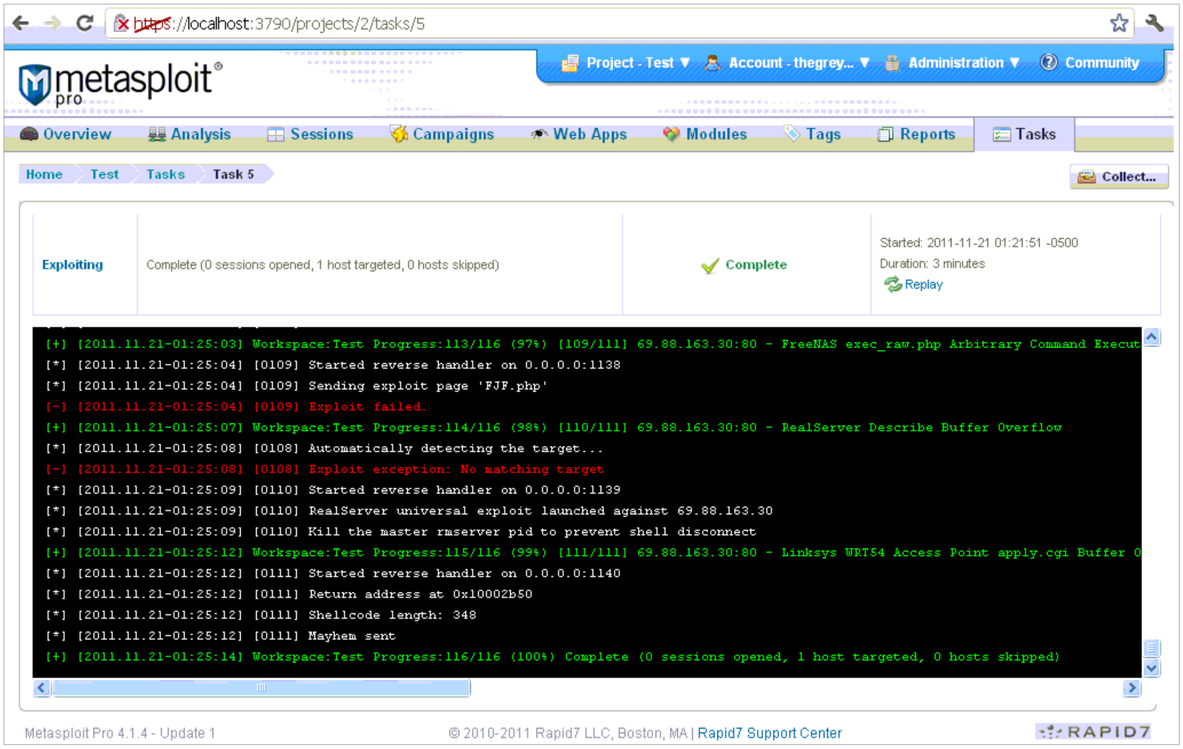


Bruteforcing | Complete (0 sessions opened) | ✓ Complete | Started: 2011-11-21 01:01:52 -0500 Duration: 7 minutes

```
[*] [2011.11.21-01:08:28] 69.88.163.30:161 - SNMP - Trying pollito...
[*] [2011.11.21-01:08:28] 69.88.163.30:161 - SNMP - Trying ashlee...
[*] [2011.11.21-01:08:28] 69.88.163.30:161 - SNMP - Trying tucker...
[*] [2011.11.21-01:08:28] 69.88.163.30:161 - SNMP - Trying cookiel...
[*] [2011.11.21-01:08:28] 69.88.163.30:161 - SNMP - Trying shelly...
[*] [2011.11.21-01:08:28] 69.88.163.30:161 - SNMP - Trying catalina...
[*] [2011.11.21-01:08:28] 69.88.163.30:161 - SNMP - Trying 147852369...
[*] [2011.11.21-01:08:28] 69.88.163.30:161 - SNMP - Trying beckham...
[*] [2011.11.21-01:08:28] 69.88.163.30:161 - SNMP - Trying simone...
[*] [2011.11.21-01:08:28] 69.88.163.30:161 - SNMP - Trying nursing...
[*] [2011.11.21-01:08:29] 69.88.163.30:161 - SNMP - Trying iloveyou!...
[*] [2011.11.21-01:08:29] 69.88.163.30:161 - SNMP - Trying eugene...
[*] [2011.11.21-01:08:29] 69.88.163.30:161 - SNMP - Trying torres...
[*] [2011.11.21-01:08:29] 69.88.163.30:161 - SNMP - Trying damian...
[*] [2011.11.21-01:08:29] Attributing credential sources...
[+] [2011.11.21-01:08:29] Workspace:Test Progress:4293/4293 (100%) Complete (0 sessions opened)
```

Figure 11 Metasploit Brute-Force Results

Figure 11 displays the very end of the Metasploit brute-force test. The entire attack log is very long and would not fit in this report. As one can see in the line at the end of figure 11, “Complete (0 sessions opened)”, this means that none of the dictionary passwords and usernames were able to hack the system. The fake attack took 7 minutes to complete. The test is indeed a result that shows good security levels for the SCADA server. It means that the passwords and usernames used for the various services used are at a very complex level that a brute-force attack could not compromise. However, there are more complex and long lasting brute-force attacks that could eventually compromise the system.



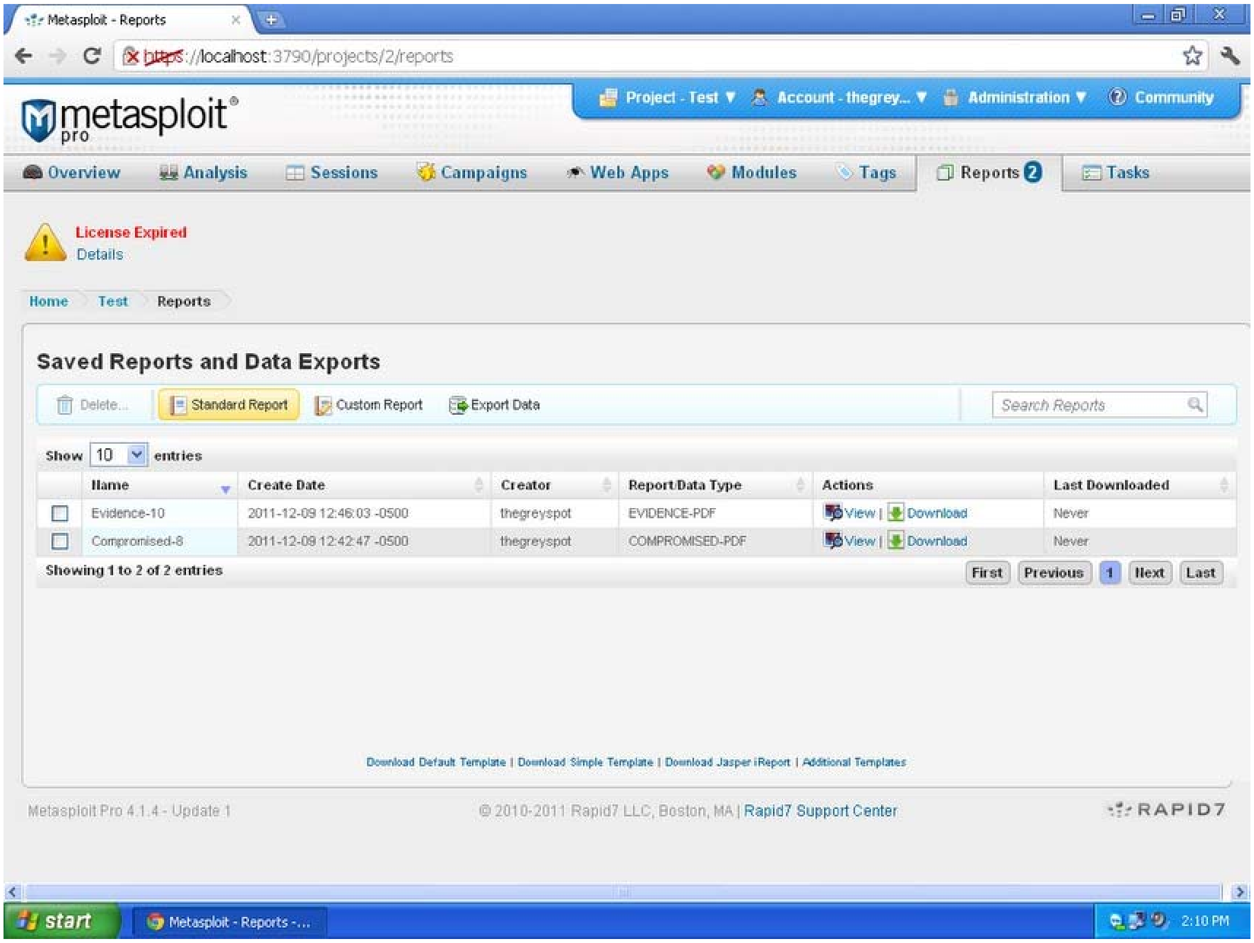
Exploiting | Complete (0 sessions opened, 1 host targeted, 0 hosts skipped) | ✓ Complete | Started: 2011-11-21 01:21:51 -0500 Duration: 3 minutes

```
[+] [2011.11.21-01:25:03] Workspace:Test Progress:113/116 (97%) [109/111] 69.88.163.30:80 - FreeNAS exec_raw.php Arbitrary Command Execut
[*] [2011.11.21-01:25:04] [0109] Started reverse handler on 0.0.0.0:1138
[*] [2011.11.21-01:25:04] [0109] Sending exploit page 'FJF.php'
[-] [2011.11.21-01:25:04] [0109] Exploit failed.
[+] [2011.11.21-01:25:07] Workspace:Test Progress:114/116 (98%) [110/111] 69.88.163.30:80 - RealServer Describe Buffer Overflow
[*] [2011.11.21-01:25:08] [0108] Automatically detecting the target...
[-] [2011.11.21-01:25:08] [0108] Exploit exception: No matching target
[*] [2011.11.21-01:25:09] [0110] Started reverse handler on 0.0.0.0:1139
[*] [2011.11.21-01:25:09] [0110] RealServer universal exploit launched against 69.88.163.30
[*] [2011.11.21-01:25:09] [0110] Kill the master rmserver pid to prevent shell disconnect
[+] [2011.11.21-01:25:12] Workspace:Test Progress:115/116 (99%) [111/111] 69.88.163.30:80 - Linksys WRT54 Access Point apply.cgi Buffer O
[*] [2011.11.21-01:25:12] [0111] Started reverse handler on 0.0.0.0:1140
[*] [2011.11.21-01:25:12] [0111] Return address at 0x10002b50
[*] [2011.11.21-01:25:12] [0111] Shellcode length: 348
[*] [2011.11.21-01:25:12] [0111] Mayhem sent
[+] [2011.11.21-01:25:14] Workspace:Test Progress:116/116 (100%) Complete (0 sessions opened, 1 host targeted, 0 hosts skipped)
```

Figure 12 Metasploit Exploit results

The next step in Metasploit tests was to run the Metasploit Exploit test. This attack is the most common test in testing the vulnerability of a server. An exploit is a piece of software that takes advantage of a bug, glitch or vulnerability in order to cause unanticipated behavior to gain access to a computer system. The exploit took three minutes to run, relatively quick. Again, at the end of the screenshot shown in Figure 12, the attack resulted with 0 opened sessions. This means that the current Red Hat operating system and Apache/MySQL installation are up to date and patched. This is very important for the security of the entire SCADA system.

Metasploit PRO, which is used in the testing, allows for the creation of reports. To create a report, login to the Metasploit frontend as described in the Appendix. In the top right corner of the page there is a “Report” tab, click on that. This will display the page as shown in Figure 13.



License Expired
Details

Saved Reports and Data Exports

Show 10 entries

	Name	Create Date	Creator	Report/Data Type	Actions	Last Downloaded
□	Evidence-10	2011-12-09 12:46:03 -0500	thegreyspot	EVIDENCE-PDF	View | Download	Never
□	Compromised-8	2011-12-09 12:42:47 -0500	thegreyspot	COMPROMISED-PDF	View | Download	Never

Showing 1 to 2 of 2 entries

Download Default Template | Download Simple Template | Download Jasper iReport | Additional Templates

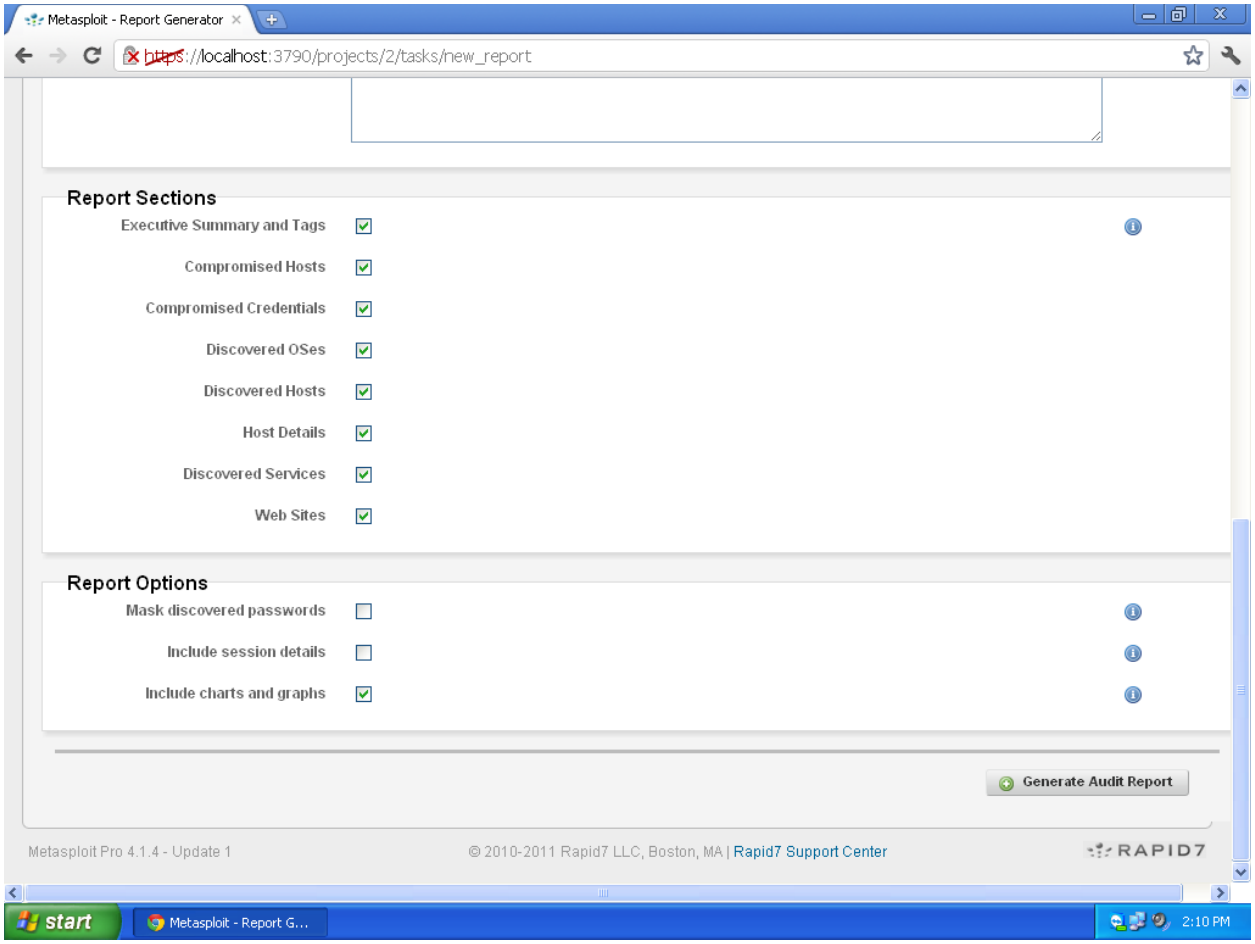
Metasploit Pro 4.1.4 - Update 1

© 2010-2011 Rapid7 LLC, Boston, MA | Rapid7 Support Center

RAPID7

Figure 13 Metasploit Reports Page

Continue by clicking on the “Standard Report” button. This will bring you to a page where you can choose to run different reports to your liking. This is shown in Figure 14. Finally click on “Generate Audit Report”.


Metasploit - Report Generator
https://localhost:3790/projects/2/tasks/new_report
Generate Audit Report
Metasploit Pro 4.1.4 - Update 1
© 2010-2011 Rapid7 LLC, Boston, MA | Rapid7 Support Center
RAPID7
start
Metasploit - Report G...
2:10 PM

Report Sections

Executive Summary and Tags ☑

Compromised Hosts ☑

Compromised Credentials ☑

Discovered OSes ☑

Discovered Hosts ☑

Host Details ☑

Discovered Services ☑

Web Sites ☑

Report Options

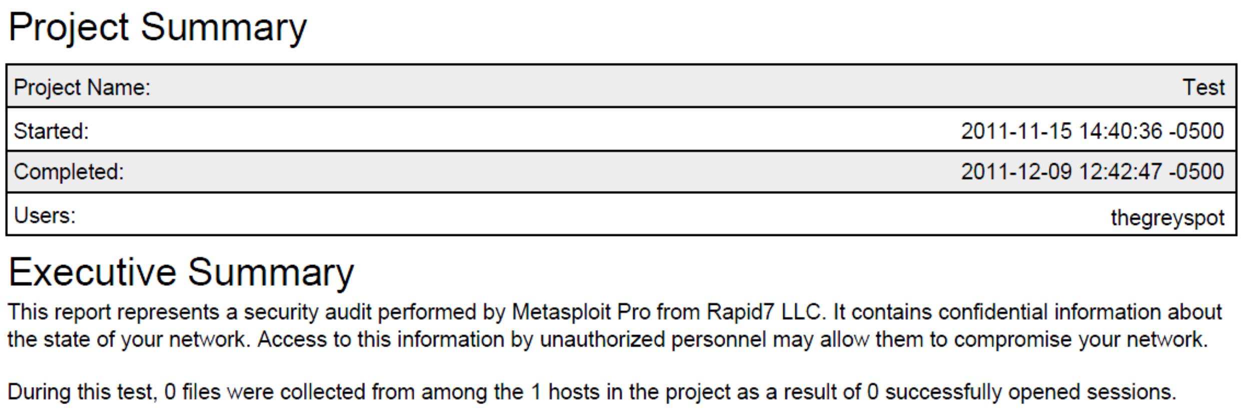
Mask discovered passwords □

Include session details □

Include charts and graphs ☑

Figure 14 Report Generator

After you run the report generator, it will create files in “C:\metasploit\apps\pro\reports”. Figure 15 shows an example report generated by Metasploit.



Project Summary

Project Name:	Test
Started:	2011-11-15 14:40:36 -0500
Completed:	2011-12-09 12:42:47 -0500
Users:	thegreyspot

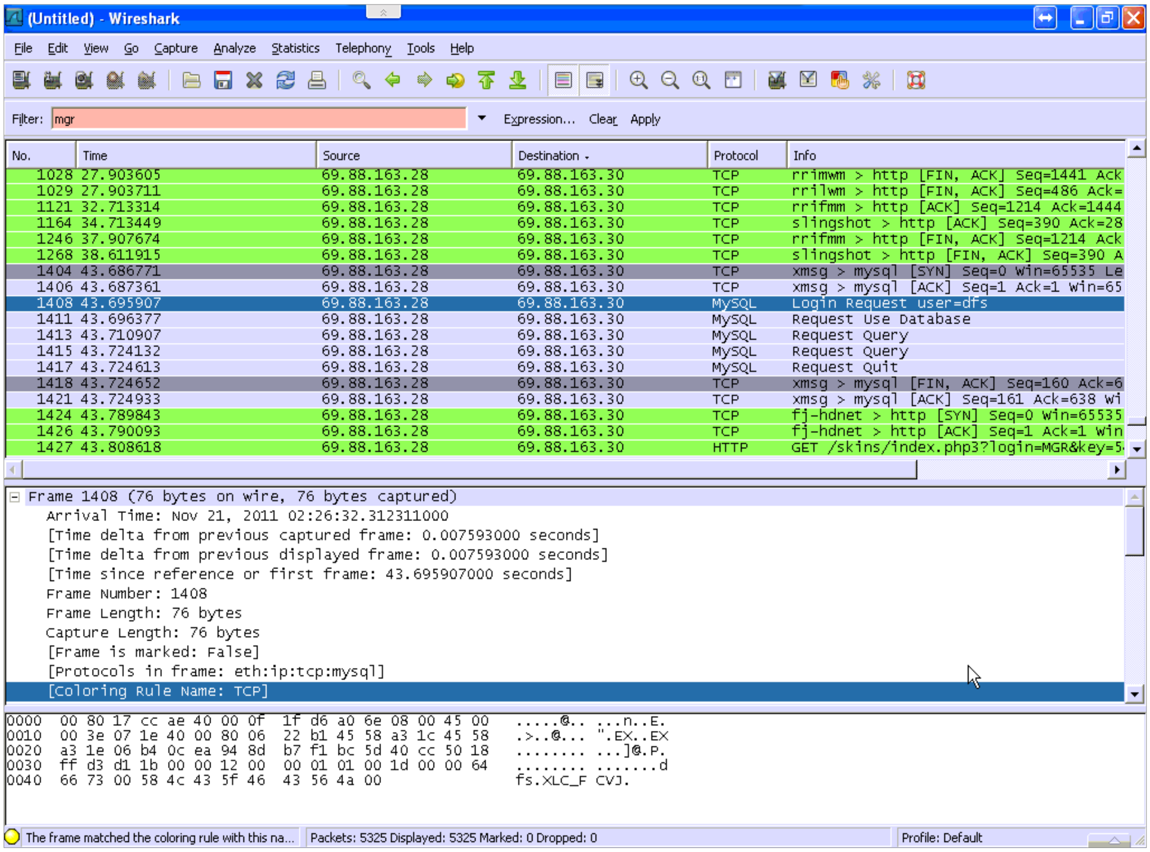
Executive Summary

This report represents a security audit performed by Metasploit Pro from Rapid7 LLC. It contains confidential information about the state of your network. Access to this information by unauthorized personnel may allow them to compromise your network.

During this test, 0 files were collected from among the 1 hosts in the project as a result of 0 successfully opened sessions.

Figure 15 Sample Metasploit Report

4.2.3 Wireshark Results


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Expression... Clear Apply

Filter: mgr

No.	Time	Source	Destination	Protocol	Info
1028	27.903605	69.88.163.28	69.88.163.30	TCP	rrimwm > http [FIN, ACK] Seq=1441 Ack
1029	27.903711	69.88.163.28	69.88.163.30	TCP	rrilwm > http [FIN, ACK] Seq=486 Ack=
1121	32.713314	69.88.163.28	69.88.163.30	TCP	rrifmm > http [ACK] Seq=1214 Ack=1444
1164	34.713449	69.88.163.28	69.88.163.30	TCP	slingshot > http [ACK] Seq=390 Ack=28
1246	37.907674	69.88.163.28	69.88.163.30	TCP	rrifmm > http [FIN, ACK] Seq=1214 Ack
1268	38.611915	69.88.163.28	69.88.163.30	TCP	slingshot > http [FIN, ACK] Seq=390 A
1404	43.686771	69.88.163.28	69.88.163.30	TCP	xmsg > mysql [SYN] Seq=0 Win=65535 Le
1406	43.687361	69.88.163.28	69.88.163.30	TCP	xmsg > mysql [ACK] Seq=1 Ack=1 Win=65
1408	43.695907	69.88.163.28	69.88.163.30	MySQL	Login Request user=dfs
1411	43.696377	69.88.163.28	69.88.163.30	MySQL	Request Use Database
1413	43.710907	69.88.163.28	69.88.163.30	MySQL	Request Query
1415	43.724132	69.88.163.28	69.88.163.30	MySQL	Request Query
1417	43.724613	69.88.163.28	69.88.163.30	MySQL	Request Quit
1418	43.724652	69.88.163.28	69.88.163.30	TCP	xmsg > mysql [FIN, ACK] Seq=160 Ack=6
1421	43.724933	69.88.163.28	69.88.163.30	TCP	xmsg > mysql [ACK] Seq=161 Ack=638 Wi
1424	43.789843	69.88.163.28	69.88.163.30	TCP	fj-hdnet > http [SYN] Seq=0 Win=65535
1426	43.790093	69.88.163.28	69.88.163.30	TCP	fj-hdnet > http [ACK] Seq=1 Ack=1 Win
1427	43.808618	69.88.163.28	69.88.163.30	HTTP	GET /skins/index.php3?login=MGR&key=5

```
⊟ Frame 1408 (76 bytes on wire, 76 bytes captured)
    Arrival Time: Nov 21, 2011 02:26:32.312311000
    [Time delta from previous captured frame: 0.007593000 seconds]
    [Time delta from previous displayed frame: 0.007593000 seconds]
    [Time since reference or first frame: 43.695907000 seconds]
    Frame Number: 1408
    Frame Length: 76 bytes
    Capture Length: 76 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:ip:tcp:mysql]
    [Coloring Rule Name: TCP]
```

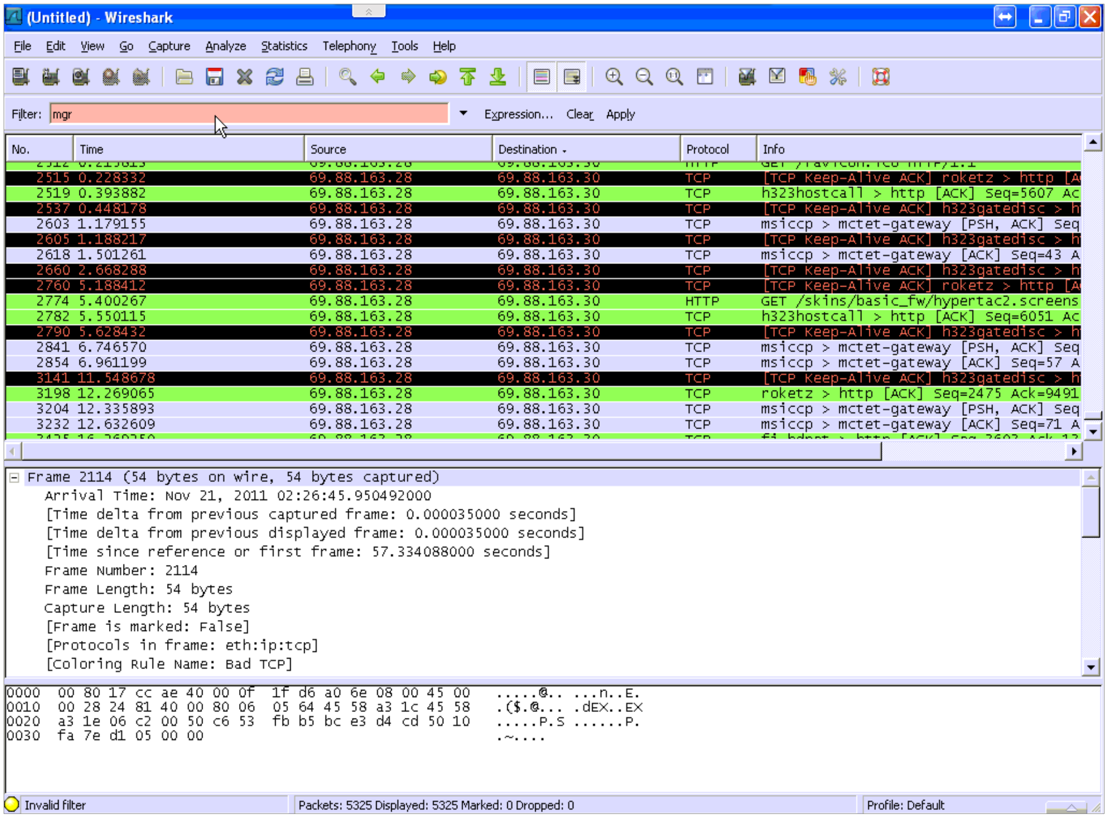
```
0000  00 80 17 cc ae 40 00 0f  1f d6 a0 6e 08 00 45 00   .....@.. ...n..E.
0010  00 3e 07 1e 40 00 80 06  22 b1 45 58 a3 1c 45 58   .>..@... ".EX..EX
0020  a3 1e 06 b4 0c ea 94 8d  b7 f1 bc 5d 40 cc 50 18   ........ ...]@.P.
0030  ff d3 d1 1b 00 00 12 00  00 01 01 00 1d 00 00 64   ........ .......d
0040  66 73 00 58 4c 43 5f 46  43 56 4a 00               fs.XLC_F CVJ.
```

The frame matched the coloring rule with this na... | Packets: 5325 Displayed: 5325 Marked: 0 Dropped: 0 | Profile: Default

Figure 15 Wireshark Login Packet

In this final security analysis test, common open-source packet analyzer software called Wireshark was used. The test was conducted by starting a packet capturing session in Wireshark. Then I opened up a browser from the workstation and navigated to the SCADA server front end, where I was prompted to log in, and provided my username (dfs) and password. Then I continued to click on the view tab of the frontend. After seeing a successful view of the current SCADA summary, I terminated the capturing session in Wireshark.

During this session, 5325 packets were captured. Most of these packets have nothing to do with our SCADA security testing, however they can be filtered out by looking for the server's IP address (69.88.163.30). In figure 13, one can see the login attempt packet. The packet displays the username "dfs", however successfully encrypts the password part of the data packet. This is extremely important, because a third party hacker could be sitting on the network and be able to get the password. But if he was able to capture this packet, he would need to unencrypt the password.



Filter: mgr

No.	Time	Source	Destination	Protocol	Info
2515	0.228332	69.88.163.28	69.88.163.30	TCP	[TCP Keep-Alive ACK] roketz > http [A
2519	0.393882	69.88.163.28	69.88.163.30	TCP	h323hostcall > http [ACK] Seq=5607 Ac
2537	0.448178	69.88.163.28	69.88.163.30	TCP	[TCP Keep-Alive ACK] h323gatedisc > h
2603	1.179155	69.88.163.28	69.88.163.30	TCP	msiccp > mctet-gateway [PSH, ACK] Seq
2605	1.188217	69.88.163.28	69.88.163.30	TCP	[TCP Keep-Alive ACK] h323gatedisc > h
2618	1.501261	69.88.163.28	69.88.163.30	TCP	msiccp > mctet-gateway [ACK] Seq=43 A
2660	2.668288	69.88.163.28	69.88.163.30	TCP	[TCP Keep-Alive ACK] h323gatedisc > h
2760	5.188412	69.88.163.28	69.88.163.30	TCP	[TCP Keep-Alive ACK] roketz > http [A
2774	5.400267	69.88.163.28	69.88.163.30	HTTP	GET /skins/basic_fw/hypertac2.screens
2782	5.550115	69.88.163.28	69.88.163.30	TCP	h323hostcall > http [ACK] Seq=6051 Ac
2790	5.628432	69.88.163.28	69.88.163.30	TCP	[TCP Keep-Alive ACK] h323gatedisc > h
2841	6.746570	69.88.163.28	69.88.163.30	TCP	msiccp > mctet-gateway [PSH, ACK] Seq
2854	6.961199	69.88.163.28	69.88.163.30	TCP	msiccp > mctet-gateway [ACK] Seq=57 A
3141	11.548678	69.88.163.28	69.88.163.30	TCP	[TCP Keep-Alive ACK] h323gatedisc > h
3198	12.269065	69.88.163.28	69.88.163.30	TCP	roketz > http [ACK] Seq=2475 Ack=9491
3204	12.335893	69.88.163.28	69.88.163.30	TCP	msiccp > mctet-gateway [PSH, ACK] Seq
3232	12.632609	69.88.163.28	69.88.163.30	TCP	msiccp > mctet-gateway [ACK] Seq=71 A

```
⊟ Frame 2114 (54 bytes on wire, 54 bytes captured)
    Arrival Time: Nov 21, 2011 02:26:45.950492000
    [Time delta from previous captured frame: 0.000035000 seconds]
    [Time delta from previous displayed frame: 0.000035000 seconds]
    [Time since reference or first frame: 57.334088000 seconds]
    Frame Number: 2114
    Frame Length: 54 bytes
    Capture Length: 54 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:ip:tcp]
    [Coloring Rule Name: Bad TCP]
```

```
0000  00 80 17 cc ae 40 00 0f  1f d6 a0 6e 08 00 45 00   .....@.. ...n..E.
0010  00 28 24 81 40 00 80 06  05 64 45 58 a3 1c 45 58   .($.@... .dEX..EX
0020  a3 1e 06 c2 00 50 c6 53  fb b5 bc e3 d4 cd 50 10   .....P.S ......P.
0030  fa 7e d1 05 00 00                                  .~....
```

Invalid filter | Packets: 5325 Displayed: 5325 Marked: 0 Dropped: 0

Figure 16 Broken Packets caught by Wireshark

Figure 14 shows that there are multiple packets being highlighted by a black highlighting and red font. Looking into the Wireshark color coding rules, one can see that this type of highlighting is reported as a “bad TCP” packet. With TCP, each packet is sent with a checksum. When the packet arrives on the server end, the checksum is checked, however with these packets, the checksums are not met. This does not necessary report a security threat. Nevertheless, having a constant stream of incomplete packets may be a potential vulnerability that could be exploited. Further investigation into why so many packets are sending with bad checksum would be recommended.

5. Conclusion

Due to the industrial strength provide by FGCU Network Services on the Computer Science local area network, this SCADA setup is very secure from outside attacks. In order for a hacker to even touch the SCADA setup, he/she will need to be physically attached to one of the devices in the room.

With the further testing of software security using Wireshark, Metasploit and netstat, we conclude that our SCADA server is fairly secure. Many of the tests reported in Section 4, show that the SCADA server running Red Hat is secure and patched with the latest updates. One slight problem is the fact that unnecessary ports are being left on the listening state.

For the future of the project, it would be important to gain Root access to the Red Hat server. This will allow us to install the client version of Nexpose, which is an extensive vulnerabilities scanner.

It is also highly recommended that the workstation computer is replaced, as it has multiple software issues. These issues severely delayed the success of the project.

6. References

[1] National Communications System. “Technical Bulletin 04□1: Supervisory Control and Data Acquisition (SCADA) Systems”. Published by Communication Technologies, Inc. 14151 Newbrook Drive, Suite 400 Chantilly, VA 20151. Published October 2004. http://www.ncs.gov/library/tech_bulletins/2004/tib_04□1.pdf

[2] Bennett, T. “Security in SCADA Applications”. Florida Gulf Coast University. Spring 2010. http://itech.fgcu.edu/faculty/zalewski/CNT4104/Projects/SCADA_Report6.pdf

[3] Krutz, R L. “Securing SCADA Systems”. Published by Wiley Publishing, Inc. 10475 Crosspoint Boulevard, Indianapolis, IN. Published 2009

[4] Baumgarter, F. “netstat(8) – Linux man page”. Accessed December 2011 http://linux.die.net/man/8/netstat

[5] Rapid7. “About Metasploit”. Accessed December 2011 http://metasploit.com/about/

[6] Sharpe, R. “Wireshark User’s Guide”. Published 2004 http://www.wireshark.org/docs/wsug_html_chunked/

Appendix User Manual

To use the SCADA system and the software tools properly instructions in the Appendix should be followed.

1. Hardware Setup

1. Please note the power supply unit should never be turned off. Thus it is implied in this user manual that the power supply is on.
2. Begin by turning on the SCADA server. This is done by pressing the up arrow, as shown in Figure 1.

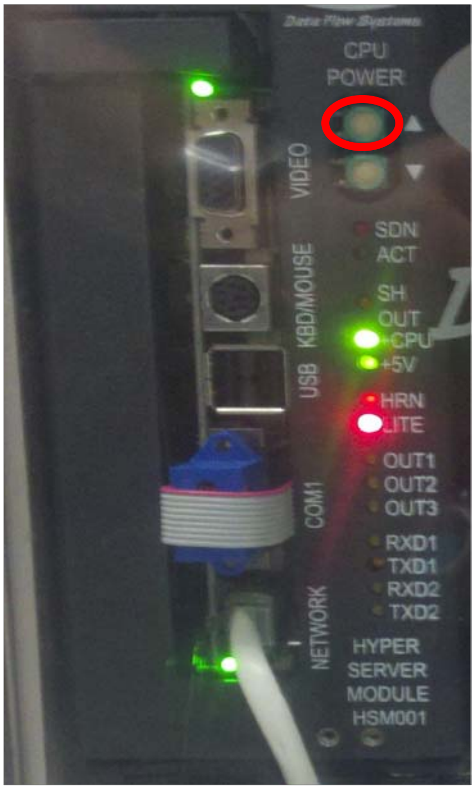


Figure 1 SCADA Server Power Up Button

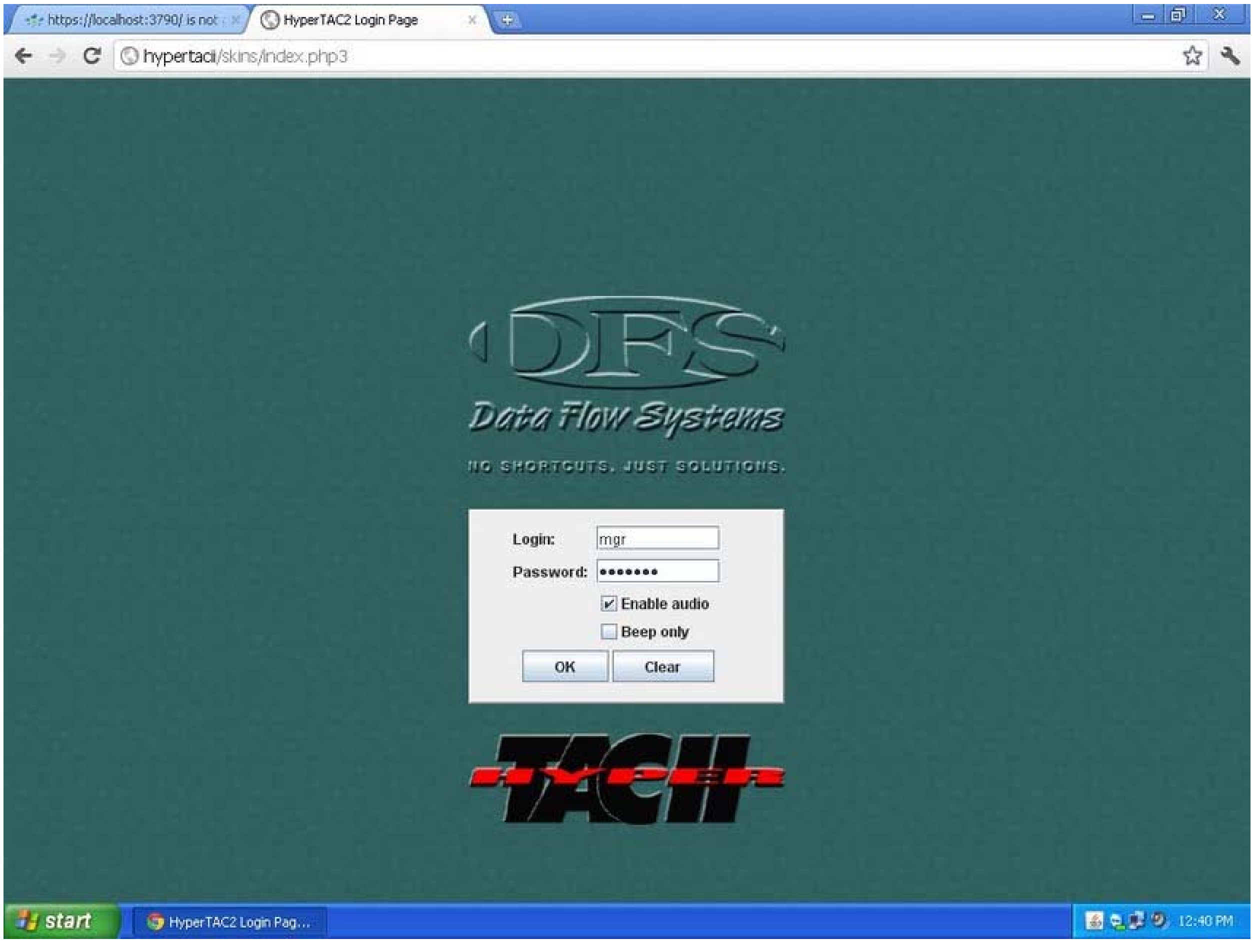
3. Wait a few minutes so that the server has enough time to turn on.

2 Software Setup

1. Turn on and log into the workstation computer using the username "Operator" with no password.

2. Once the computer logs in completely, open up firefox which can easily be found on the desktop.

3. In firefox navigate to: hypertacii



hypertacii/skins/index.php3

Login: mgr

Password: •••••••

☑ Enable audio

☐ Beep only

Figure 2 Hypertacii Frontend Login

4. Once the page loads, you will be prompted for a username and password. You can use: mgr for the username and htiimgr for the password.

5. You have successfully accessed the frontend web interface of our SCADA setup.

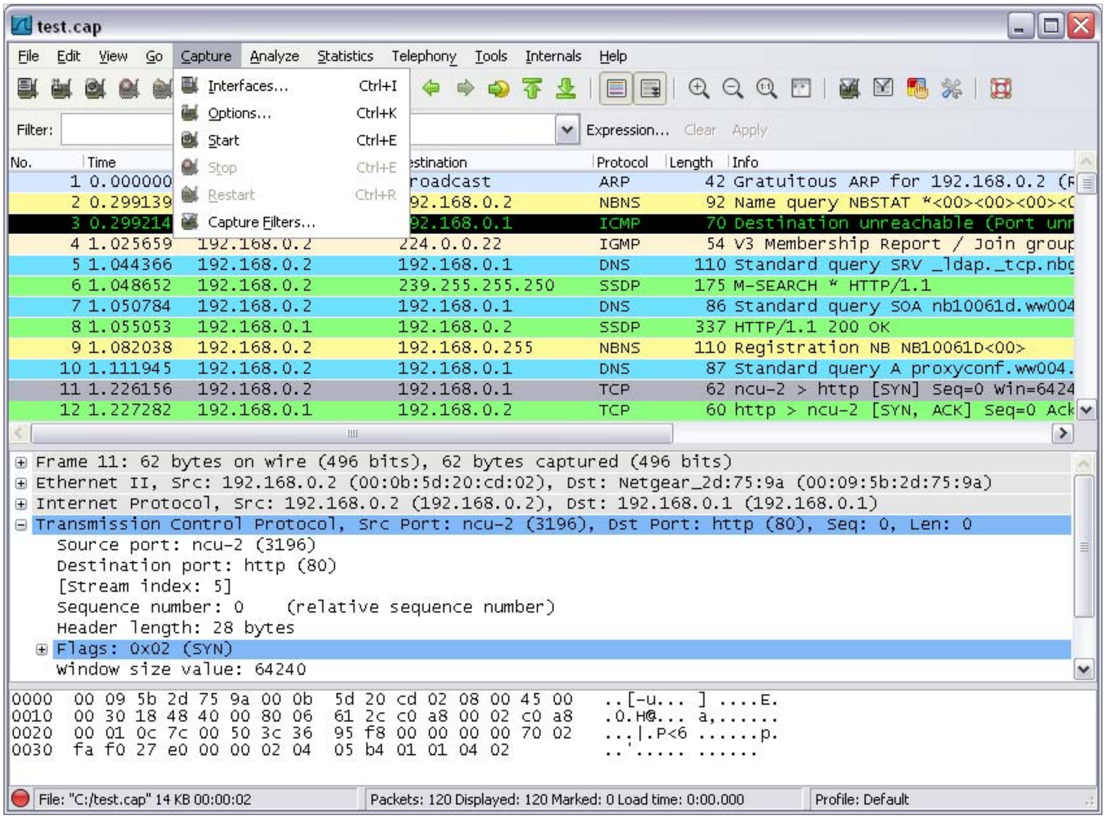
3 Netstat

Netstat is only available on the Red Hat SCADA server. Red Hat comes preinstalled with netstat, thus there is no need for installation. You must plug in a monitor into the SCADA server. To the right of the power buttons is a VGA port. Please see Figure 16 for a reference. Next you will need to find a PS2 keyboard. There are USB ports on the server, however the Red Hat installation does not have USB drivers installed. Once you have your monitor and keyboard installed and ready to go, log into the console using mgr and htiimgr. Once logged in, reference section 4.2.1 for netstat commands and usage.

4 Wireshark

1. Once logged into the workstation computer (see section 5.2), open “Wireshark” from the desktop.

2. You will then want to create a connection to the SCADA server by visiting the frontend interface (see section 5.2).



```
No.  Time                   estination         Protocol Length Info
  1 0.000000               roadcast           ARP       42 Gratuitous ARP for 192.168.0.2 (R
  2 0.299139               92.168.0.2         NBNS      92 Name query NBSTAT *<00><00><00><0
  3 0.299214               92.168.0.1         ICMP      70 Destination unreachable (Port unr
  4 1.025659 192.168.0.2   224.0.0.22         IGMP      54 V3 Membership Report / Join group
  5 1.044366 192.168.0.2   192.168.0.1        DNS      110 Standard query SRV _ldap._tcp.nbg
  6 1.048652 192.168.0.2   239.255.255.250    SSDP     175 M-SEARCH * HTTP/1.1
  7 1.050784 192.168.0.2   192.168.0.1        DNS       86 Standard query SOA nb10061d.ww004
  8 1.055053 192.168.0.1   192.168.0.2        SSDP     337 HTTP/1.1 200 OK
  9 1.082038 192.168.0.2   192.168.0.255      NBNS     110 Registration NB NB10061D<00>
 10 1.111945 192.168.0.2   192.168.0.1        DNS       87 Standard query A proxyconf.ww004.
 11 1.226156 192.168.0.2   192.168.0.1        TCP       62 ncu-2 > http [SYN] Seq=0 Win=6424
 12 1.227282 192.168.0.1   192.168.0.2        TCP       60 http > ncu-2 [SYN, ACK] Seq=0 Ack
```

```
⊞ Frame 11: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
⊞ Ethernet II, Src: 192.168.0.2 (00:0b:5d:20:cd:02), Dst: Netgear_2d:75:9a (00:09:5b:2d:75:9a)
⊞ Internet Protocol, Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1)
⊟ Transmission Control Protocol, Src Port: ncu-2 (3196), Dst Port: http (80), Seq: 0, Len: 0
    Source port: ncu-2 (3196)
    Destination port: http (80)
    [Stream index: 5]
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊞ Flags: 0x02 (SYN)
    Window size value: 64240
```

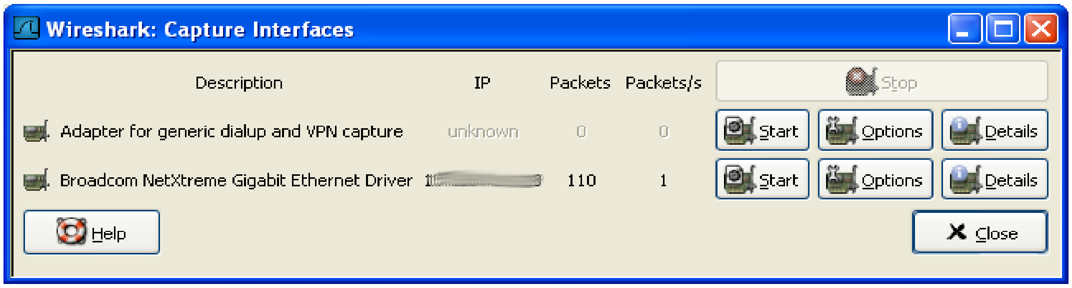
```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 30 18 48 40 00 80 06  61 2c c0 a8 00 02 c0 a8   .0.H@... a,......
0020  00 01 0c 7c 00 50 3c 36  95 f8 00 00 00 00 70 02   ...|.P<6 ......p.
0030  fa f0 27 e0 00 00 02 04  05 b4 01 01 04 02         ..'..... ......
```

File: "C:/test.cap" 14 KB 00:00:02 | Packets: 120 Displayed: 120 Marked: 0 Load time: 0:00.000 | Profile: Default

Figure 3 Wireshark Capture Screenshot

3. In Wireshark, navigate to the capture menu and choose “Interfaces”. This is shown in Figure 3.

4. This will bring up the Capture Interfaces window as shown in Figure 4. Choose the network device labeled “Broadcom NetXtreme Gigabit Ethernet Driver”.



Description	IP	Packets	Packets/s
Adapter for generic dialup and VPN capture	unknown	0	0
Broadcom NetXtreme Gigabit Ethernet Driver	[redacted]	110	1

Figure 4 Wireshark Capture Interfaces Dialog Box

5. Let this run for a few seconds and log into the Frontend interface. Once you think you have all the captured packets, you can stop the scan.

5 Metasploit

Metasploit is already installed on the workstation computer. It also use a webbased frontend that can be viewed by using any browser on. This section will explain how to scan for vulnerabilities and run a brute-force attack.

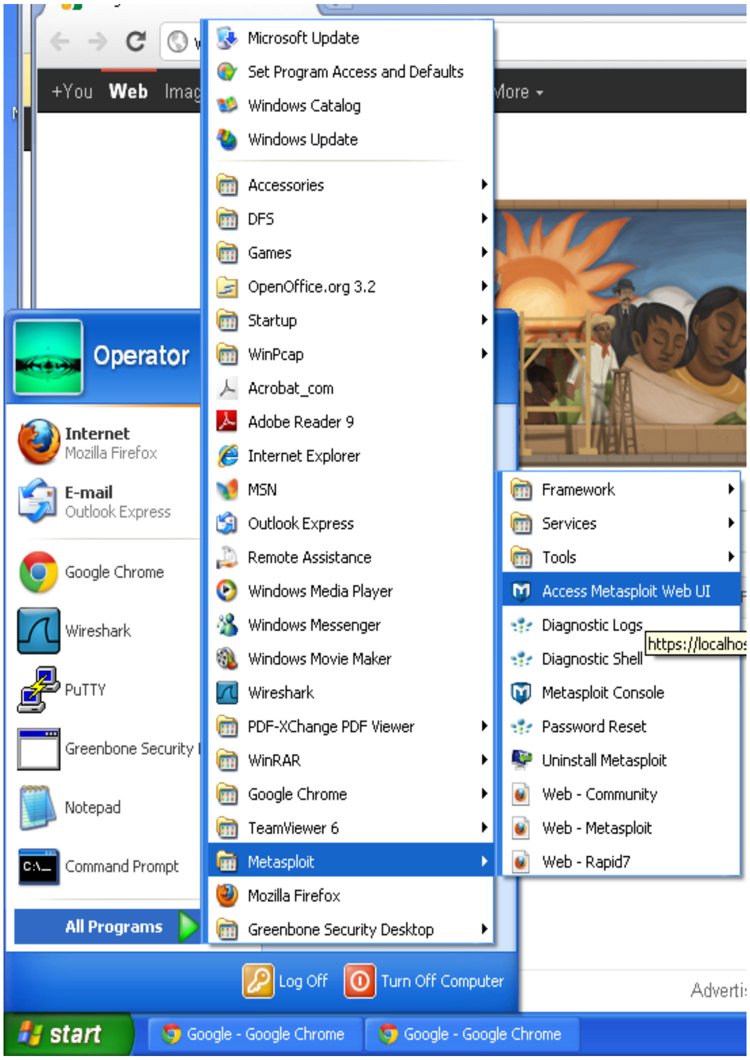
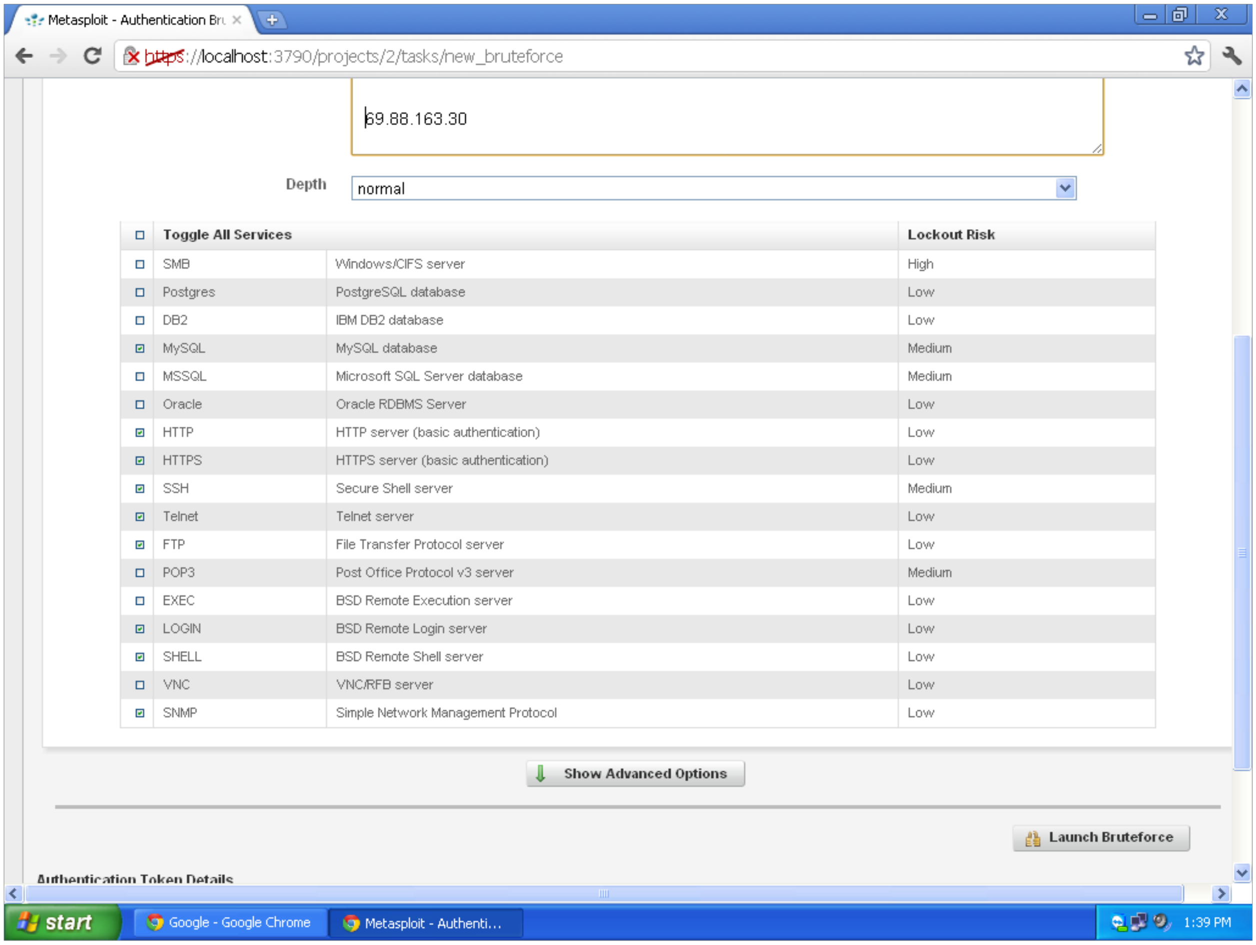


Figure 5 Metasploit in Start menu

1. Click on start on “Start” → “All Programs” → “Metasploit” → “Access Metasploit Web UI”. As shown in Figure 5.

2. This will open your default browser and bring you to "http://localhost:3790". You maybe prompted with a security certificate issue, just click accept and continue.
3. Then login using zalewski and zalewski1!
4. On the projects page choose "Test"
5. To do a brute-force attach choose "Bruteforce".


Metasploit - Authentication Bru
~~https~~://localhost:3790/projects/2/tasks/new_bruteforce
Show Advanced Options
Launch Bruteforce
Authentication Token Details
start
Google - Google Chrome
Metasploit - Authenti...
1:39 PM

69.88.163.30

Depth: normal

□	Toggle All Services		Lockout Risk
□	SMB	Windows/CIFS server	High
□	Postgres	PostgreSQL database	Low
□	DB2	IBM DB2 database	Low
☑	MySQL	MySQL database	Medium
□	MSSQL	Microsoft SQL Server database	Medium
□	Oracle	Oracle RDBMS Server	Low
☑	HTTP	HTTP server (basic authentication)	Low
☑	HTTPS	HTTPS server (basic authentication)	Low
☑	SSH	Secure Shell server	Medium
☑	Telnet	Telnet server	Low
☑	FTP	File Transfer Protocol server	Low
□	POP3	Post Office Protocol v3 server	Medium
□	EXEC	BSD Remote Execution server	Low
☑	LOGIN	BSD Remote Login server	Low
☑	SHELL	BSD Remote Shell server	Low
□	VNC	VNC/RFB server	Low
☑	SNMP	Simple Network Management Protocol	Low

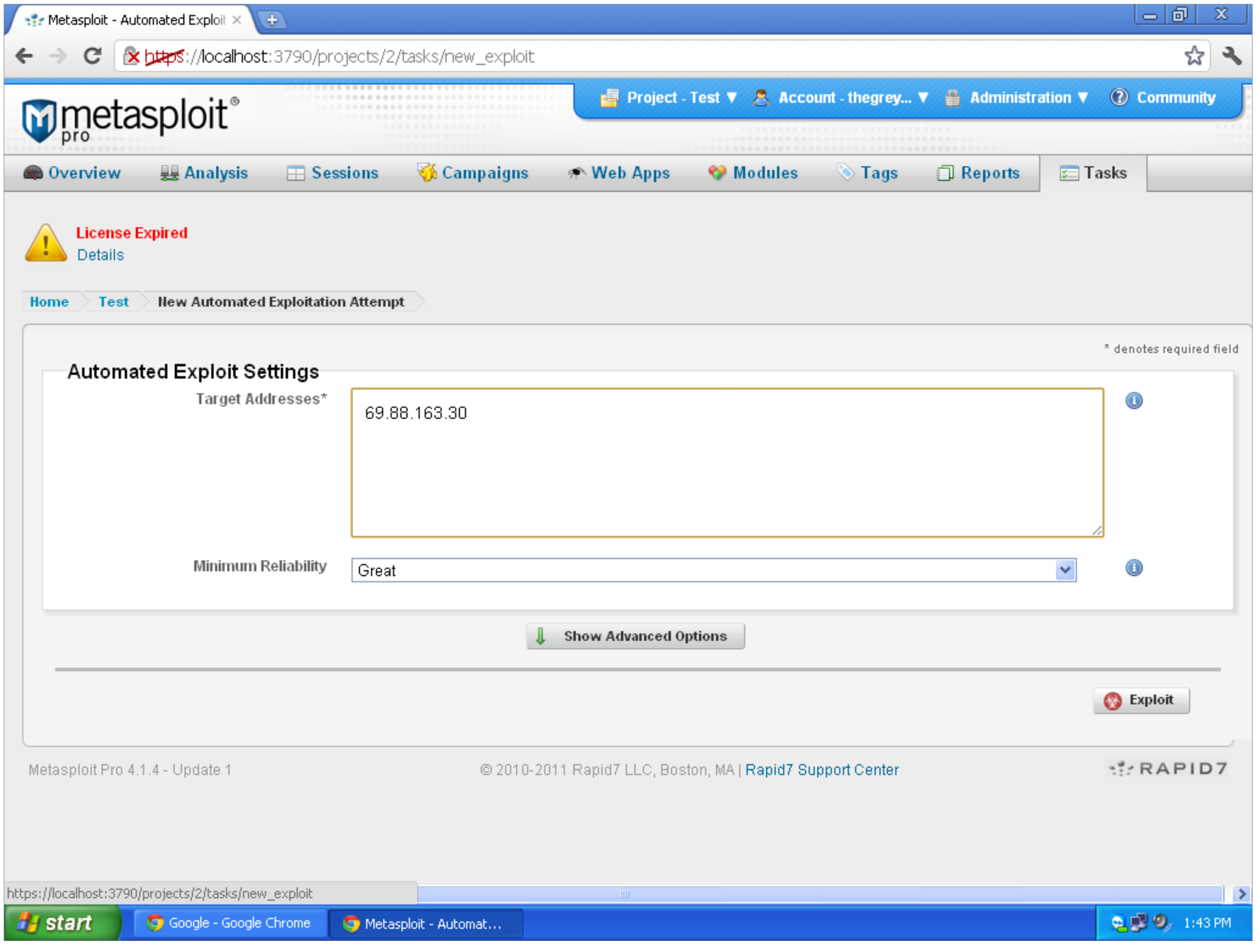
Figure 6 Metasploit Brute-Force Configuration

6. Figure 6 shows the brute-force page. You must type in the IP address of the SCADA server which is 69.88.163.30. You can also choose the different services to attack.

7. Finally click “Launch Bruteforce”

8. Next you can run a Exploit test. You can do this by choosing the “Exploit” option on the project Test dashboard page.

9. On the “Automated Exploit” page, you will also have to type in 69.88.163.30 as your IP address to the SCADA server. See Figure 7 for other configuration settings.



Automated Exploit Settings

Target Addresses* 69.88.163.30

Minimum Reliability Great

Figure 7 Metasploit Exploit Scanner