SCADA Honeynets
How to Build and Analyzing Attacks

Landon Lewis, Dale Peterson
Digital Bond, Inc.
1580 Sawgrass Corporate Parkway, Suite 130
Sunrise, FL 33323
lewis@digitalbond.com, peterson@digitalbond.com

Abstract: Honeynets are a useful research tool to better understand attacks and attackers, and a useful early attack warning tool for network owners. In this paper we detail the design and implementation of a SCADA Honeynet that appears to an attacker to be a popular PLC. The SCADA Honeynet described in this paper relies on Open Source software and VMware server images so there is no software cost and it is easy to deploy. The paper concludes with a description of two different SCADA Honeynet deployment scenarios and an analysis of attack data observed in those scenarios.

Keywords: Calculating Risk, Honeynet, PLC, SCADA

1 Introduction

Honeynets [1, 2] are security tools that can be used in research projects to learn about the threat to a system or network or as an early warning device for cyber attacks on production systems. The Honeynet Project has led the way in developing Honeynet technology, but to date the Honeynet targets have been limited to very common operating systems and applications. There was a previous SCADA Honeynet effort by Franz and Pothamsetty [3] that showed the possibilities for a SCADA Honeynet but did not leverage existing Open Source software for the SCADA services.

In this paper, we discuss a Digital Bond research project[1] that leveraged existing Honeynet technology and Open Source SCADA tools to create and monitor SCADA Honeynets.

The goals of the design portion of the project were:

- To add SCADA intelligence to existing Honeynet solutions
- To design a SCADA Honeynet that realistically simulates a PLC
- To design a SCADA Honeynet that requires no payment for software licenses for operating systems, simulators or services
- To design a SCADA Honeynet that is easy to deploy and does not require detailed knowledge of Honeynets or PLC's

[1] Thanks to NISCC, http://www.niscc.gov.uk, for funding Phase I of Digital Bond's SCADA Honeynet Research Project

All of these goals were achieved in the project and will be described in Sections 2 and 3. In addition to the SCADA Honeynet design and implementation, we also deployed SCADA Honeynets in two different scenarios. The scenarios and the results from months of monitored SCADA Honeynet is provided in Section 4.

2 SCADA Honeynet Architecture

The goals of the SCADA Honeynet dictated the selection of Linux as the Operating System and other Open Source services discussed in Section 3.

VMware was a key tool to providing a single PC SCADA Honeynet solution that was easy to deploy. Recent changes in VMware licensing allowed the creation of SCADA Honeynet virtual machines (VM's) that can be distributed at no cost to the recipient. With VMware there is a quite a bit of communication between virtual interfaces occurring in the single SCADA Honeynet PC, as shown in Figure 1. Fortunately this complexity is hidden from the user.

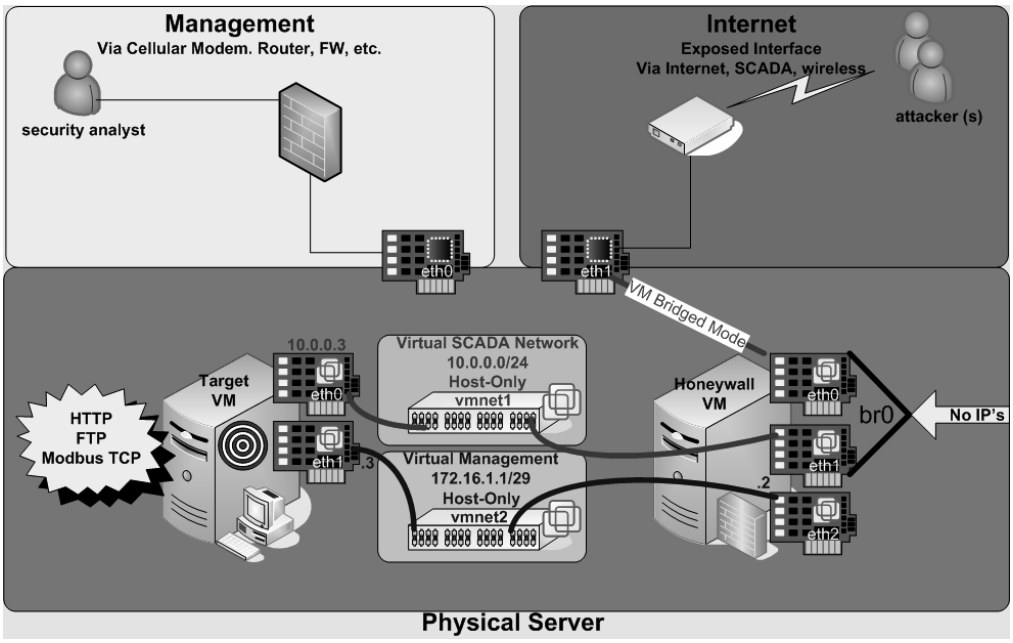

Figure 1 – SCADA Honeynet Architecture

The large box at the bottom of Figure 1 represents the PC. In the PC is the host Linux OS that runs VMware with two virtual machines, a Honeywall VM and a Target VM. Communication is switched and bridged between the three systems as required.

The diagram shows two physical interfaces, eth1 that is exposed to hackers and eth0 that is used for management. The physical eth1 actually does not have an IP address. All traffic is bridged through the Honeywall VM and to the eth0 interface of the Target VM.

We have tested the ease of deployment by providing the two VM's and a two page installation instruction document to a few beta testers. They were all able to install and operate the SCADA Honeynet without any support calls, although they did provide a number of suggested improvements.

2.1 Honeywall Modifications

The Roo Honeywall CD, a "third generation" (i.e. production quality) honeynet, was used in the Honeywall VM. The Honeywall CD provides a modified version of Fedora Core 3, with a number of core components:

- Management Interface – web and command-line (via SSH) options for initial installation, configuration, and monitoring of the honeynet.
- Monitoring Agents – integrates raw PCAP data, Snort IDS alerts, Argus flow data, iptables firewall logs, as well as data from Sebek agents which can capture activity from Windows or Unix hosts.

2.1.1 Configuration Changes to Roo

The SCADA Honeynet design implements a number of changes and additions to the standard Roo deployment. This paper only describes changes and additions that are necessary or significant to successful monitoring of our SCADA field devices target. The document does not detail enhancements and configuration changes that other non-SCADA Honeynet users might adopt, such as upgrading RPM's via "yum".

2.1.2 Integration of Digital Bond's SCADA Snort Signatures

Since the 3rd generation Honeywall integrates Snort signatures, raw packet captures, and flow data, Digital Bond's Modbus TCP and DNP3 IDS signatures were added to the base set of rules. This enables SCADA protocol attack detection and visibility in the Honeywall management interface.

2.1.3 Changes to Forwarding Path and Firewall Rules

When using the Honeywall as a filtering bridge between the external network and a standalone device, the Honeywall firewall rules are used without modification. However, the simulated field device requires modifications to restrict protocols and ports that are passed through the bridging firewall. For example, the Honeywall VM blocks all ports that are not open on the target PLC to better simulate the PLC.

2.1.4 Miscellaneous Configuration Changes

Minor modifications to the daily traffic summary script were made to provide additional information needed for daily reports and increase the amount of detail on ports and source IP addresses included. Additionally, the design modifies the criteria for alerts that will be generated by modifying the "swatch" scripts that continuously monitor iptables logs, and adds a "health check" script from the Honeywall VM to the Target VM. An alert is automatically generated if communication or application failures are detected.

3 SCADA Honeynet Services

3.1 Web Simulation

Most PLC's released in the last five years offer a web interface for administration, which exposes a commonly attacked protocol to adversaries who may initially lack any knowledge of control systems. Automated attacks on tcp/80 may identify web server vulnerabilities or weak login credentials. Manual attacks may view the PLC's web pages, identify the device, and enable an attacker to research known vulnerabilities to craft an attack on the web server or SCADA ports.

The target PLC that the SCADA honeynet simulates has a web administrative interface. A http service was required to be realistic and increase the attack surface.

We selected FizMez, a java-based GPL licensed project, for the web service. FizMez is an intuitive yet, portable service for emulating a PLC web interface. Emulation of some SCADA services may require proprietary simulators or services running on Microsoft Windows and java provides this future flexibility.

Attackers can often determine the web server vendor and version number by telneting to tcp/80 and issuing a GET command. Since the target PLC does not use the FizMez http service it was necessary to modify the PLC web server fingerprint as shown below.

```
telnet popular.plc.digitalbond.com 80
Connected to popular.plc.digitalbond.com.
Escape character is '^]'.
GET index.htm HTTP/1.1

HTTP/1.0 200 OK
Server: DECORUM/2.0
Content-Type: text/html
```

Emulating the PLC web service daemon required a minor change to the source of the FizMezWebServer.java file. As the Server response-header field output showed, the realistic response to emulate is"DECORUM/2.0".

```
conf.put("ServerString","DECORUM/2.0");
```

The second step essential to emulating the web server daemon, is propagating FizMez with PLC content. This can easily be done by copying the content off the PLC device with a tool such as wget.

Spidering tools such as wget connect to the PLC web services, authenticate with the proper default credentials and begin to download all content into a hierarchical output. This content contains the java applets and graphics that FizMez will use when an attacker connects and attempts a web administration request.

Once the content is loaded an attacker is presented with identical web administrative interface as shown in Figure 2 below.

Figure 2 – SCADA Honeynet Web Interface

The web administrative interface does not have the full functionality of the actual PLC. Most importantly, it supports the user login and is set to accept the default credentials. Various configuration and diagnostics capabilities are supported, and it is likely that even a sophisticated attacker with knowledge of the PLC will interact with the web interface for a period of time before realizing something is amiss.

3.2 FTP Simulation

Another widely used administrative interface is the File Transfer Protocol (FTP) service. FTP allows users to transfer files to and from the PLC for firmware upgrades and device management. FTP has a history of known vulnerabilities and confidentiality weaknesses.

The target PLC contained a VxWorks FTP server. To appear realistic, the SCADA Honeynet must have both the directory and file structure of the legitimate VxWorks FTP server along with the appropriate FTP banners.

This was achieved with another GPL licensed portable java project, iFTPd. We found iFTPd to be flexible, and it allows for virtual file systems, users, and groups. This has the advantage of allowing researchers to disregard real file system structure and system users/groups keeping the FTP service platform independent.

The actual PLC's FTP server contained version information within login banners. Details on software versions can provide an attacker with information useful for

researching vulnerabilities and gathering exploits. This is a typical early reconnaissance step by a directed attacker. A FTP client or telneting to port 21 of the target PLC will display a banner similar to the one below.

```
telnet popular.plc.digitalbond.com 21
Connected to popular.plc.digitalbond.com
Escape character is '^]'.
220 VxWorks FTP server (VxWorks 5.3.1)
530 USER and PASS required

Connection closed by foreign host.
```

PLC vendors have recently started removing version information from login banners. It is not known if this obfuscation is done for security purposes or possibly a side effect from different development processes. Below is a sample from a PLC without FTP server version information.

```
telnet popular.plc2.digitalbond.com 21
Connected to popular.plc.digitalbond.com
Escape character is '^]'.
220 FTP server ready.
530 USER and PASS required

Connection closed by foreign host.
```

The SCADA Honeynet duplicated the target PLC's FTP login banner by changing a setting in the ifpd.conf configuration file as shown below.

```
# Server Name
servername = VxWorks FTP server (VxWorks 5.3.1)
```

If the simulator IP address is using Network Address Translation (NAT) for connectivity, as it is in our SCADA Honeynet design, the iFTPd must properly be configured with the externally available address and a range of useable passive ports (PASV). Below is an excerpt from the iftpd.conf file with variables relevant for iFTPd behind NAT.

```
# FTP data ports
#
# Range of data ports to accept PASV connections on.
# Have at least as many ports open as you expect concurrent users and
transfers.

# Default:  20001-20100
dataports = 50000-50020

# PASV IP
#
# If you are running iFTPd on a NAT, it will not be able to
# detect the IP the client has connected to.
#
# Help iFTPd by setting what IP it should report to the client.
```

```
#
# Note: this only works if the iFTPd's IP is private and the client
# IP is on a public network.
#
# default: [empty]
pasvip = 166.x.x.x
```

Once an attacker has identified the PLC via the banners or other fingerprinting, he may try to login to the FTP server. The SCADA Honeynet has implemented the default usernames and passwords to login into the system and present directory structure and files. Users are configured in the SCADA Honeynet by creating a text file with the name of the user in the conf/users directory of iFTPd. The user named file contains variables specific to that user along with permissions and password information. Below is the output from the file conf/users/sysdiag. sysdiag is one of the default accounts on a Schneider Modicon PLC's[2].

```
password=factorycast@schneider
groups=standard
access=
```

A group file can be created for easier permission handling across multiple users. Below is what is contained in the conf/groups/standard file. Note that the root directory presented to the FTP users maps to a physical location on the device.

```
"/digitalbond/simulators/ftp/vfs/" r+ "/"
```

3.3 SNMP Service

The Simple Network Management Protocol (SNMP) is used for monitoring the health of networked devices. Administrators can read an extensive number of values from devices ranging from interface statistics, software versions, and internal component temperatures. SNMP also provides the ability to write certain values remotely as a form of automated administration.

SNMP simulation on the SCADA Honeynet was achieved by utilizing a SNMP emulator. Two projects exist, fake-snmp and Net-Raddle, both designed to interoperate with Honeyd. Both projects provide information to an attacker when he performs an snmpwalk on a Honeynet. The fake-snmp script does not change values such as interface statistics or timeticks, this may be suspicious to an attacker. Net-Raddle provides the ability to have timeticks change and interface statistics increase during each walk. Both projects have been very useful when assembling a honeynet.

At the time of this paper's publication the SCADA Honeynet is using the fake-snmp service with static values. We have integrated the Net-Raddle in the lab, but found stability issues that are not yet resolved. We hope to upgrade the SCADA Honeynet to SNMP values that change in the near future.

---

[2] The default credentials for the target PLC are easily identified in an Internet search.

The target PLC has a SNMP Management Information Base (MIB), and this MIB has been implemented in the SCADA Honeynet. We exported values from a working PLC to provide realistic, albeit static, SNMP data.

3.4 Telnet Service

Many PLC's are still managed by telnet even though the security problems with the cleartext transmission of passwords is widely known. The target PLC supports telnet so it is also implemented in the SCADA Honeynet.

In addition to the cleartext password issue, some PLC's will freeze or reload if large characters strings are passed to the username or password fields of the PLC. We do not know if this is a problem in some RTOS telnet services or poor programming techniques of PLC vendors. In either case it is interesting to see if and how attackers will go after the telnet service.

Providing an attacker with login capabilities was essential to developing the SCADA Honeynet. The simulated telnet daemon available from the Honeynet Project [1] did not meet the portable java specification of the other services, but did utilize a portable Python script. An older SCADA Honeynet Project [3] utilized a VxWorks telnet daemon with low to medium interaction for the attacker. This Python script required some minor editing, but fulfilled the requirements of allowing an attacker to login and issue VxWorks telnet based commands. The minor editing encompassed customizing the output when an attacker issues a 'help' command and changing the log file path. Below is an example of a login provided by the telnet simulator and a 'help' command issued.

```
VxWorks login:
Password:
PLC# help

help                          Print this list
ioHelp                        Print I/O utilities help info
dbgHelp                       Print debugger help info
nfsHelp                       Print nfs help info
netHelp                       Print network help info
spyHelp                       Print task histogrammer help info
timexHelp                     Print execution timer help info
h         [n]                 Print (or set) shell history
i         [task]              Summary of tasks' TCBs
ti        task                Complete info on TCB for task
sp        adr,args...         Spawn a task, pri=100, opt=0,
stk=20000
taskSpawn name,pri,opt,stk,adr,args... Spawn a task
td        task                Delete a task
ts        task                Suspend a task
tr        task                Resume a task
d         [adr[,nunits[,width]]] Display memory
m         adr[,width]         Modify memory
mRegs     [reg[,task]]        Modify a task's registers
interactively
pc        [task]              Return task's program counter
```

```
Type <CR> to continue, Q<CR> to stop: nfsHelp

iam       "user"[,"passwd"]     Set user name and passwd
whoami                          Print user name
devs                            List devices
ld        [syms[,noAbort][,"name"]] Load stdin, or file, into memory
                                (syms = add symbols to table:
                                -1 = none, 0 = globals, 1 = all)
lkup      ["substr"]          List symbols in system symbol table
lkAddr    address               List symbol table entries near address
checkStack  [task]              List task stack sizes and usage
printErrno  value               Print the name of a status value
period    secs,adr,args... Spawn task to call function periodically
repeat    n,adr,args...    Spawn task to call function n times
(0=forever)
version                    Print VxWorks version info, and boot line

NOTE:  Arguments specifying 'task' can be either task ID or name.

value = 1 = 0x1
```

As with ftp and http, the SCADA Honeynet implements the default telnet credentials in the target PLC.

3.5 Modbus TCP Service

Modbus TCP [4] is a commonly used application layer SCADA protocol that originated from development work by the vendor who manufacturers the target PLC. So it is highly likely that the target PLC would support Modbus TCP for reading and writing points.

However Modbus TCP is not nearly as common of a service as ftp, http, snmp or telnet, and it was more difficult to identify a Modbus TCP simulator that was Open Source and built with java. Eventually we found the Jamod development tool kit for Modbus TCP masters and slaves, and it met all of the SCADA Honeynet criteria.

A custom Modbus slave was written and popular function codes were implemented. Attackers can read values and write values, both analog and discrete. Below is a sample log output from the Modbus slave simulator showing function code activity.

```
INFO: Making new connection
Socket[addr=/10.0.0.87,port=2721,localport=502]
Received ModbusRequest FC:1
Request:00 06 00 00 00 06 01 01 00 00 00 01
Response:00 06 00 00 00 04 01 01 01 01
Received ModbusRequest FC:3
Request:00 07 00 00 00 06 01 03 00 00 00 01
Response:00 07 00 00 00 03 00 83 02
Received ModbusRequest FC:3
Request:00 07 00 00 00 06 01 03 00 00 00 01
Response:00 07 00 00 00 03 00 83 02
Received ModbusRequest FC:3
Request:00 07 00 00 00 06 01 03 00 00 00 01
```

```
Response:00 07 00 00 00 03 00 83 02
Received ModbusRequest FC:6
Request:00 08 00 00 00 06 01 06 00 00 00 00
Response:00 08 00 00 00 03 00 86 02
Received ModbusRequest FC:6
Request:00 08 00 00 00 06 01 06 00 00 00 00
Response:00 08 00 00 00 03 00 86 02
Received ModbusRequest FC:6
Request:00 08 00 00 00 06 01 06 00 00 00 00
Response:00 08 00 00 00 03 00 86 02
```

We programmed in a points list and sample data obtained from a large electrical utility located in the United States. This provides a highly realistic target. An extremely savvy attacker could probably even identify the instruments that were being monitored and controlled.

3.6 Future Target VM Development

There are a number of potential improvements to the Target VM. Two that interest the research team most are:

1. Adding the capability of customizing the points list and data through a configuration file. This would allow asset owners to have the SCADA Honeynet appear very much like one of their own PLC's and make it more effective as an early warning device. It also would be an easy way for researchers to deploy a variety of SCADA Honeynets.
2. Adding one or more virtual HMI's running on Windows, a virtual OPC server and a few other devices to the SCADA Honeynet. This would provide a larger attack target surface and simulate a small control system or a small portion of a larger control system.

4 Analyzing Attacks

Honeynets are often deployed to learn how adversaries will perform their attacks and to provide information for the threat and the vulnerability variables of the risk equation. Most honeynets deployed today are mimicking devices that are very popular and common in corporate networks. They may appear to be vulnerable web servers with inherent application flaws connecting to back-end database servers.

SCADA Honeynets are unusual. They present a set of unique services uncommon to most attackers. In this project we exposed the SCADA Honeynet in two different scenarios.

4.1 SCADA Honeynet Deployment I – Internet Exposed

SCADA networks are typically segmented from corporate networks by a firewall or other perimeter security device. Field SCADA devices such as PLC's, RTU's, and IED's and their services are generally not accessible directly from the corporate network, let alone the Internet.

Strangely enough, even as awareness of cyber security issues increases, the number of Internet connected SCADA field devices is increasing. The reason for this is dramatically lower communication costs and increased bandwidth. Carriers such as Verizon, T-Mobile and Cingular are specifically targeting the electric utilities, pipeline operators, and canal operators with offers that appear to be too good to pass up.

The SCADA Honeynet Deployment I was exposed to the Internet using Verizon's service that is being sold to asset owners. The SCADA Honeynet was connected directly to the Internet on August 1, 2006 without any filtering through a GPRS Raven Modem that was recommended by Verizon.

Through the end of calendar year 2006, the SCADA Honeynet has not received any 'SCADA' attacks. There has been no activity on the Modbus/502 port. Port tcp/10000, which is used by DNP3, was scanned by attackers and captured by the SCADA Honeynet. Analysis determined that the scans were destined for a popular vulnerability in Cisco VPN Concentrators that serve VPN connectivity on tcp/10000.

While the lack of SCADA specific attacks is not unexpected, it does provide some indication that the automated attacks taking place on the Internet are not yet attacking SCADA ports and protocols. This warrants continued monitoring and SCADA Honeynets are one way to track this.

The activity on the Internet exposed SCADA Honeynet mirrored what one would likely be seen on any exposed system on the Internet. Figure 3 shows the activity by port.

A large number of HTTP attacks were targeted for vulnerabilities in web applications and services that did not affect the simulated PLC's web service.

Most interesting were the scans and login attempts on the FTP server. During one attack over 775 username and password combinations were launched against the SCADA Honeynet FTP server. The attack ran for 15 minutes and the FTP service became unstable which likely shortened what would have been an even more sustained attack.

This type of attack was automated and unintelligent as the timing of the username and password entries was very quick and no effort was spent researching what the device may be. Based on our very limited evidence, it appears that default credentials for SCADA devices have not yet been added to the popular automated attack tools. The SCADA Honeynet is one tool to determine if and when this changes.

The Internet exposed SCADA Honeynet showed no evidence of an attacker doing any research to exploit the simulated PLC. It would have been a simple task to look at the web interface, determine the PLC type and search and obtain default credentials. This again is not a surprise because the IP address was not in a range of a utility or other interesting target. It would be useful to see data from a SCADA Honeynet deployed in a SCADA asset owner's IP address range.

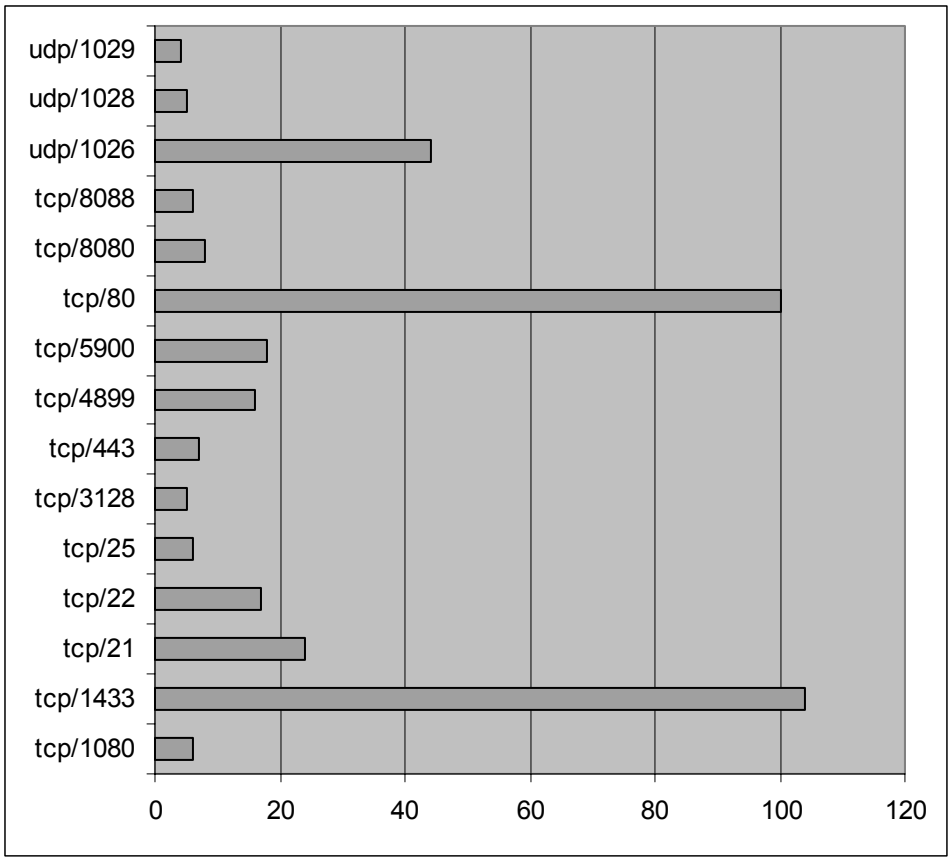

Figure 3 – Scanned Ports

4.2 SCADA Honeynet Deployment II – Electric Substation

Wireless LAN's are another popular communication media because they are easy to deploy and eliminates the time and cost of wiring. It can provide a quick ROI and a significant convenience to manufacturing plants, electric substations, pumping plants and other facilities where control system devices must be networked.

The SCADA Honeynet Deployment II was deployed on October 1st and consisted of an open wireless 802.11g access point in an electric substation. The substation was located around highly populated, lower income apartment complexes. The SSID was intentionally named to represent the electric utility and DHCP was enabled to easily allowing remote computers to connect.

As we expected almost all of the communication to this SCADA Honeynet involved computer users in the area trying to get Internet access. Over 90 different computers, tracked by MAC address, connected to the wireless access points in the first 60 days. What is most striking is the number of unique users that connected in the first five days as displayed in Figure 4.

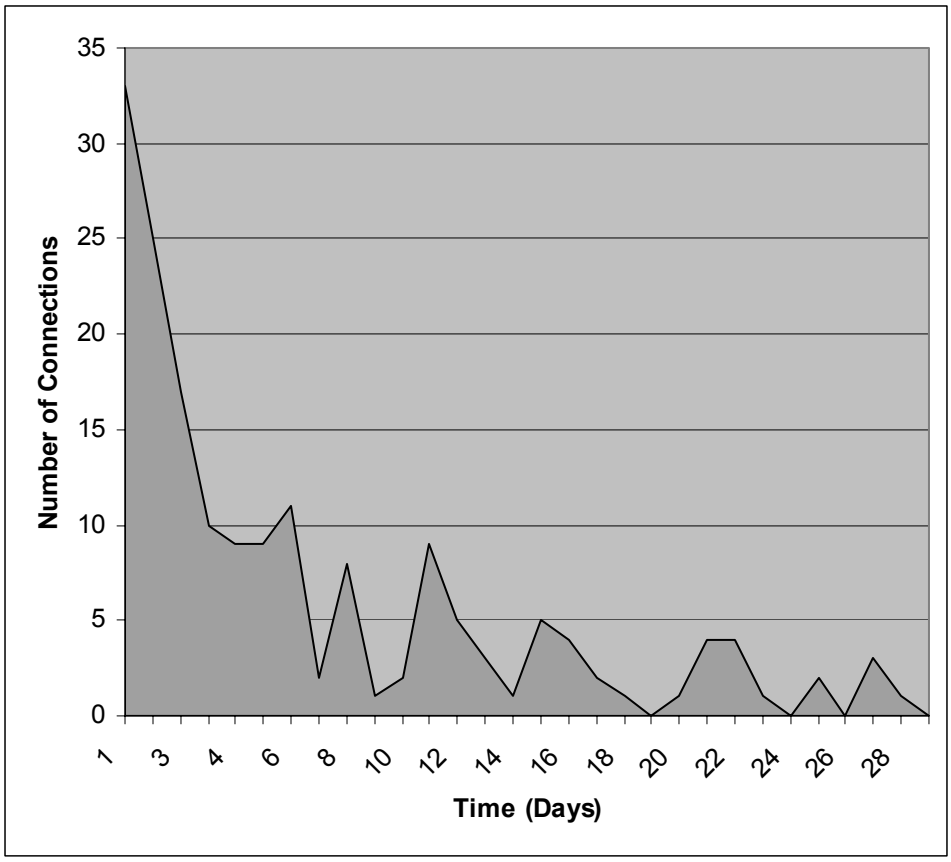

Figure 4 – Users Connecting to the Wireless Access Point

What Figure 4 demonstrates is there is no grace period during installation and testing of a wireless access point. All wireless communication must be secure from day one, and the risks due to a faulty deployment could be serious.

We continued the traffic analysis to determine what Internet sites the remote users were attempting to reach. Figure 5 shows the captured DNS queries. Note that a significant portion of the queries are spyware or software programs used to download copyrighted materials. This indicates that even if the user is not directly trying to compromise confidentiality or integrity of the SCADA data or network (only trying to steal some bandwidth) the mere use of the network could expose the SCADA network to malware and bots.

The situation would be even worse if systems on the SCADA network were allowed to access the Internet. Over time there could be a noticeable impact on network performance as more 'users' learn of the availability of 'free' Internet access.

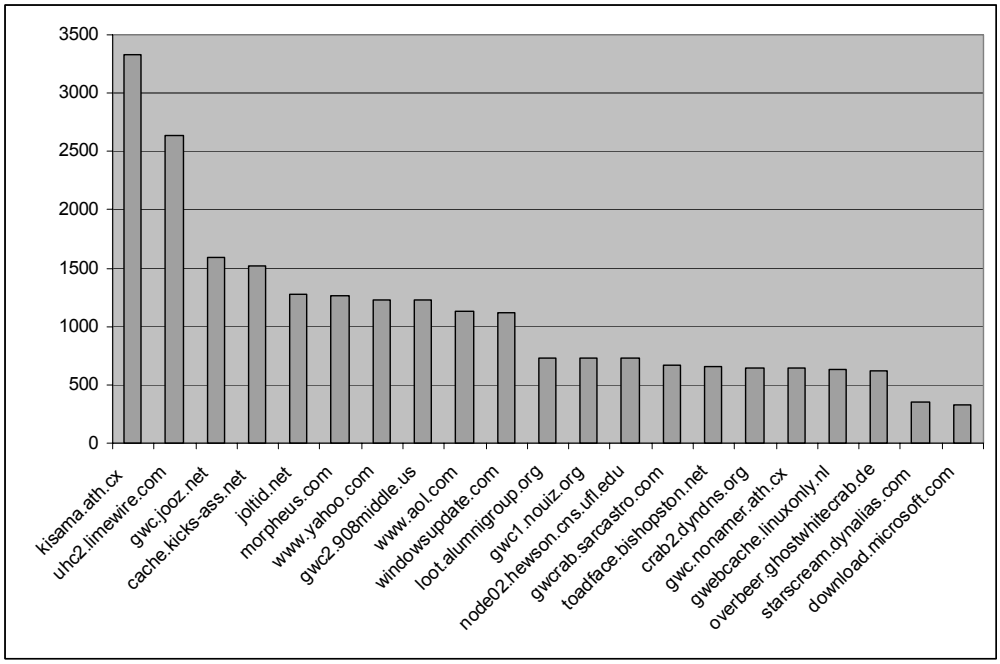

Figure 5 – Captured DNS Queries

Just as in the Deployment I, we did not see any attacks on the SCADA ports or applications. The number of adversaries who can attack the wireless access point in Deployment II is smaller than Deployment I because physical proximity is required. That said, Deployment II is associated with a major electric utility so a directed attack is more likely.

5 Conclusion

The research project was successful in designing and implementing a high interaction SCADA Honeynet that is a realistic simulation of a popular PLC. The use of VMware virtual machines (VM's) makes it easy for a reasonably skilled IT professional with minimal knowledge of Honeynets or PLC's to deploy a SCADA Honeynet. The SCADA Honeynet VM's are on Digital Bond's site and are available to any S4 participant.

The data from the small number of deployed SCADA Honeynets indicates that control systems still have the benefit of security by obscurity. One of the biggest benefits of a SCADA Honeynet may be to identify when this changes. If attacks on SCADA ports or using SCADA default credentials are seen in a SCADA Honeynet this would indicate a shift in the threat profile.

Given there is no license fee associated with the SCADA Honeynet software, a wider deployment of the SCADA Honeynets is inexpensive and would provide a more statistically significant assessment of threat. This technology may be an ideal project for one or more of the many graduate and doctoral students that are beginning research in SCADA security.

About the Authors – Landon Lewis is Security Consultant in Digital Bond's SCADA Security Consulting and Research Practice. He is the lead researcher in Digital Bond's SCADA Honeynet Project and a key contributor in the development of Nessus SCADA plugins. Prior to joining Digital Bond, Mr. Lewis was a Cyber Security Analyst with Midwest ISO (MISO), a large Independent System Operator/Regional Transmission Operator (ISO/RTO). While at MISO Mr. Lewis and his team designed, implemented, and maintained the security architecture around the world's largest energy market.

Dale Peterson is the Director of Digital Bond's SCADA Security Consulting and Research Practice. He has over twenty years experience in information security beginning as a cryptanalyst for the National Security Agency (NSA). Mr. Peterson has led many of the efforts to integrate SCADA intelligence into IT security solutions including SCADA IDS signatures deployed in most commercial network IDS products today, Nessus SCADA plugins, and now the SCADA Honeynet.

References

[1] Spitzner, Lance. Honeypots: Tracking Hackers. Addison-Wesley, September 2002.

[2] The Honeynet Project. Know Your Enemy: Learning About Security Threats. Addison-Wesley, Second Edition, 2004.

[3] Pothamsetty, Venkat. SCADA Honeynet: PLC Simulation Concepts, Design, and Implementation.

[4] MODBUS Application Protocol Specification V1.1, November 2002