COMP 8045 - Practicum I

Project Proposal

nCase: Designated Access Devices

Prepared by:

David Klassen A00026160

Researcher Introduction

So far my time in the software field has involved purely Engineering Services work for Enterprise and Packaged Software companies. By Engineering Services, I mean services which help complete the software engineering lifecycle as it relates to producing a product that customers can use. Typically the services of Customer Support, Configuration Management, Build and Release Engineering, Quality Assurance (QA), and ThirdParty Software Integration can be considered as Engineering Services. With the proliferation of the latest distributed computing paradigms involving SOAP, CORBA, .NET, J2EE, Applications Services Provision (ASP), and Service Orientated Architectures (SOA), a large portion of Software Engineering and Design work has been redirected towards filling in the gaps which these new architectures enunciate. In other words nearly the end of Commercial Off the Shelf software (COTS).

As RPC, DCOM, RMI, CORBA and SOAP merged into SOA, and as ASP slowly shifted into Business Process Outsourcing (BPO) and Software As A Service (SaaS), the Engineering Services departments of most companies have dramatically shifted to keep up. Essentially the generally poor software engineering practices accidentally discovered by Microsoft in 1995, have been fully proliferated into every computing company on the globe, perhaps even into wireless software companies. At this point in my career I am fairly sure I should just shake hands with the industry, and start developing software.

Employment History

Dec '01 – Apr '05 **Interwoven, Inc.,** Sunnyvale CA, U.S.A

Release Engineer: Integrated several Release Engineering environments into a centralized multi-site environment and test suite, to help manage change to internal processes and release methodologies. Participated in automating service pack development, created new build script features, corrected build errors, and prepared new components for building. Helped create the toolsets for cross platform builds, and porting initiatives. Performed build tests to determine optimum hardware configurations for build servers. Created a build request CGI and helped maintain an internal build reports website. Resolved build issues/errors with developers (ie. Build watching etc.).

Aug '00 – June '01 **XUMA Inc.**, San Francisco CA, U.S.A.

Release Engineer: Implemented a GNU based build system for java developments. Designed and implemented an installer for XUMA's multi-tiered internet content server (CommerceX), using SUN packages, Perl and shell. Helped to manage the CVS repository. Responsible for giving input into discussions related to release management, and the proper tracking of product version numbers. Installed the product for one of XUMA's main customers AEGIS (www.aegislink.com).

June '99 - Aug. '00 **Xcert International Inc.** (hitherto RSA/EMC), Vancouver BC, Canada

Quality Assurance Engineer: Participated in the design and development of product test suites. Developed new ways to automate the detection of security holes in the products using C/C++, Expect, s-client, and SilkTest (an Automated Test Tool). Responsible for finding, reporting, and tracking bug details and information. Organized testing for the "WebSentry" product (Test scenarios, thirdparty product integrations, and reporting). Designed a departmental web site. Attended design and review meetings concerning product development.

Table of Contents

Illustration Index

I. Background

The Internet has brought average consumers the ability to perform many activities remotely. For communities that are quite small and remote this is an added benefit. However these same communities are still far away from most corporate distribution sites. So while they can order goods and shop for services on-line, reasonable shipping rates and reliable shipping methods do not always prove to make the experience entirely rewarding. For instance I have a sister-in-law who lives in Fort Nelson, BC who ordered a product from Ikea multiple times. Every time the product was shipped to her destination address it arrived maimed or broken. Finally in mid-2004 she asked me to buy the item for her, as I had decided to visit their family on a summer vacation. The Internet definitely allowed her to see what she could buy, and that is good (much better than a paper catalog), but perhaps paying for an item which takes an infinite time to arrive as ordered, is not a good service.

Currently the Internet is the most integrated computer based network ever to exist on earth, it may even be the largest network to ever exist. Its growth carries with it nearly all the same sociological repercussions that regular metropolitan growth incurs. From increased usability and practicality to rumored increased profits and reduced expenses, all provided at the speed of light. These conceived benefits to society however also come with new forms of crime, espionage, sabotage, and destruction. The onset of computer worms, viruses, Denial of Service (DOS) attacks, and exfiltration reveal the volatile and insecure nature of electronic information (Cunningham, Robert et. al). Like in the above story the Internet does deliver something new to us, but this new medium comes with many of the same risks the non-Internet world has, and worse yet our public critical infrastructures are at risk.

Computer Networks

Contrary to common public knowledge there was not only one remote network created (ie. The Internet). In fact most of the first computer networks created used various different communication protocols and mediums. Needless to say when the first computer network technologies were designed (Shannon, Claude E.), the idea was to connect remote computer systems together, not necessarily remote societies. Even the first complex electronic calculator was exhibited remotely (Stibitz, George), and so computing in general has never been about a location or fixed areas, so much as net achievable processing power. Some might say the introduction of electronic computing and the Turing machine, allowed World War II (WWII) allied nations to succeed in learning how to end the war (Turing, Alan) via. decryption.

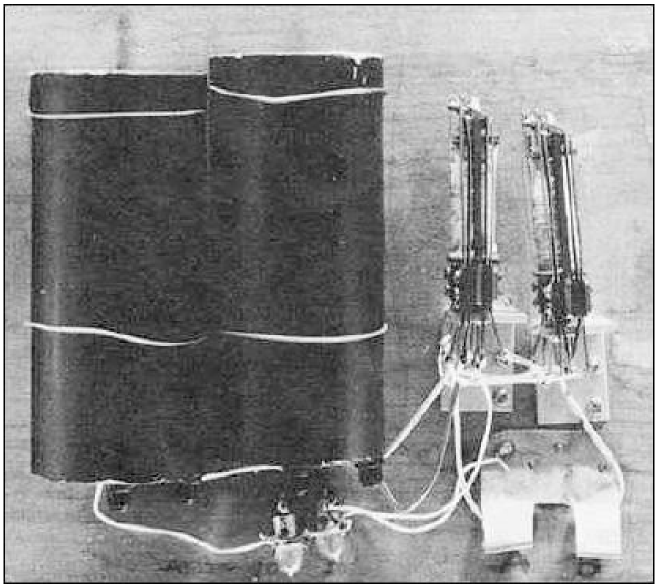


Illustration 1: Bell Labs Complex Electronic Calculator.

After the end of the war large learning institutions, government organizations, corporations, and the engineering profession benefited from these technologies. The conversion of machinery from analog instrumentation to binary digital instrumentation, was not a fast process however. When practical computing machines were eventually built (leo-computers.org.uk), connecting the various components of the computer required its own communication protocols. It would have been considered unnecessary to connect two computers as the goal at that point was not communication, as much as it was creating the machine itself. Bell labs used their resulting designs to log and charge for calling times on the public phone networks. However eventually computer to computer communication did become desirable. Luckily the communication industry had over a century earlier created methods for communicating information through telegraph and later facsimile, and modulating data onto ordinary communication lines was their main business. Bell labs demonstrated remote computer user connectivity first in 1949, but their first proposed communication mechanism for computer to computer exchange was not demonstrated until 1956 (Brown, W. Stanley. et. al).

Illustration 2: Early phone line communications modem.

This was later implemented in the SAGE radar military defense network (later to be called NORAD), over AT&T (Bell) and Canadian Telephone lines using modems. Since at this time AT&T had a monopoly in the US Telephone Network market, they provided and regulated which modems and equipment could be attached to their network. In 1968 AT&T was forced to break-up the monopoly and this rule was relaxed for other manufactures whose products were then required to pass stringent AT&T designed tests. Digital transmissions have taken quite a while to proliferate throughout the phone networks, due to their historical placement in society. Even today phone lines are still analog from the local office to the end user's phone jack (Leon-Garcia, Alberto. et. al). Computer to computer communications however have taken off regardless of the co-operation of telephone networks, and many other network mediums have arisen.

Essentially the early private networks around the globe all used different communication protocols. At that time a proprietary protocol may have been considered secure, since very few other people on earth possessed the design documents, or that information was not for public release. The perception of public threat had been reduced with of the end of WWII. However security by obscurity does not make for an entirely secure system, since the information conveyed may perhaps only be encoded. If the encoding could be reverse engineered, an informed organization would be able to decode the network information. Generally communications were not encrypted using two factor encryption, until the late 1970s with the introduction of RSA and DES. However there are rumors essentially that the British had conceived these techniques before their mainstream arrival (Schneier, Bruce).

Communications Protocols

These initial master/slave communication protocols dealt mainly at the physical and data link layers of network communication. Proprietary serial line Text Terminal (TVI950, VT100), and file transfer (X-Modem, SEAlink) protocols were used for inter-computer communications. This stage of computer communication was not highly popular with the general public, since you needed to have explicit computer knowledge and reason for connecting to another computer. During this time it was not uncommon for engineers to work for institutions which used electrically operated machines and perhaps computers to control the various operations which kept our 20th century society moving along smoothly. From bridges, railways, industrial manufacturing, heating and A/C systems, water and waste management, oil refinement, hydro dams, to aerospace, research labs, and nuclear power generating plants, the machinery responsible for orchestrating these facilities gradually required a method of computerized and automated control.

With the adoption of Programmable Logic Controllers (PLC) in the late sixties, the operation of these machines became controllable remotely. When the PLC devices were installed they were connected via. serial lines and modems called a Fieldbus, to a controlling computer. These engineer orientated computer networks became known as Process Control Networks (PCN). The protocols for controlling devices on the Fieldbus were developed on a manufacturer specific basis (Modbus, Profibus) and so, many variations still exist in operation today. Of special consideration here is the deterministic nature of these networks, they have been created for industrial automation and scientific purposes that demand real-time network availability for precise device action. No measure of network delay due to packet collisions can be afforded in systems like this, therefore probabilistic network media are normally not used.

Illustration 3: Control panel at Kennedy Space Center.

It was not until packet switched protocols like X.25 (npl.co.uk) and Internet Protocol (IP) were created could multiple remote computers connect to each other in unison. Prior to this multiple lines of communication were essentially circuit switched (ie. multiple modems). Essentially a network bus with multiple processing units was provided for by X.25, IP, IPX, VINES, and NetBIOS. All of these protocols were initially proprietary and the defined networks rarely extended outside a licensed installment. From the gradual amalgamation of the TCP/IP networks of the late seventies until the first commercially available connections in the mid Eighties (greatachievements.org), the Internet was not the incredibly large network that it is now. In fact the first publicly available ISP did not surface until 1989 (wikipedia.org). Prior to this any home computer user seeking interconnectivity relied on using a modem, to connect to a tertiary system (mainframe, or other computer) which was accessible on the receiving end via. modem. The technologies used via. modem did not initially use TCP/IP, but the master/slave serial communication protocols mentioned above.

Social Networks

From the very beginning the development of the Internet was considered to be more of a computerized social network, very much like the phone system had provided an analog/electrical social network. It was determined that a computer network would not suffer from the same vulnerabilities inherent in the old circuit switched network. If one communication system had been severed from the network due to catastrophic circumstances the other portions of the network could recover smoothly and adapt to the new found circumstances (state). As the history of scientific exploration has shown time and time again two people or two groups of people happen upon having the same idea at the same time (Or so it appears to their conscious state of mind.), such was the case with packet switching and the Internet (Kirstein, Peter T.). While in this circumstance the two nations worked together in one form or another, I wonder how lucky the rest of us will be working with our Internet twin?

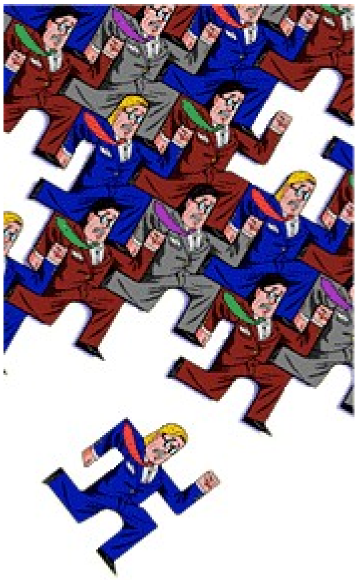


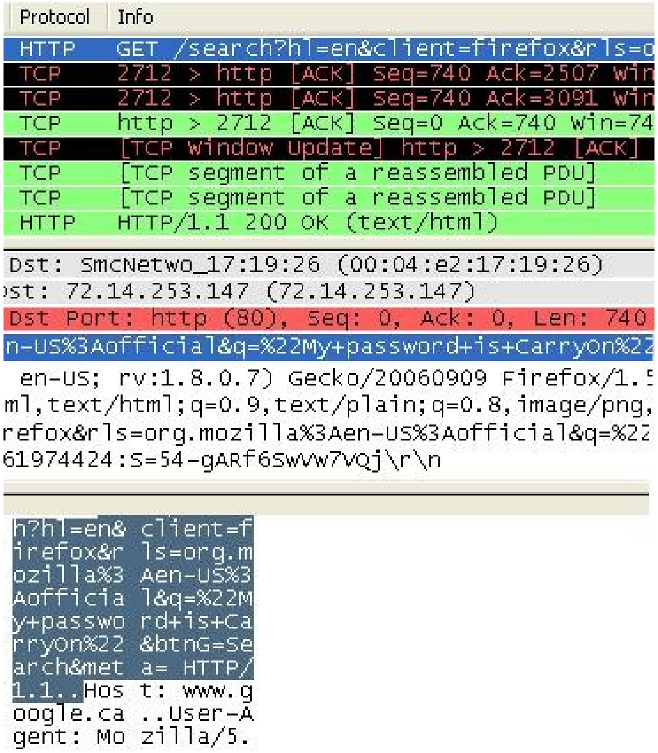
Illustration 4: Social Networks?

While at least two different implementations of this abstract network layer protocol were created, the end result was that both networks were eventually connected from North America to Great Britain, and eventually the IP protocol was adopted as opposed to X.25. Like X.25 the usage of other network layer protocols decreased over the years. With the increased availability of cheap IP compatible communications hardware and software the Internet nearly has taken over from where the phone system left off. ASDL has created a reason for delaying ISDN services (Leon-Garcia, Alberto. et. al), and Internet capable communication network providers have entered into a competition recently to provide Voice Over Internet Protocol (VOIP), digital phone services. In addition Wireless Internet Access coverage has risen at a staggering rate during the last ten years. Cellular phones have ceased being only SMS capable and full Internet connectivity has resulted. In fact the concept of using VOIP over cellular networks is also currently available using Session Initiation Protocol (SIP).

The Internet has already created a net decrease in the number of newspaper sold, and most tree-abiding citizens consider this a good thing. At the same time as VOIP was being implemented by various competing organizations (vovida.org), on-line music was becoming quite prevalent. On-line video programming is now also starting to see some traction towards Internet delivery, who knows whether it will do better or worse than Video On Demand (VOD) or Digital TV (DTV). However this is a key concept with many aspects of the Internet, “Who knows?”. Perhaps this is the same thought that went through the scientist's minds who have abandoned the Internet for Internet2 and High Performance Grid Computing. I paid $15 dollars for a three month VOIP phone number subscription this summer, and I was not overly impressed, the system was unreliable. This is the way it is with most things related to the Internet, there are undeniable quirks.

I am not saying I do not make use of the Internet because I think it was poorly designed, however for me the Internet is a great on-again off-again network. I do not see it as a long term network solution. It is the first step towards a much more reliable network. Like-wise I do not trust the Internet, I am sure that I trust the Internet more than some people, but it is only because I understand a fair bit about it that I can justify my trust. Ordinary users might not be so lucky to learn the necessary precautions to use while using the Internet. Luckily I am naturally very cautious, of what I know nothing about. However this leads us to something of a galactic misconception. Using the Internet is not the same as choosing to turn on an electrical switch so I can perform transactions of some kind, turn off the switch and rest assured all my transactions completed safely and the necessary data stored locally. No when my computer sends data to the other computers on the Internet it can pass through a number of other networks first. How do I know that there is not an ill-fated entity waiting on one of these intermediary networks to intercept my personal information? Worse how do I know that such an entity is not performing reconnaissance on the hardware that I have left running during the day?

There are plenty of tools on the Internet that can be used to analyze my computer hardware for vulnerabilities, crack network passwords, plant reconnaissance programs, damage my computer, or obtain information about me. All the data that passes from my computer to another computer by my permission and my omission via. Internet technologies, does not just disappear when I turn off my computer. Protocol analyzers like ethereal shown to the right can process any network data I transmit and using unencrypted communication methods will tell an attacker more than they need to know about my habits. But even if I do use encryption for all my client actions, because basic Internet technologies do not use encryption, someone watching my network traffic can discover something about my habits without even attempting to decrypt the data. It just so happens somebody was capturing all that network traffic and picked up on my mistake. Or perhaps they merely want to know what my next research and invention topic of interest is?



```
Protocol  Info
HTTP      GET /search?hl=en&client=firefox&rls=o
TCP       2712 > http [ACK] Seq=740 Ack=2507 Win
TCP       2712 > http [ACK] Seq=740 Ack=3091 Win
TCP       http > 2712 [ACK] Seq=0 Ack=740 Win=74
TCP       [TCP Window Update] http > 2712 [ACK]
TCP       [TCP segment of a reassembled PDU]
TCP       [TCP segment of a reassembled PDU]
HTTP      HTTP/1.1 200 OK (text/html)

Dst: SmcNetwo_17:19:26 (00:04:e2:17:19:26)
st: 72.14.253.147 (72.14.253.147)
Dst Port: http (80), Seq: 0, Ack: 0, Len: 740
n-US%3Aofficial&q=%22My+password+is+CarryOn%22
 en-US; rv:1.8.0.7) Gecko/20060909 Firefox/1.5
ml,text/html;q=0.9,text/plain;q=0.8,image/png,
refox&rls=org.mozilla%3Aen-US%3Aofficial&q=%22
61974424:S=54-gARf6SwVw7VQj\r\n

h?hl=en& client=f
irefox&r ls=org.m
ozilla%3 Aen-US%3
Aofficia l&q=%22M
y+passwo rd+is+Ca
rryon%22 &btnG=Se
arch&met a= HTTP/
1.1..Hos t: www.g
oogle.ca ..User-A
gent: Mo zilla/5.
```

Illustration 5: Whoops I entered a secret on the google site.

So if the original community that invent the Internet no longer use it why am I, maybe they decided that it was too hard to get all the ISPs to agree on using the same security method for DNS services. Worse yet maybe they actually want the Internet to expose the secrets of my life. Can you imagine any woman writing her actual diary on the Internet? No the Internet is not about sharing your deepest darkest secrets its for performing activities that are completely superficial in nature, just like reality TV programming. The only way to use the Internet with a remote chance of obscurity and privacy is to use encryption. However now that my important data is out there on a corporate server, how do I know that the corporate servers will not be compromised. The proliferation of bot-nets, malware, viruses, worms, trojans, backdoor programs as well as network security test programs, only supply hackers with a larger arsenal for determining how to attack my home systems.

Social SCADA Networks

It was fine to connect PCNs to the initial computer networks, because they were mostly hosted in private and government sectors of the economy, and the connecting entities usually worked for these organizations under tight physical security parameters. The technologies were not widely known and few people other than specially trained engineers, were involved in the operation of these computer-centric operations. These initial systems were far too expensive, to be purchased by organizations which might focus on causing catastrophe. At that time even if these networks did encounter sabotage, it was the responsibility of designated organizations to protect against intrusion. Today however we have an economy that is driven by the US consumer market. If you want to make a deal with America, you must first satisfy the basic consumer. The profit from these endeavors is what fuels research and development, and therefore the older PCN networks which have been removed from government control and funding, are now considering using consumer based products for upgrading their PCNs to current state-of-the-art equipment (Schiffer, Viktor).

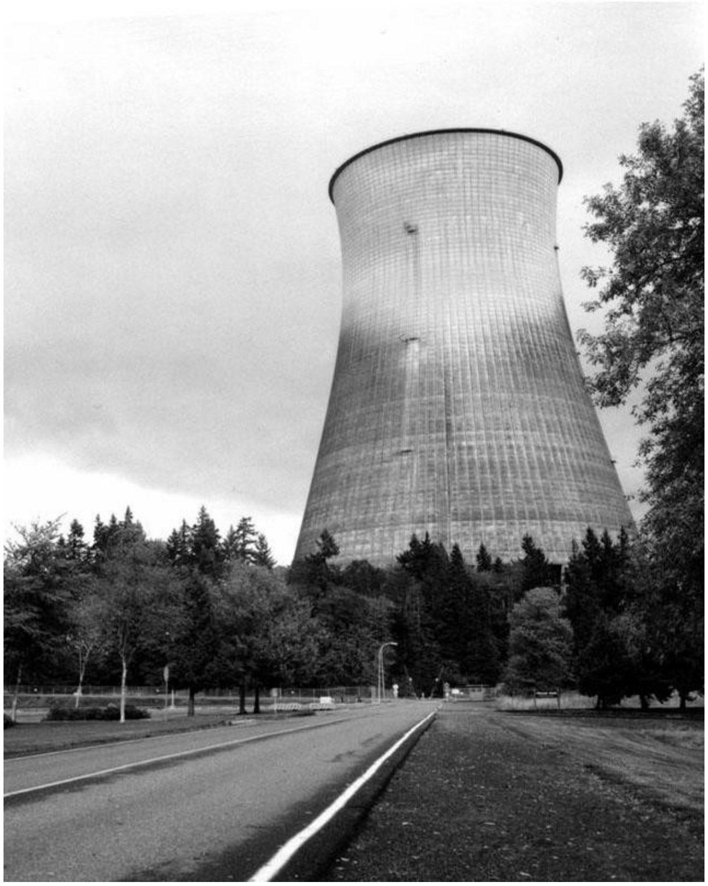


Illustration 6: A nuclear reactor now in hibernation.

Today's state of the art equipment uses non-deterministic equipment and protocols for connecting computer networks. In the past most PCNs and associated Supervisory Control And Data Acquisition (SCADA) systems depended on having deterministic networks to properly handle real-time automation issues. In deterministic networks it is imperative that each command sent to a device be received by that device. Since deterministic networks were designed to provide real-time and time critical, commands and data for automation equipment, sending a repeat packet for a packet that was not well received would cause the device to operate out of time, in time critical operations. Essentially a nuclear fission or electrical power plant operation might be off by a small time frame, and cause instability in a reactor or generator. Since a non-deterministic network cannot guarantee this reliability, it is conceived that using a probabilistic network in a switched star configuration, would make probabilistic network protocols (ie. CSMA/CD commonly called Ethernet) behave in a deterministic way.

For instance since a switched star configuration using the Carrier Sense Multiple Access with Collision Detect (CSMA/CD) Data Link Layer protocol would become deterministic, since the Collision Detect portion of the protocol would never need to be used. Essentially Collision Detection was implemented because every device on the network needed to detect if the network bus was free for sending data, otherwise a data collision might occur on the network, causing a positive feed back loop of all the packet resends issued from the devices. In a star configuration with only one switched device end-point, using a network medium and standard that use CSMA/CD like 802.3, would purportedly make the network error free and deterministic. However a study performed at Cisco Systems, Critical Infrastructure Assurance Group (CIAG) has shown that common 802.3 switches (which provide for IP Multicasting), are quite prone to DOS attacks that use the multicasting protocol (Hamadeh, I. et. al). In this paper it is shown that a hybrid computer worm that combines the functionality of both the Slammer Worm and Ramen Worms could effectively prevent multicast packets from being sent across the network. These study results released in the summer of 2005, document how these worms (which caused major security concerns in 2003 and 2001 respectively) are of great concern to PCNs.

The reason this finding is of great concern is that Rockwell Automation has created a new Fieldbus protocol (Ethernet/IP™) which uses the multicast protocol to create the device data bus on a switched CSMA/CD star configured network (Schiffer, Viktor). While Rockwell automation does not suggest using this Common Industrial Protocol (CIP™) for deterministic networks, perhaps the network designer will still consider using this protocol in situations which actually do require some sort of deterministic conditions without knowing it. After all a DOS attack does not just cause problems in a probabilistic network, it renders that network useless. In theory if someone from the Internet could hack into a PCN and launch an attack like this against the SCADA systems, a major catastrophe might occur. What adds to this problem is that Worm viruses after they are created and released into the Internet, do not necessarily provide any basis for determining who exactly created the worm. This is actually a major issue, worms tend to grow to infinitum unless systems are invulnerable (ie. blocked by the computer).

Illustration 7: Determine what happens.

The basic theory behind an Internet Worm is that it uses weaknesses in the innate properties of the Internet protocol, or the specific operating systems and environment which have access to a packet switching network. Once a worm is triggered the full effects of its devastation are unknown. Not all computers may be susceptible however the ones that are, will most likely pass on the Worm virus to any other computer in its reach, that is susceptible. By deduction therefore if an Internet Worm is introduced and makes it past corporate firewall defenses, into a PCN, that computer controlled PCN might be rendered useless. If a PCN is rendered useless major expenses may be incurred due to lost production, even worse if a catastrophe (ie. Pressure damage within dams, nuclear reactor instability etc.) results within the SCADA system controlled PCN, people may die or property may be damaged, creating a high degree of liability for the owning corporation.

Eric Byres from Tofino Systems, Inc. has mentioned several cases where the Slammer Worm has caused major problems due to human accident. One can only imagine what might happen if a targeted attack takes place. In a recent conference of PCN workers and device designers Eric detailed how unpatched Microsoft systems in these networks are particularly vulnerable to accident and attack (Byres, Eric). If we start to think about all the hacking tools available on the Internet, it is even conceivable that someone could stumble on creating problems like this on the Internet even by the accident of an inquisitive nature. Then there are people that do not take lightly to being slighted, like the vengeful Maroochy Shire, contractor/hacker. In 2000, this person caused a sewage plant in Queensland, Australia to dump sewage into several areas of the region. The attack was perpetrated remotely using a laptop and wireless radio equipment (theregister.co.uk). However according to recent news from the Director of Research at the SANS Institute (A major organization dedicated to Internet Security) actual extortion events related to PCN systems have also taken place (Paller, Alan).

Illustration 8: The consequences of a nuclear hack

If we mix a social computer network like the Internet, with automated PCN and SCADA systems which control the critical infrastructures present in the developed nations of the world, what will happen? This is not something that our governments are stopping to consider today. Some of these governments are still trying to deal with the effects that terrorism revealed in 2001. However while our house of commons and parliament are currently blocked by world-wide indecision, the critical infrastructures of our nations are currently reviewing the next upgrade for their critical infrastructure computer networks. Actually I am willing to bet that nearly all of them are halfway completed migrating to Internet technologies, leaving behind serial lines for today's modern cheap and fast mediums (CAT 6). In some cases they are even skipping the migration to Ethernet, and proceeding directly to highly insecure wireless automation solutions for manufacturing industries.

The cover story of the June 2006 issue of Manufacturing Automation details how a Canadian automobile transmission manufacturing facility redesigned their whole facility this year to use wireless control technologies. Here is an excerpt which details part of the upgrade (automationmag.com):

> *"They connected the server to a floor-mounted switch via a fibre optic cable and connected wireless access points to this switch, along with the PLC. The control devices are simply rugged tablet PCs, with standard 802.11b wireless Ethernet cards."*

I can remember it was only 2001 and I started working with someone who wanted to setup a Wireless LAN (802.11b) at home. He must have bought three different router devices before the technology matured and became usable. Then there is the legacy issue concerning the initial Wireless Equivalent Privacy (WEP) available during the release of some of the first 802.11 hardware. Let us hope that the plant is not connected to the Internet, they are using proper firewall procedure, and that they are using something better than WEP to secure the tablet PCs and wireless access points.

SCADA Configurations

Historically SCADA system configurations looked something like the picture to the left. A Front End Processor (FEP) in the control center communicated via. modem to any Remote Terminal Units (RTU) within the network. The network could be spread-out over the continent in a WAN configuration connected through leased-lines and modems (like in the case of SAGE/NORAD), or the system could consist of a local Fieldbus only (Nuclear Reactor). There are any number of network configurations possible. The most important concept to grasp here is the general configuration. One controlling machine (master) controls one or more slave devices (Chu, Bei-Tseng. et. al).

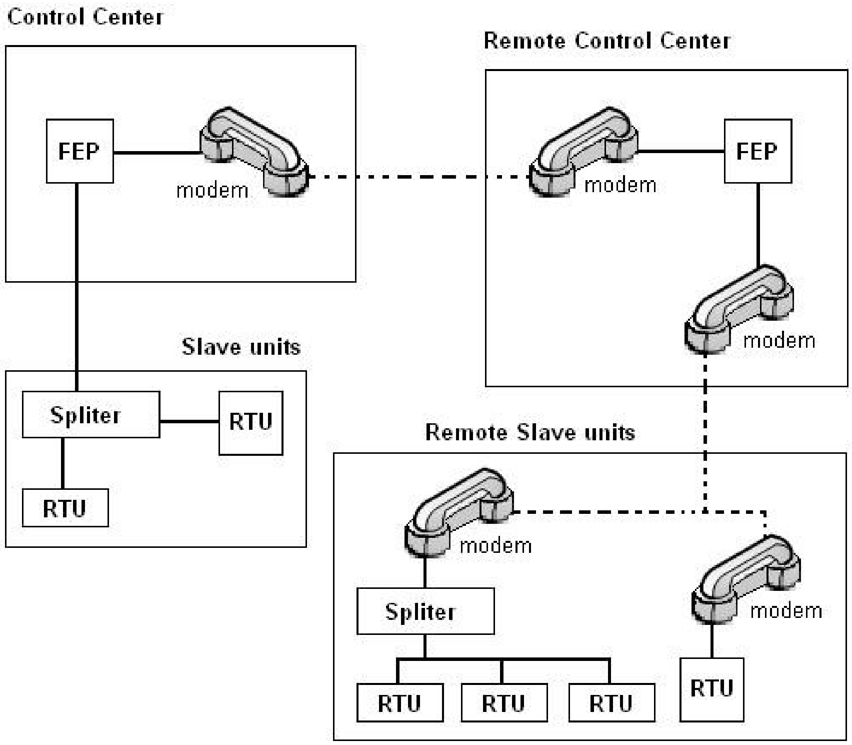


Illustration 9: Typical SCADA networks using a serial cable Fieldbbus.

In a modern day SCADA network instead of a modem a whole host of communications medium, devices, and configurations may be present, including Ethernet, Token Ring, Fiber Optics, Wireless, Microwave, and Satellite. According to the SCADA Honeynet project the following configurations are of interest (Franz, Matthew et. al):

1. Direct serial device: Industrial devices that have a modem which can be directly dialed into from a public phone network.

2. Remote Access Server (RAS): dial-in access via. PPP and password to an Industrial network.

3. Ethernet serial gateway directly plugged into the Internet: A bridge between the IP network and serial network (Fieldbus). The IP side of the device is connected to the network, and the switch or router connects to the serial network controlling the Industrial Devices.

4. Ethernet enabled industrial device: A device connected to an Ethernet bus inside the PCN.

5. A router directly connected to the Internet: PCNs are typically not directly connected to the Internet, but it is possible for there to be connection from the PCN to the corporate network which does have a direct connection to the Internet (hopefully with firewalls in between).

6. Wireless: Most of the Industrial wireless devices use proprietary wireless protocols, but some of them use 802.11b standard. A wireless bridge is typically used to connect to the serial device.

7. Remote desktop access and HMIs: The Human Machine Interfaces and the software that communicates with Industrial devices usually run on a Windows machine. Administrators who want remote access to these devices typically run a remote desktop viewer, such as VNC or PC anywhere. This would show up in any successful attack scan and needs to be considered.

In 2004 the United States General Accounting Office (GAO) prepared a report to document the Challenges and Efforts to Secure Control Systems (GAO-04-628T). Here is how they illustrated current and legacy PCN networks:

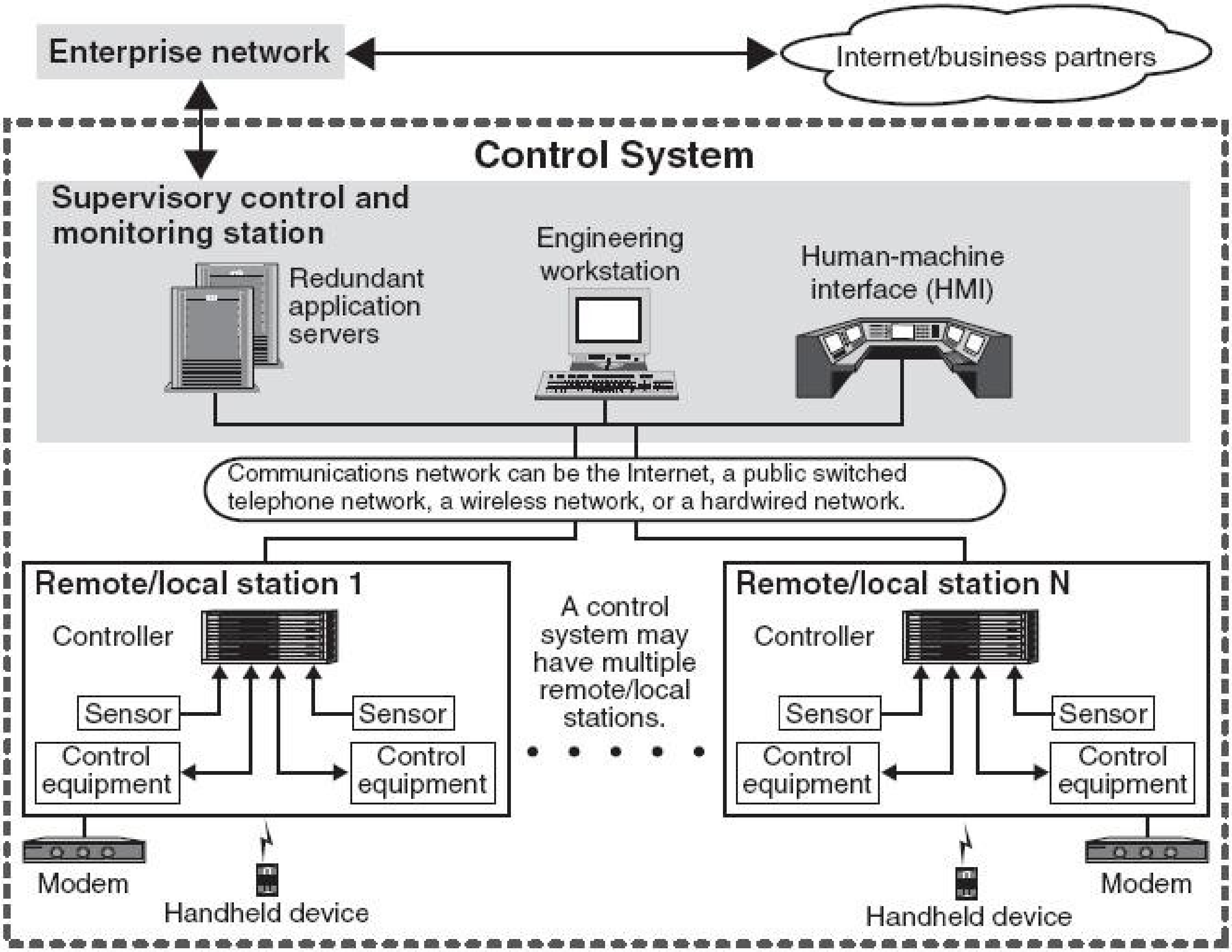

Enterprise network
Internet/business partners
Control System
Supervisory control and monitoring station
Redundant application servers
Engineering workstation
Human-machine interface (HMI)
Communications network can be the Internet, a public switched telephone network, a wireless network, or a hardwired network.
Remote/local station 1
Controller
Sensor
Sensor
Control equipment
Control equipment
A control system may have multiple remote/local stations.
Remote/local station N
Controller
Sensor
Sensor
Control equipment
Control equipment
Modem
Handheld device
Handheld device
Modem

Illustration 10: Ways a PCN can be configured (note the multiple forms of data communication).

Notice that custom networks, telephone networks, and wireless networks may be involved. Additionally satellite, microwave and many other network mediums and protocols could be involved. With the recent push to Internet Protocol in SCADA systems the Modbus (modbus-ida.org) and Profibus (profibus.com) protocols have been revised to allow for an Ethernet network architecture. The end goal of each architecture appears to have started under the presumption that implementing the new protocols would allow each protocol to provide an application layer for integrating with modern Distributed Object Models (DOM). Essentially these protocols would allow the use of common SOA architectures like CORBA, SOAP, and DCOM. The Common Industrial Protocol (CIP) initially created by Rockwell Automation, has also created a similar Internet Protocol extension to their network protocols called Ethernet/IP™. The following diagram illustrates how they view PCNs and their position within an Enterprise network (rockwellautomation.com):

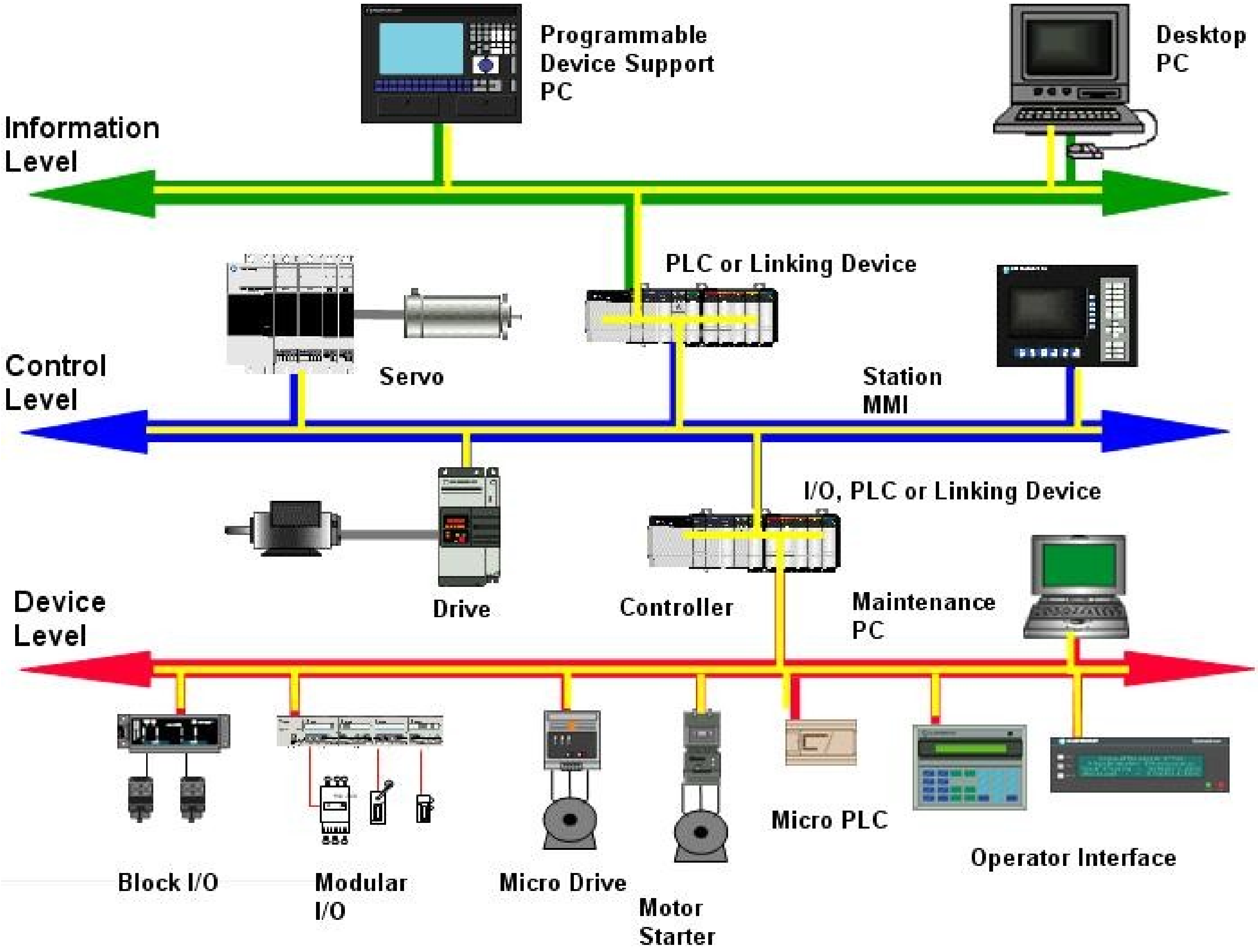


Illustration 11: The PCN as seen by the architects of Common Industrial Protocol (CIP).

This picture fully illustrates the multiple network segments of a PCN, and their potential connection points. It is important to again note that SCADA systems represent an abstract network configuration, that could comprise many different network mediums, protocols, and configurations depending on the needs of any particular organization. Notice the three types of network level segments pictured. Each level has the potential to use a different protocol, however some configurations only use one protocol for each level. The number of protocols in use does not necessarily protect this network, since as we can observe the whole network is still connected to the Information Level services, which in turn have a direct connection to the Internet.

SCADA Vulnerabilities

A leading researcher in the field of SCADA system security Eric Byers notes that before the advent of the Internet net most SCADA system faults and security incidents and breaches, occurred from entities within corporations. Starting in the time period after 2001 there have been an increased reporting of incidence coming from external sources such as the Internet (Byres, Eric et. al). I think it is also beneficial to group these incidences according to Direct and Indirect occurrences, since by merging PCNs with a social network like the Internet, there are Direct and Indirect consequences.

Indirect Occurrence

The potential to see indirect consequences in SCADA systems due to Internet Worms is recently been shown. The SQL Slammer Events occurring on an Asynchronous Transfer Mode (ATM) network saturated the provider's network. ATM networks were created in order to provide a network protocol that could multiplex many other protocols over an abstract amount of bandwidth. The goal was to provide a highly reliable and extensible network backbone for multiple data networks. In this case an electrical utility company was sharing the ATM network to transfer SCADA system data over the ATM network. The company was not even using the Internet for communication but had been using a Frame Relay service with no interface to the Internet, however when the ATM network was overloaded by the SQL Slammer Worm, the Frame Relay service experienced an increased delay in its SCADA data communication (NERC, 2003). This presents a very dangerous indirect way in which the Internet can affect SCADA systems. It was advised after in the NERC report for companies to ensure they review their Quality Of Service (QOS) contracts with network providers, to ensure that a redundant QOS is allowed for their time critical SCADA networks.

Essentially organizations using SCADA systems have to ensure that if one system or network becomes unavailable a secondary or tertiary network is available. This incurs added expenses to their operations and most likely a higher cost for consumers. With a situation like this it is highly unlikely that a company will upgrade their systems ahead of time. In an economical situation like this it is more likely that they let the situation become known to the public through publications of actual accidents, than become known as the first company to impose security expenses which drive up the associated cost and increase prices in a consumer competitive market. However if companies plan ahead and become vocal on just how insecure these technologies we are now all using truly are, it would be a good thing for everyone.

Illustration 12: Interpretation of an Internet Worm

Direct Occurrence

As we have mentioned above Ethernet/IP™ has proposed to use a Multicasting protocol for establishing a message bus for part of the network in its new Ethernet configurations. However only two years after these white papers were released from Rockwell Automation, the CIAG at Cisco discovered that the multicast protocol has inherent problems at the switch level making DOS attacks quite feasible. As the study explains if the DOS attack was implemented as a Worm its growth and spread on the Internet might have unknown consequences, since the spread of Internet Worms has no known bounds until the worm can be examined. The SQL Slammer worm in January 2003 infiltrated a electricity sector internal network, it slowly migrated through the corporation until it entered a critical SCADA network via. a remote computer through a VPN connection (NERC, 2003). The propagation of this worm then blocked the legitimate and time sensitive packets being routed on the PCN. No information from the NERC report mentioned the net incurred damages, however when blackouts in the electrical and power sectors occurs loss of life is often not far behind.

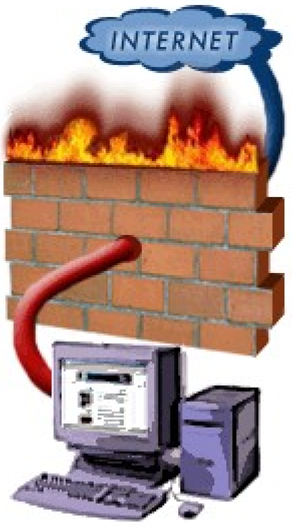


Illustration 13: Firewalling

As the above pictures imply if something goes wrong in a nuclear reactor because of network problems a major catastrophe could result. A security research report in 2005 detailed the numerous dangers SCADA systems are being presented with, because the increased use of unsecured Windows machines. In addition the advent of PCNs moving their SCADA systems to consumer standard networking equipment making increased use of Ethernet and Wireless, there are increased security risks (Fernandez, Andres E. et. al). This has the potential to cause a major catastrophe if Worms are developed that can pass through the appropriate corporate firewalls (Assuming the organizations are using the appropriate firewall settings.). We have shared that traditionally SCADA systems have been the responsibility of the government, scientific and security communities. Extending such systems into a public/social network like the Internet (which is much less controlled than when AT&T had their monopoly of the phone industry), seems only to reduce the level of implied security in the system.

However if your company uses substandard security no matter how secure you make your Internet connections entities may find a way around these precautions through other methods. In short you can never under estimate your enemy, however it is also important to keep a good balance. In the case of the sewage plant attack in Queensland, what sort of security precautions were involved in hiring a contractor? Hiring a contractor for a SCADA system can be potentially disastrous in a time of market sector down turn. Organizations must either plan ahead and not allow the confusion of higher contract rates to deter them from following a strict procedure, or agree to grow internal experience instead of relying on outsiders. Having a good corporate ethos might be another method of defense. Then there is the subject of publicizing your private information. Corporations who make known their operations publicly with consumer orientated pamphlets etc. should be cautious to keep from saying more than they should. Internal company documents should be stored behind an encrypted channel not accessible to the general public. In the next section we detail how ignoring simple security guidelines poses a risk.

II. Project Description

We have already discovered that a SCADA system can be a very important network to secure, in order to keep national critical infrastructures safe from harm. It is a good thing that there is so much concern and focus currently on this topic as of late. I have defined in the previous section what a SCADA system is and how PCNs are typically configured. Then I described some of the current weaknesses and potential vulnerabilities of these sites, as we are moving further into the 21st century. In this section I will focus on what is currently being done by SCADA related organizations and discuss some of the solutions to current site weaknesses. Then I will describe the nCase™ security solution for securing SCADA systems.

SCADA Security

After reading many various documents about SCADA systems and vulnerabilities it was discovered that the Group for Advanced Information Technology (GAIT) created a document to define Good Practices on deploying firewalls to secure PCNs. The document was created for the National Infrastructure Security Co-ordination Center (NISCC) in the UK, and was produced during the time that Eric Byres was working at the Internet Engineering Lab (IEL) at BCIT (GAIT, 2005). This document describes in detail some Good Practices for providing firewall security for SCADA systems, given the level of risk and implementation details of each site. Essentially tri-homed DMZ firewalls are recommended between the PCN or SCADA system and the enterprise network (If there needs to be a connection to the enterprise network at all.). A tri-homed configuration allows for a local data historian machine to be accessed from both the PCN, and the enterprise network. A picture of the configuration taken from the document (GAIT, 2005), explains the situation better than words:

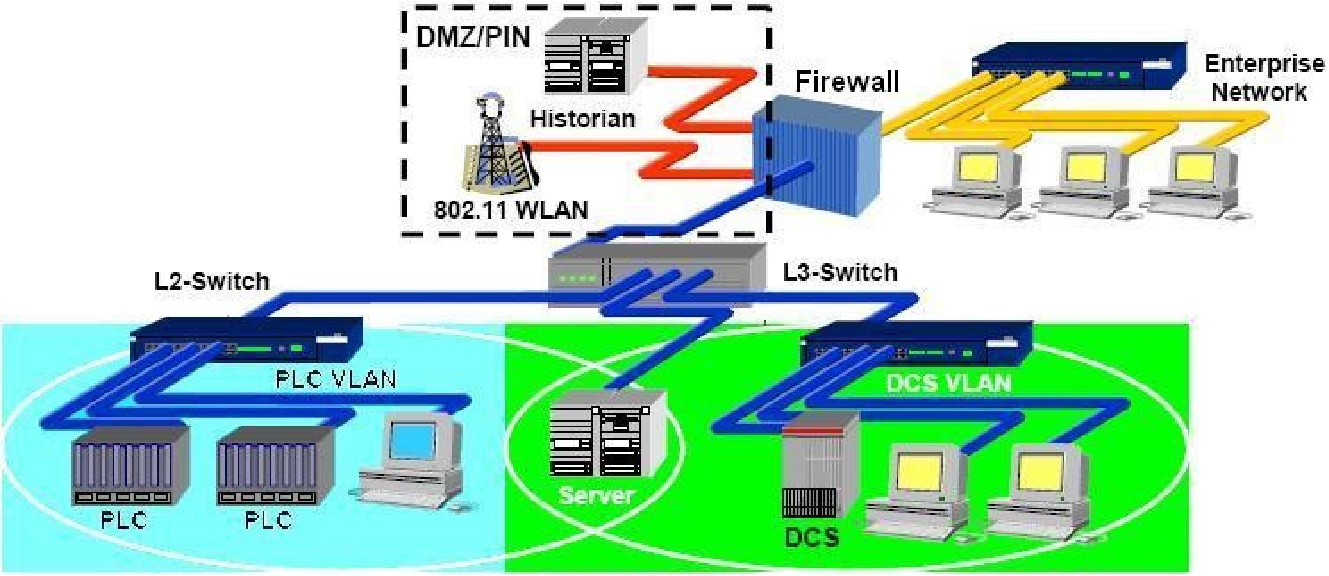


Illustration 14: How to secure a SCADA system that provides Information to the Organization.

This document was followed up by a document produced by the PA Consulting Group (UK) for the NISCC, which details some Good Practices for PCN and SCADA security. The PA Consulting Group document reiterates most of the issues uncovered by the document produced by GAIT, but reduces the desired business flow of information into rules and regulations (NISCC, 2005), for providing secure business practices. It is a good document that will help to establish good security procedures in PCNs which currently have poor or inadequate security practices (like the YWD situation mentioned earlier in this document). It gives some detail about how to establish proper procedure for allowing designated remote access to PCN devices for the support organizations which work for the PLC device manufactures. Given all these guidelines and our knowledge of SCADA architectures, our aim has been to develop a software solution that will provide for Designated Access Devices (DAD). Essentially the nCase™ solution has been designed to provide for DADs. More specifically the circumstances in which corporations need external party access to PCN devices, is a circumstance that requires a DAD.

The 2005 demonstration of how to hack into Cisco System routers running the IOS operating system at a Blackhat conference alone, is a good enough reason to consider Ethernet technologies connected to the Internet as a less than safe practice. If we add to this the fact that fixing defects (bugs) in current SCADA related software and devices is blockaded on nearly every side (digitalbond.com), we are at a critical point in SCADA vulnerability. Approximately one year ago Venkat Pothamsetty of the CIAG at Cisco Systems wrote a paper titled “Where Security Education is Lacking”. The document details how the lack of focus on security education instruction has created working individuals with little awareness concerning secure software design. The article goes on to specify what percentage a typical software course should focus on security matters relating to software. It explains that the way core Computer Science courses are being taught must change, to also instruct students on a full breadth of secure software knowledge. To conclude the paper he explains that only by changing the educational system will the number of vulnerabilities being introduced in to software be reduced (Pothamsetty, Venkat).

This is fairly alarming information and with information like this we see the task of securing SCADA systems using Ethernet technologies as paramount. To this end the following rules apply to securing a device that operates in a PCN:

1. Devices used should not be allowed to communicate with any devices other than their local PCN devices, using the appropriate destination ports, services, and Communications states.

2. No other devices (enterprise, local, or remote) should be able to communicate directly with SCADA devices, except by using the planned communication states, services and devices operating on the PCN (Essentially the system is completely locked up.).

3. No wireless access point connections are permitted to access the devices unless a proprietary protocol and encryption (WPA or better) is being used. In other words 802.11 networks that are provided as a convenience, are not allowed access to the SCADA devices or control networks.

4. Designated connections may be permitted between the designated machines and the SCADA devices for short periods of time, only if the proper local PCN procedures are carried out.

5. After security procedures are carried out and documented, a special program operated by a PCN worker may be invoked to open up a designated access path to a device or machine.

Security Solution

In one way it would have been nice if SCADA systems had been upgraded to a completely separate privatized network solution, however we now have to live with Industries that follow at the tail-end of consumer developments. The nCase™ deliverable of this project will ensure that SCADA devices on a PCN will no longer be open to Internet attacks. In saying this we have to explain straight off that the firewall rules of any PCN will be specific to each site. The access rules given above should be implemented in the firewall(s) that protect PCN networks. We should mention here that the computer security industry is an evolving field and no Silver Bullet (Brooks, F. P.) solution can possibly solve every issue. The goal behind the nCase™ design is to transform a device that is open to attack, into a Designated Access Device (DAD). To this extent the deliverable of this proposal will be to implement a set of firewall rules that meet the for-going set of rules. The following picture illustrates the before and after impact of nCase™ on a PCN:

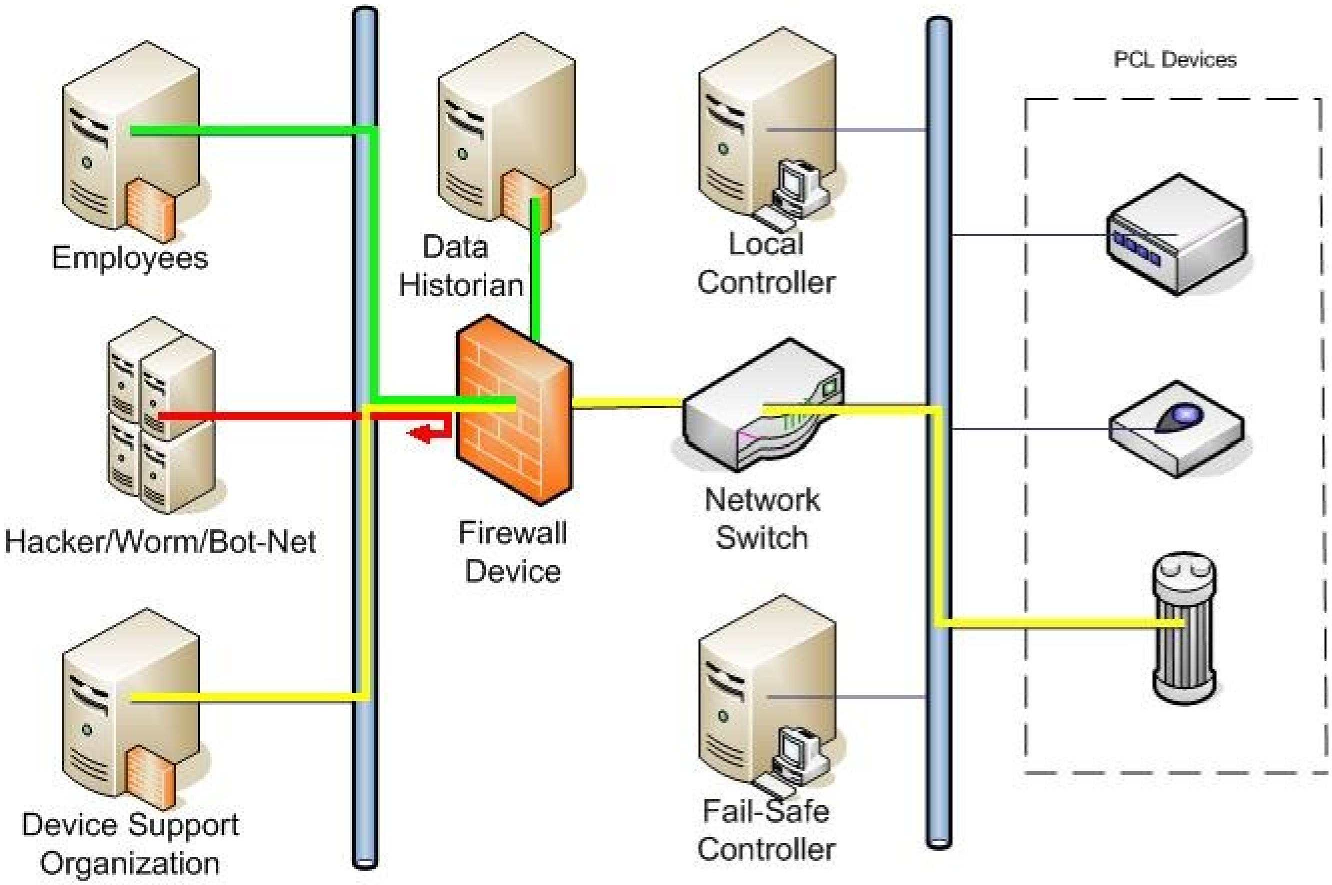


Illustration 15: After nCase is installed Support Organizations have Designated Access

As can be seen above employees in the organization have remote access to the PCN data. Secondly without a firewall protecting the PCN, it is likely that hackers and worms could find a way into the PCN and cause disastrous circumstances (even by mistake). However working around the firewall, allows external support organizations to work on the PLC devices remotely. Currently there are no known facilities which provide for this type of detailed secure remote access. The nCase™ solution will allow employees to provide remote access to the PCN through Designated Access Controls (DAC).

An analysis of the above functionality and set of rules and comments may elicit the following response from a trained security professional, “How are you going to use a firewall to allow designated device access?”. This is a good question. The answer is we are not going to use a firewall to provide this access directly. nCase™ includes a DAD Packet Inspection Daemon (PID) which listens to Internet traffic on the firewall device it is installed on. When a DAD initiating packet is sent to the firewall this PID will perform the required operations on each packet, in order to translate what kind of access is required to the device (The level of access granted may not always mean direct external access, the purposes of a support organization might also require that the device contact the remote organization. Whichever connection parameters are required for the device access, these parameters will be provided.). When the required access has been determined the PID will adjust the firewall settings to allow for this connection to pass through. The operation of the PID on the firewall machine can be demonstrated as follows:

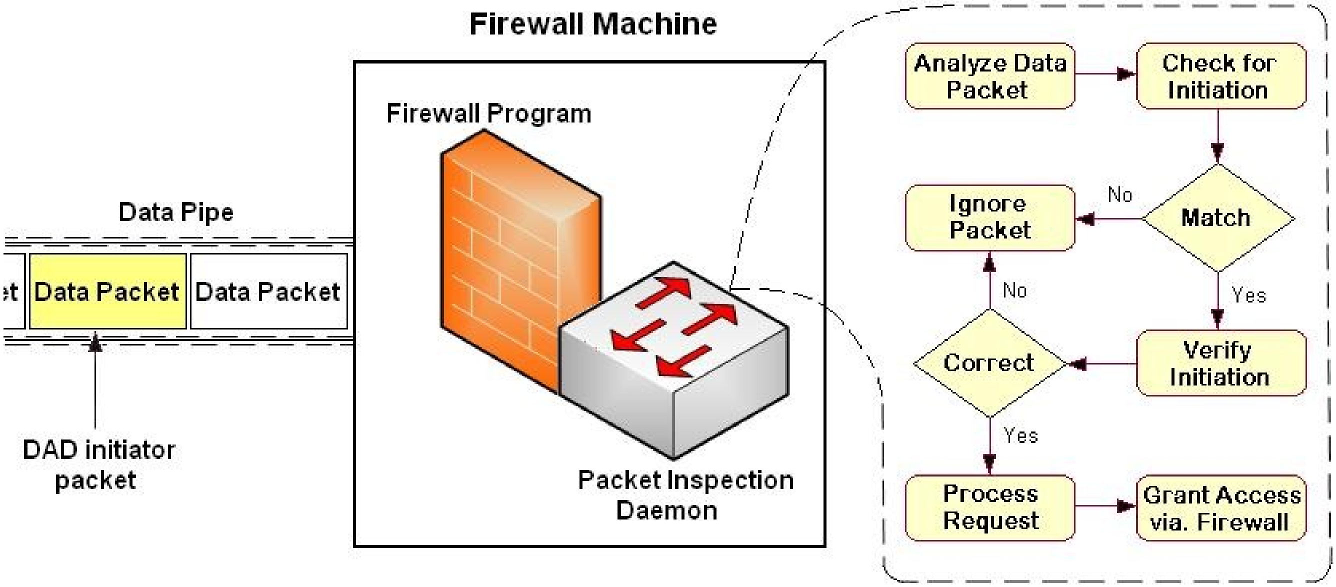
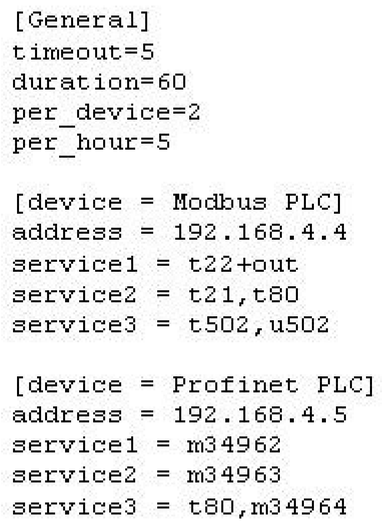


Illustration 16: The Packet Inspection Daemon analyzing packets to detect DAD initiators

The above picture shows the external network connection to the firewall machine. This machine's sole purpose is to direct network traffic by analyzing the received network packets, against a set of rules that govern what data should pass in and out of the device between networks. In the instance shown above an DAD initiator packet is sent to the firewall. The PID program analyzes the network data as well, except it is looking for specific packets which signify a DAD initiator packet. When an initiator packet is received the PID applies a verification process to check if the initiating packet is requesting DAD services. If the verification process succeeds the request is processed. The exact type of access required is ascertained and then the PID program applies the appropriate firewall rules which facilitate this access. In this way the firewall is manipulated to facilitate designated access to the devices under its supervision.

The PID program will have several adjustable configuration settings. One of these settings will specify the timeout period in which the remote entity has to make its connection to the device or vice versa. Another configuration setting will specify the maximum duration for any DAD connections. A third setting will specify how many designated access connections will be allowed to a single device at the same time. A final setting will specify the number of connections that can be initiated per hour.

Finally it is perceived that each device may have specific services which are required for a remote access. During the installation of nCase™ these services will become obvious. It is desirable that administrators of an encased PCN network be allowed to configure each nCase™ device according to the devices under its control. To this end the specification of device specific configuration information will be possible via the configuration file. An example configuration is given on the right. Notice that a device can be identified by its IP address and by its device name. The services that are common to a specific device can be specified, and it is possible for a service to require more than one communications port. The descriptors t, u, m refer to the transportation protocol required (ie. TCP, UDP, or Multicast/UDP).



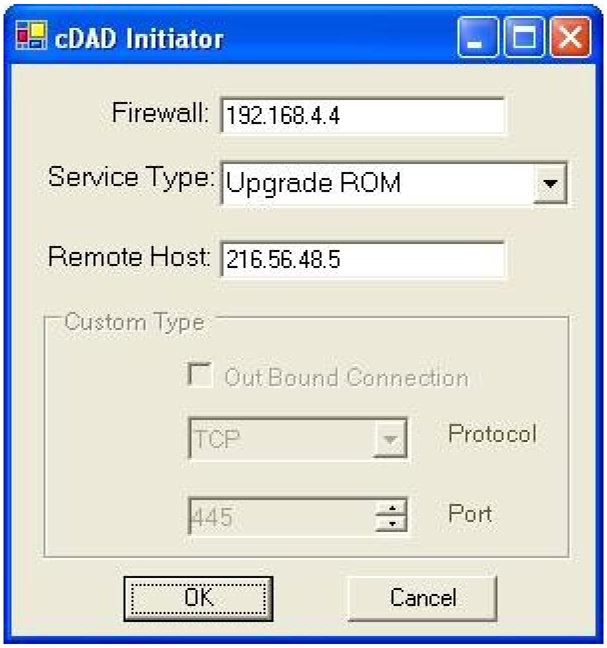
```
[General]
timeout=5
duration=60
per_device=2
per_hour=5

[device = Modbus PLC]
address = 192.168.4.4
service1 = t22+out
service2 = t21,t80
service3 = t502,u502

[device = Profinet PLC]
address = 192.168.4.5
service1 = m34962
service2 = m34963
service3 = t80,m34964
```

Illustration 17: A PID configuration file.

As shown above there are multiple configurations possible for DAD connections. A user that wants access to a specific feature-set of a device, can choose one of these services when sending an DAD initiator packet. For instance a device might have an interface for downloading a new ROM image for burning to flash memory. The service2 entry of the "Modbus PLC" device from Illustration 23 consists of TCP ports 21 and 80. Perhaps in this scenario port 21 is used to transfer the binary ROM, and port 80 is used to access a web interface for launching the flash process. In order for this service operation to be completed, both ports 21 and 80 need to be opened. This is the main reason service types have been specified (ie. Users sometimes have specific reasons for accessing a device.).



Firewall: 192.168.4.4

Service Type: Upgrade ROM

Remote Host: 216.56.48.5

□ Out Bound Connection

TCP Protocol

445 Port

Illustration 18: Initiating DAD client application.

DAD initiator packets are created and sent to the PID program from a client application called cDAD (short for DAD client). The cDAD program has the ability to contact as many PID enabled firewall devices as necessary. When operating the cDAD program a user must enter the IP address (or DNS name) of the firewall machine, and specify the service and remote host (ie. IP address) of the allowed connection. Optionally a custom service declaration may be specified. A custom service declaration will require the user to enter the protocol and port number of the connection to be allowed on the fire wall device. Optionally during a custom service specification the user can specify if the connection will be initiated from the device (ie. Outbound). Illustration 24 to the left details what the client application might look like.

III. Technologies

In order to complete the nCase™ solution a minimum of two platforms are necessary to host both the client and server applications. First we detail the server implementation, then we discuss the possibilities for a client application.

Server Specification

To implement the demonstration prototype we have purchased an embedded microprocessor board from Intrinsyc, Inc. called the Cerfboard 270. The Cerfboard 270 uses an Intel® Xscale® PXA270 processor (system on a chip design), which makes use of a RISC based instruction set from Advanced RISC Machines Limited (ARM), version 5. The device has been custom made to provide a reference PDA type platform (416 Mhz). Essentially any size processor or system could be used to host the PID server application. However the embedded processor board was chosen for its small size to demonstrate that this service could be hosted in any level of the PCN network. Device manufactures might even consider adopting the nCase™ solution in the devices themselves using an on-device firewall.

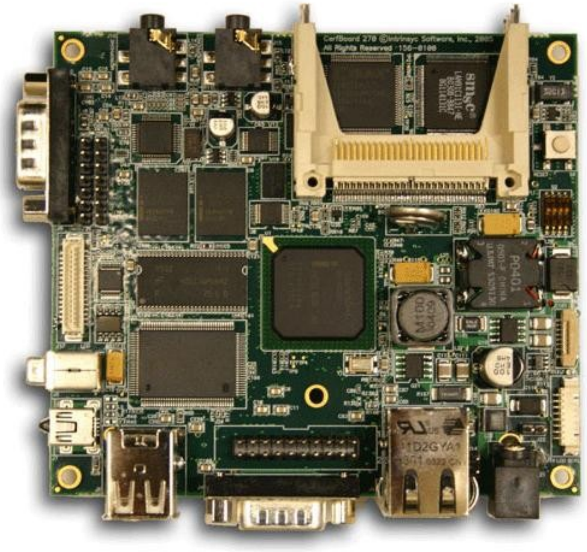


Illustration 19: DAD Server implementation hardware

The Linux operating system will be installed on this board using a Linux kernel version 2.6.14-cerf1. Due to the specifics of the embedded board's manufacture, a custom version of Linux called iLinux v5.1 is installed. The Linux operating system is well suited to this application since it allows for a detailed manipulation of the base data communications protocols. We are unaware if this level of detailed protocol manipulation is possible on the Windows platform. Until a raw socket API is made available for Microsoft Windows it is doubtful that it will be possible to implement any part of the application (client or server) on a Windows OS. Other forms of Unix may be supported in the future. The Packet Inspection Daemon (PID), device utility programs, and kernel will be cross-compiled especially to meet the above reference architecture. A Linux desktop computer which makes use of a similar version of the Linux Kernel and GCC version 3.3.3, will be used for all cross-compile actions.

Through one of our associates we have found a non-proprietary Packet Inspection Daemon (PID) called BeachHead™, that will facilitate all PID purposes in the proposed application. BeachHead™ is merely a reference PID application it contains no user configurable customizations. Thus the BeachHead™ program will need to be radically changed. The program also uses a very poor method of encrypting the application data being sent from client to server. The current security scheme for packet encryption does not encrypt all the data being sent. Without this encryption the custom encoding the BeachHead™ program uses to encode/decode each packet, will be a non-random piece of information that hackers could easily use to mimic the protocol, and this has already been proven in a bug report, and reproduction attempt. To repair the faulty encryption protocol and introduce device specific configurations the application will need to be completely redesigned.

While some portions of the One Time Pad (OTP) encryption technique will be preserved, the redesign of the encryption protocol, will allow for the program to inter-operate with proprietary encryption protocols like Turret™ (Which has not yet been fully designed. However a reference architecture plan has been included in Appendix B : nCase Family of Associated Programs). Turret™ is a custom encryption technology that makes use of several different methods to produce significantly randomized data. The reason a random OTP is perfect for this implementation, is that the size of the data that will be transferred from client to server is significantly small (using only basic TCP header fields). If specific OTP data can be generated ahead of time both the client and server will be able to use what could be called a completely unique non-repeating language in which to communicate. It has been mentioned that cracking truly random OTP encryption mechanisms, is almost impossible (wikipedia.org).

Illustration 20: Firewall client Ethernet port

The firewall program used on the nCase server device will be the iptables program that is the precious work of the Netfilter Core Team (netfilter.org). The iptables program is the result of another hard-working quality-driven Open Source development team/collective. The iptables program is an interface for configuring this entirely kernel based network firewall. It is more than adequate for implementing the required firewall. However here we need to note that the Cerfboard 270 currently only provides one Ethernet port, and one USB hub. To this end the USB port will be used for providing an extra Ethernet port, using the CATC NetMate USB Ethernet Link product. Thus the resulting firewall will not be a tri-homed firewall as recommended by leading researchers in the field, but a dual-homed firewall. While a tri-homed firewall is feasible the current hardware will not support it. Even so a dual-homed firewall provides ample security for the basic PLC scenarios considered in this proposal.

Client Specification

The client program which sends the DAD packet to the firewall for the purposes of this project, will be compiled to run on the Linux x86 architecture only. The client program will be a command line program and may possibly include a GUI interface. As mentioned in the Security Solution section the client application will allow the user to specify:

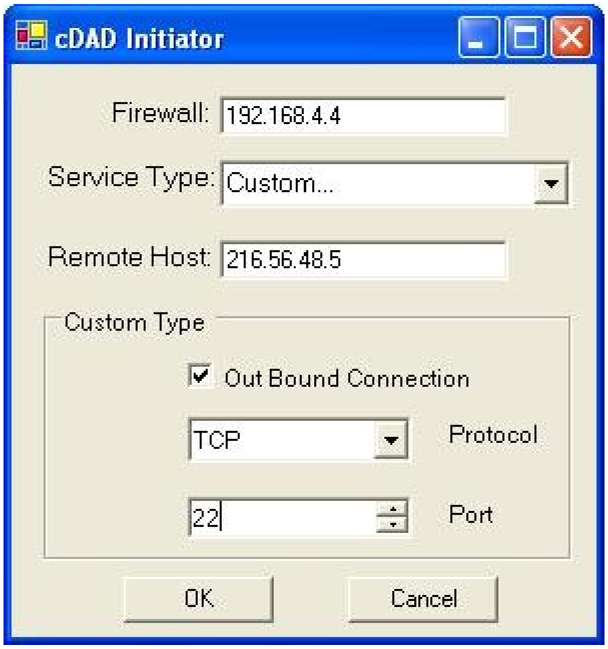
1. A firewall host name (or IP address).

2. A service to open on the firewall.

3. An IP address to allow through the firewall

4. If a default service is not given the user will specify the transport protocol, port, and whether or not the connection is initiated from the PLC device (ie. Out-bound. The default is inbound.).

In the future it is foreseeable that the client application be hosted by an Internet server of some kind (ie. Web or Application Service), however for now the focus of this project is ensuring that the project is feasible with the given hardware. To this end a description of the cDAD application service has been detailed in the Appendix B : nCase Family of Associated Programs. In the application service paradigm the actual application server resides in the DMZ with the data historian. In this way the hosted application is physically located within reach of both the internal PCN and external network. It also keeps the OTP data within the confines of the DMZ and PCN.



Firewall: 192.168.4.4

Service Type: Custom...

Remote Host: 216.56.48.5

Custom Type

☑ Out Bound Connection

Protocol: TCP

Port: 22

Illustration 21: User provides a custom service.

While an application service would provide an easy always connected option for remote users who need to establish a DAD connection, it also provides this service to anyone with the ability to connect to your Web Application. While the service could make use of SSL for encrypting the data allowing for only the restricted user base to connect to the device, often application servers are very complex configurations to maintain. They are also subject to numerous different computer security risks because of the large amount of source code they are dependent on, and the amount of access they provide to external users. If not configured correctly these servers could be compromised, and given the numerous exploits seen over time concerning these applications, it is highly likely a new exploit could arise at any time. Given this fact it is a better idea to only provide the service as a non-web based utility program.

Given the fact that the client program interface provides for a user to create a DAD request for a custom defined service (As shown in Illustration 27), it may be in the best interest of an organization to only allow preset services entries for devices (In order to rule out social engineering or circumstances where an Internal security breach has occurred.). In this case another configuration item will be made available to prevent client DAD request customization from being authorized by the server.

IV. Innovative Component

The concept of using a PID program to remotely manipulate a firewall machine is the main innovative component of this project.. However the particular nCase™ solution we are implementing according to current research reports on the Internet, is one which is only now starting to receive wide-attention in the SCADA security industry. The following comment was elicited by a PCN operator when he received word of the development of a product similar to nCase (Peterson, Dale):

> *"There is a need for PLC level protection to augment (not replace) SCADA/Control system border protection. Making this device look and feel like an I/O device will ensure the field technician can deploy this easily."*

To date we only know of one company producing a competing product, and the product is not expected to be released until Q2 of 2007 (mtl-inst.com). The press clippings which announced the potential product originated on October 24, 2006, so the concept itself is emergent and innovative.

In the state that we received the reference PID program it is not sufficient to implement the desired functionality proposed in the nCase™ solution, since their are notable weaknesses. Many weeks of work will go into fine tuning the application for use on an embedded board, as well as adding custom options to provide for the full remote access functionality. The source code is littered with security defects which will need to be resolved if anyone is to purchase the end product with confidence that the BeachHead™ itself does not succumb to attack. To our knowledge programs like BeachHead™ are usually only installed or created by organizations as a customization to some aspect of their application, that requires private and secure remote access. Creating such programs are nothing particularly new in the software business but are very dangerous facilities to provide, given the level of remote control they allow for. In general however each customized remote access utility is different. The part that especially makes this application innovative is the TCP header encoding and encryption technique.

In terms of personal innovation this project is a personal landmark. I had not worked with embedded processors until venturing on this project (Though it had always been a dream.). Learning about PLCs, SCADA systems, PCNs, and their associated protocols has been a long but very informative process (Matter of fact without doing an in-depth research report I would not have truly understood the domain knowledge required in order to properly propose this solution.). Sometimes innovation can be considered as using old concepts in a new and cutting-edge way. This is the way I look at the nCase™ solution.

V. Project Scope

Based on the project description and specifications discussed above there are two functional parts that comprise this project, the client application (cDAD) and the server application (PID). In the following section we detail exactly what is required for each of these parts of the project. These descriptions are detailed and cover the breadth of what will be delivered in the end product. Finally at the end of this section we elaborate on what was left out, why, and how these features are less important than the client and server applications (ie. unrelated to delivering a fully functioning product).

Client Scope

We will deliver a client application (cDAD) which allows a user to communicate with a firewall device, which Designated Access Device (DAD) connections should be allowed through the firewall. To this end the application will allow a user to specify the following parameters, for each proposed connection:

1. A target firewall host name (or IP address), to request the DAD connection.

2. An IP address to allow through the firewall.

3. A service to open on the firewall. Each service is associated with a group of protocols and port numbers that are required for opening the connections required by the service (ie. Upgrade ROM). These services will be predefined for each nCase™ implementation.

4. If a default service is not given the user will optionally be able to specify a custom service to open. A custom service will include the following information:

   - The transport protocol to use (ie. TCP, UDP, or UDP Multicast).

   - The port number that will host the connection on the end device.

   - A boolean specifying whether or not the connection is initiated from the PLC device (ie. Out-bound. The default is inbound.).

5. An appropriate encryption mechanism for encoding and encrypting the DAD initiating packets. The design will be modular to allow the use of different cryptographic libraries.

6. An end user guide that describes how to use the client application for its intended purposes.

7. Time permitting the client application will be implemented in the form of a X-Window GUI application (Using either Lesstif, GTK, Gnome, or KDE graphical tool kits.).

It should be stressed that these connections are non-standard in their origination, in that they allow an external computer to connect through the PCN firewall to a device inside the firewall. In this way the connections themselves are temporary and may involve time limited durations. The main reason for the time-limits are to minimize the duration for which the external party accesses the devices of critical importance to an organization (In the case of SCADA systems authorized access only, is central to promoting security.). It is expected that the connecting computer will use the appropriate encryption protocols for obfuscating the communication into the PCN network, so the connection data cannot be intercepted by unexpected on-lookers. The parameters of the allowed connections will be specific for each firewall and organization. It is expected that each nCase™ implementing organization will plan a set of services to offer ahead of time, or make this known to our consulting company before installation.

Server Scope

We will deliver a fully functioning firewall device called nCase™. Configuration of the embedded device will include:

1. A customized Linux kernel that will allow for the correct operation of a firewall enabled device

2. Associated embedded packages, programs, and drivers that will allow for the correct administration of the firewall enabled device, and one protected Ethernet hosted device (computer).

3. A set of firewall rules that provide the following features of a dual-homed firewall:

   - Devices used should not be allowed to communicate with any devices other than their local PCN devices, using the appropriate destination ports, services, and Communications states.

   - No other devices (enterprise, local, or remote) should be able to communicate directly with SCADA devices, except by using the planned communication states, services and devices operating on the PCN (Essentially the system is completely locked up.).

   - No wireless access point connections are permitted to access the devices unless a proprietary protocol and encryption (WPA or better) is being used. In other words 802.11 networks that are provided as a convenience, are not allowed access to the SCADA devices or control networks.

   - Designated connections may be permitted between the designated machines and the SCADA devices for short periods of time, only if the proper local PCN procedures are carried out.

   - After security procedures are carried out and documented, a special program operated by a PCN worker may be invoked to open up a designated access path to a device or machine.

4. A PID server program for inspecting data packets sent to the firewall for establishing DAD connections. This program will provide:

- An appropriate decryption mechanism for decoding and decrypting the DAD initiator packets. The design will be modular, allowing multiple different cryptographic libraries to be used, by client and server.

- Appropriate configuration settings to ensure the nCase™ system is only used in the appropriate circumstances. These configuration setting will include the following Device Access Controls (DAC):

    - Timeout: An initiation timeout value detailing the duration (in minutes) for which a client can connect through the firewall after a DAD initiator packet has been received (ie. After the DAD packet has been approved, the client has a limited window in which to start the actual connection to the device hosted by the firewall service.).

    - Connection Limit: The number of service connections per device at any one time may be limited by setting this to an appropriate integer value.

    - Hourly Limit: The number of service connections per hour will be limited to the integer value present in this setting. This provides for organizations which have in depth security practices already in use.

    - Customizable: A boolean setting specifying whether custom connections can be specified by a client, or whether only predefined service connections will be allowed.

    - Duration: A connection duration setting specifying the time in minutes for which a connection will be allowed to operate (This might prove to be over-kill seeing that each device access will require a DAD initiating packet. This feature might not be implemented or scrubbed if it proves of little use to end users.).

    - Device Specific: On a device by device basis the above configuration settings can be over-ridden. Additionally device specific services can be defined in this section. A device specific service includes the following information for each connection that is to be allowed:

        - the transport protocol to use (ie. TCP, UDP, or UDP Multicast)

        - the port number that will host the connection on the end device

        - A boolean specifying whether or not the connection is initiated from the PLC device (ie. Out-bound. The default is inbound.).

Future Scope

While planning for the design of the nCase™ solution, certain aspects of the end product entered considerable aspects of the research process, and were considered as possible candidates for including in the end solution. The time and budget allotted did not allow for their full discovery. Thus since our knowledge base does not include full knowledge surrounding these items, they cannot be included in the project. However we detail purposed ideas in this section so that you might know the future directions the nCase™ solution may take. These possible projects are further depicted in Appendix B : nCase Family of Associated Programs.

Turret™ Encryption

Implementing the security data provision portion of the project (Turret™), will need to be delayed as the embedded device does not have enough functional USB ports to attach a Turret data acquisition device to it. Due to space limitations on the device, it also does not make sense to include the security data acquisition portion of the project, since the small device could easily run out of space for holding the cryptographic data. The security data acquisition service will not be implemented in this portion of the project as a proper method of delivery to the nCase firewall device and client has not yet been designed. It is conceived that if static storage space is expanded on the device, a secondary Cerfboard device or device could be implemented to upload the data to a firewall server. We have other ideas for this facility but they have yet to be properly researched and explored. Unresolved issues relating to obtaining the following information prevent further consideration:

1. Automated digital color photographs of naturally occurring random events. Current ideas are unrefined and require researching suitable automatable devices and host locations.

2. White noise intercepting filter devices for charting random white noise events. No research has yet been achieved to this end.

3. An algorithm for calculating the randomnicity present in the above data. This would be used to detail the success rate of the end encryption scheme.

SecureTransit™

While an application service would provide an easy always connected option for remote users who need to establish DAD connections, it also provides this service to anyone with the ability to connect to the Web Application. SecureTransit™ uses SSL for encrypting the application data allowing for only the restricted user base to connect, and request DAD connections. Since application servers are very complex configurations to maintain, SecureTransit™ will make use of a reliable and security conscious web server such as the Apache Web Server. Given the numerous exploits seen over time concerning web servers in general, it is highly likely a new exploit could arise at any time. SecureTransit™ will only be licensed to institutions that sign a security waiver concerning their internal security procedures and liability. This waiver will require that program security updates be installed as they are made available, and require IS staff (or an in house security expert) to stay abreast of security alerts.

VI. Methodology

In this section we detail the types of methodologies used to orchestrate the completion of this project. There are quite a few methodologies that have been used so far. To this end in the following sections we discuss the methodologies used relating to our research and requirements, project management, and software development.

Research and Requirements

The primary methodology used to perform the research for this proposal has been Secondary Data Collection and Technological Research. This research has been necessary as the solution stated in this proposal appeared too esoteric (to the author) in nature, to merit a simple and concise explanation without performing any research to increase our cognizant awareness. Exploring the niche market of PCN and SCADA systems has elevated the necessity and importance of this work. In a very real way if no background research had been performed the project group would not have had the incite as to which product features are truly desirable by the industry as a whole. In the end focusing on all PCNs not only allowed this solution to meet small SCADA system requirements, in the future significant changes could be introduced to retro-fit this project for elaborate and complex PCNs, which require tri-homed firewalls and Encryption Data Acquisition Devices (eDAD).

Experimentation

To establish a Proof of Concept (POC) significant Experimentation has been completed prior to this proposal. We desired to implement an embedded solution to this project to illustrate the nature of the design can be used in nearly any computational spectrum. We needed to perform these POC experiments to discover the level to which the embedded solution could feasibly deliver each project aspect. As a result of completing the POCs we discovered certain aspects would need more time for consideration, and be delayed for subsequent projects and proposals. The general flow of each experiment was:

1. Create a hypothesis of how the system will solve a remote firewall security access issue.

2. Use the available embedded tools and hardware to perform a POC concerning the hypothesis.

3. If the POC is successful include this aspect in the possibilities for an end solution.

Here is a list of the POCs performed for this project already, and their results:

1. Manufacturer supplied utilities and Linux kernel provide the required customization for configuring an iptables kernel-based firewall. After creating a non-embedded prototype, we moved on to verify the embedded kernel offered all of the same functionality of the non-embedded kernel. This POC proved the embedded kernel does offer this functionality.

2. Linux operating system supports our eDAD devices. It turns out that the embedded application has no known application which can acquire color photos. Currently only grey-scale photos can be acquired by the eDAD device. This result combined with the limited storage space on the device itself changed the project scope and ruled out this part of the project.

3. Compile the reference PID program and check for functionality. This has proved successful, however the RISC based processor revealed several faults in the PID code which does not allow for little endian computer architectures. This changed the scope of the project as it revealed the amount of work just to implement the PID may be fairly significant.

Secondary Data Research

After the POC and secondary research was completed we were prepared to merge the results of both phases of research into a conceivable proposal. This methodology has so far revealed how using an embedded device can be more complex than a regular computer, and that specific domain knowledge is very important before jumping into a proposal. If there had been a deadline for competing organizations we could have very well lost the bid, due to our lack of domain knowledge. Here are some of the ways improving our domain knowledge has made this proposal more realistic:

1. Initially we knew external users would like to conceal SCADA system accesses but we did not know all of the reasons. The secondary data we researched revealed that device manufactures and related support organizations may also need remote access. This allowed us to focus on a much broader field which includes connection service types that include more than one protocol, instead of assuming an employee will only require SSH access, or something of a more narrow nature.

2. It was conceived that a dual-home firewall would serve as a good prototype for proving our system was desirable for PCN operators. We discovered that while this might be desirable in small distributed PCNs, a large PCN would require a full tri-homed configuration. This was valuable when we considered how this project could be extended in the future.

3. The full importance of securing SCADA systems was not fully conceived until we realized that SCADA systems are the networks that operate national critical infrastructures. This information connected all the dots for the project team and unified them in their vision to secure as many SCADA systems as possible. It also connected the YWD discovery to the project, which clearly demonstrates the necessity of the project.

Project Management

We used various techniques for encouraging team members and supporters of this project to work as diligently as possible. Building positive team ethos was essential for achieving buy-in for all stakeholders in the project. Encouraging industry research allowed team members to comprehend the level of importance the success of this project may have to the industry. Because the initial understanding of the end product nature was so ambiguous, the team concluded that a project management technique that lent greatly to unexpected changes in customer needs or research findings was necessary. To this end an Evolutionary Delivery model was established. This Evolutionary Delivery model is discussed in the following sections.

Evolutionary Delivery

The Evolutionary Delivery project management methodology is being used for this project since the desirable end product was very unclear from project outset. Added to this is the fact that our organizations embedded processor development knowledge, SCADA domain knowledge, and knowledge about SCADA customers and configurations was very minimal. The ramp-up time to discover these new domains was predicted as being an unusually difficult task. As depicted in the diagram below Evolutionary Delivery seemed to be the only solution that would allow our team to be as flexible as it needed to be, while also delivering a successful end product (McConnell, Steve).

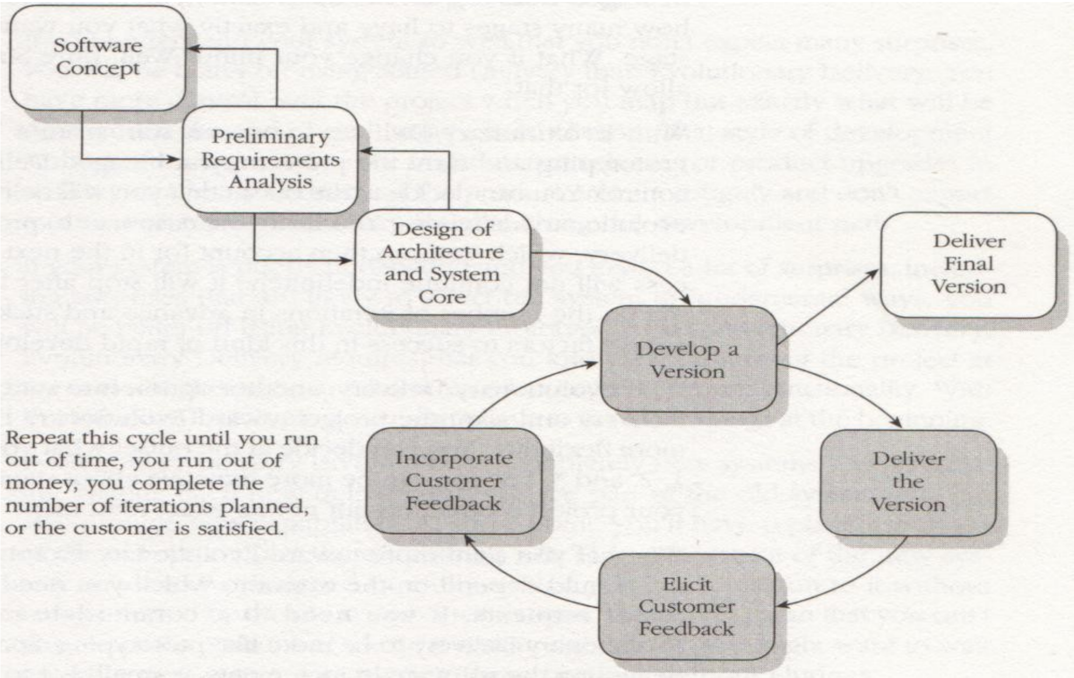

Software Concept
Preliminary Requirements Analysis
Design of Architecture and System Core
Deliver Final Version
Develop a Version
Repeat this cycle until you run out of time, you run out of money, you complete the number of iterations planned, or the customer is satisfied.
Incorporate Customer Feedback
Deliver the Version
Elicit Customer Feedback

Illustration 22: A diagram outlining the activities involved in Evolutionary Delivery.

This diagram has been taken from Steve McConnell's book on Rapid Software Development.
Notice how the Software Concept, Preliminary Requirements Analysis, and Design of Architecture and System Core are all connected, with both forward and backward arrows? This illustrates that every

attempt will be made to establish these phases of the project before commencing with the end prototypes, and product. Our research and POCs established miniature prototyping projects that will later be the subject of customer input sessions. While our current company and funding are at this time private, once we confront customers with the end product, they will most likely make quite a few suggestions that will change the product for the better, or allow specific installation customizations. While there are quite a few technologies and domain knowledge that must be ascertained before construction, during the prototyping phase it is likely that we will discover more about our customer and their desires that will change our understanding of the problem domain.

This is not unlike the Evolutionary Prototyping methodology and we will borrow from this methodology heavily. The following picture demonstrates that in Evolutionary Prototyping the product concept and the delivery of that product are heavily inter-twined:

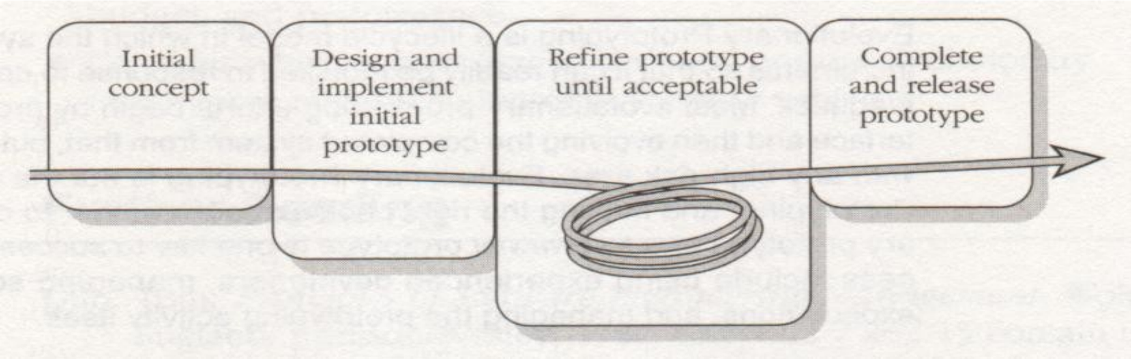

Initial concept
Design and implement initial prototype
Refine prototype until acceptable
Complete and release prototype

Illustration 23: The prototyping portion of Evolutionary Prototyping is not unlike our POCs

In our project there have been fairly little interaction with any client companies (This may come at a later time), but we do have some idea of the SCADA market and ascertained quite a bit of domain knowledge. To this end we have already partially conceived sub-projects that could help make the solution more effective. Essentially we have developed a whole family of products the nCase™ solution may comprise. However for this project we have narrowed the scoped of the solution in order for the most immediate and important solution to be made available first.

This is not all that different from the Staged Delivery methodology, but not exactly the same. We will borrow the concepts of staged delivery in that there is quite a bit of upfront design, however our POCs help to keep the project balanced in achieving its end goals. Essentially because there is no immediate customer, Evolutionary Prototyping is not a completely relevant methodology to use. We have a good deal of Internet Security knowledge so we have a good idea of how to protect systems connected to the Internet. This is what justifies our team using the Evolutionary Delivery model. However the team lacks a good understanding of small embedded devices, PLCs, SCADA systems, and PCNs. The ramp-up time to discover this domain knowledge will not be short. This is why we needed to borrow some concepts from Staged Delivery like producing the most important parts of the end-product first, and spending a majority of time in the up-front design phase. We also wanted to include a bit of the upfront design with the evolution of the delivery of the product to the customer (ie. the POCs).

Evolutionary Delivery allows us to master all aspects of the software delivery cycle without needing to be too focused on the customer or our own designs. The following picture shows how in Staged Delivery, each stage can cycle back into conceptual design, requirements analysis, and architecture:

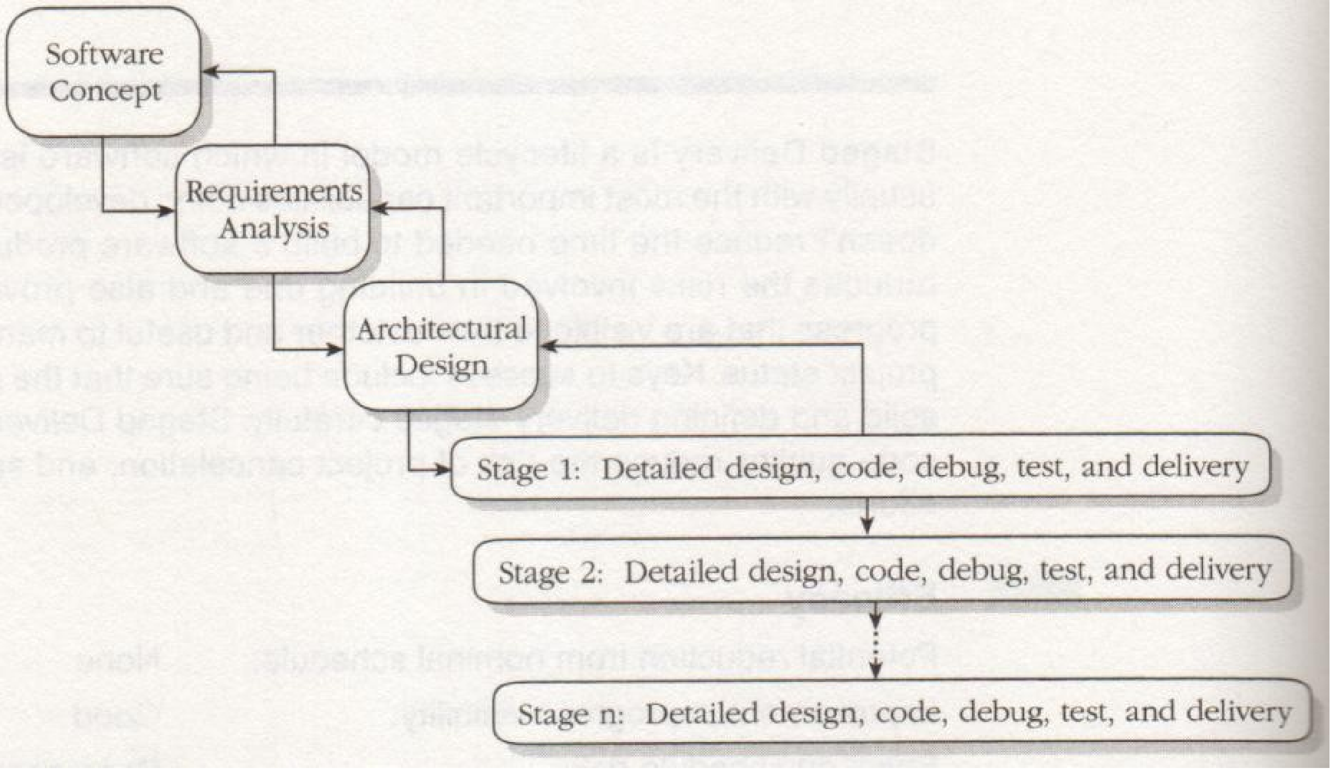

Software Concept
Requirements Analysis
Architectural Design
Stage 1: Detailed design, code, debug, test, and delivery
Stage 2: Detailed design, code, debug, test, and delivery
Stage n: Detailed design, code, debug, test, and delivery

Illustration 24: A robust design that fully analyzes the problem will give our design integrity.

This is a very valuable aspect of Staged Delivery that we would have denied ourselves if we had decided to merely develop using a prototype only methodology. Instead of fully focusing on the deliverable we have the freedom to take a step back, and draw out how each aspect of the architecture might affect other parts. This slow germination of an ideal solution is what we find so attractive in the Evolutionary Delivery model, and it is reflected by the fact we have created a whole family of products that may later be introduced as a part of the overall nCase™ solution.

Team Building Efforts

It was decided from the project outset that the project team consisted of members that all had their own unique research interests and motivations. While the end project had a fairly narrow target deliverable, it was decided that each member should have the freedom to make which ever contributions that they felt appropriate. To be honest the team was initially divided between delivering a niche solution, or a solution that solved every possible security threat to data communications. This division in the team members was not an easy obstacle to overcome, but instead of siding with any one group, management decided to diligently follow along the progressive developments of each of the groups. By the time the Research and POCs were completed it was discovered that the project requirements had both abstract and narrow (niche) aspects. As the project requirements were being drawn-up the decision of management to act as mediators in the project paid off, as the project factions disappeared during requirements analysis. The special skills of each member at this point had melded into a cohesive unit, only interested in producing a solution that would provide better critical infrastructure security.

Project Management used some special team building techniques during the research phase of the project to keep all participants interested. It was decided that the group social demographic lent towards science and discovery, definitely a more keen and brainy group. Instead of using tactics that encouraged the groups to get together for team activities (like sight-seeing, sporting events, and indoor/outdoor gaming) that would not likely excite all members of the team, it was decided that some clever project code names be developed, that would centralize the team around exactly what it was they had been brought together for. In the daily meetings ideas for project code names as well as names for the entire solution were requested. This was not an intense brains storming issue, but it accented the research brainstorming sessions, and provided team members with something colorful about the project that they could think about, when not entrenched in their own portion of the research. The team was asked to entertain producing code names and themes for the following:

1. Project Team Name
2. Project Code Name
3. Project Mascot
4. Overall Solution Name
5. Sub-project Names

This idea worked excellent. It wasn't the first idea conceived by management but it is the only idea that survived the research process and proved to be the best team defining process. Project members now possessed a part of the project that did not require them to think. It is perfect for those parts of a project that become blocked due to various circumstances (ie. personal, idea disagreements, external party negotiations etc.). We encouraged the team (if they could not work on the project for whichever reason) to at the very least consider building the ethos of the team and project goals, by pondering how these names might best fit the project. In this specific case we were especially lucky in that there was unanimous agreement (since this can also poorly affect teams). To offset the probability of inter-team disagreements we decided upon a code name filtration process, where the project names could change based on the progression of the project and the votes of the members.

During each short weekly meeting a vote concerning the code names which were desired for that week was completed. In this way if some members were absent during one week the project ethos was allowed to change in that moment. It actually made the whole process rather humorous. One week a very opinionated member went on vacation, while he was away the Project Name Honolulu was decided. When the team member returned he was surprised since he had just been to Mexico to avoid the costs of a Hawaiian vacation. This co-incidence allowed the member to talk more about his own vacation. It has been noted that even during summer vacations, some time is still spent considering non-important project based concepts such as this, making it a particular catchy way of involving the creative aspects that each member naturally possesses. The following are the names for this week:

1. Project Team Name - Mesopotamian Bohemia: The consensus is that the middle east could use a very nice vacation, or vacationing ethos. Team members have decided that the powers presently causing pain in the middle east should go on vacation.
2. Project Code Name - SteganoFolus: Essentially since an OTP style of encryption is being used

with the attempt to abstract its origination from computer hardware, steganographic concepts have been adopted. The suffix Folus is meant to be derived from the word Foil. In other words SCADA attack attempts are foiled by steganography.

3. Project Mascot - Udaspes Folus: Several members attempted to search for the root Latin or Greek meanings to the project code name. While there success was only marginal they discovered this charming moth from south east Asia would not escape their Google searches.

4. Overall Solution Name - nCase™ : The central goal of this project is to both protect devices and allow designated access, and the idea of concealing the network in a casing was conceived.

5. Sub-project Names :

   nWall™: A name for the firewall server device.

   Turret™: The opposite of a cannon or gun this turret uses sensory input information to secure actions close to the beachhead.

   BeachHead™: A name for the PID program. Essentially the job of the PID is to secure the beachhead by only allowing specific connections, preventing attackers from compromising the firewall system.

   SecureTransit™: Creating this name is almost a sub-project itself. The suggestions on how to provide a service like this, is now in its primary phase of research.

After buy-in was achieved from all stakeholders, management decided it would be much more plausible to have team building group outings which were more ambiguous in nature, and more likely to be enjoyed by every member of the team. The team would also have a better understanding of how they wanted to enjoy there team outings, since they had all shared some of their private ideas about the project (no matter how embarrassing) during the name creating exercises. In the process of resolving the weekly project names, various movies have been suggested by each team member. Management has provided outings and group activities to watch the following movies:

- War Games (1983)
- Sneakers (1992)
- Fearless (1993)
- Winged Migration (2003)
- Rivers and Tides (2004)
- Jet Li's Fearless (2006)
- Death of a President (2006)
- Lord of the Rings (Extended DVDs)
- Various Studio Ghibli Productions

Software Development

In our observation of the many different software development methodologies in existence, methodology seems to be of less importance than the actual project at hand. Smaller companies tend to require less formal standards (Extreme Programming) and some have no standards. Medium-sized companies tend to require standards after a number of failed projects, and may incorporate multiple in-house solutions (State Diagrams, Object Orientated Design, and Agile Techniques). Larger companies tend to adopt a specific methodology and require all teams to use the common methodology (Booch, OMT, UML). In our opinion this is justified due to the nature of the number of individuals involved in the larger projects hosted by larger companies, however in some way we can't help but ask the question, “Is it the problem domain that dictates the methodology or the organization?”. Extremely large companies realize there is no one methodology (MSF for CMMI, and MSF for Agile Programming) that can solve the problem of either too many employees, or too many managers (For instance Microsoft's Windows 2003 and Vista releases.). In the following section we discuss the development methodology we felt suited this particular project.

Architecture and Design

When we first visited the embedded board design company Intrinsyc Incorporated we were confronted with the fact that two of the three USB ports on the board did not work, because they were designed to the USB 2.0 specification and there was no associated linux driver. Essentially this means while the board has a fairly fast processor (416 Mhz) some customizations may be needed to provide maximum utility. Essentially a member of the company offered to pay us a fee to create the USB 2.0 driver, however we declined since our domain level experience was not a match. This brings up some interesting points, developing software for a small embedded board is not the same as developing software for larger computer systems, and many common resources are not readily available. To this end had we offered our assistance creating the driver software we would need to use a language and methodology which fit this device level situation (aka. system on a chip architecture). This event in itself made us realize that we would need to stick to fairly old software development methodologies. Methodologies that are tried and true. Using an object orientated design methodology to create a device driver would not be appropriate, since normally non-object orientated languages like C and assembler are used to create these Operating System (OS) features.

Most of the software that we used to help build the POCs also made use of non-object orientated languages. Some parts of the project may benefit from the number of pre-created object orientated libraries (such as XML configuration file processing). Our view is that any part of the project for which an object orientated code base is available, that part of the system will use an object orientated design methodology. It is highly unlikely that any object orientated methodologies will be adopted however, since the tools received from the manufacturer do not include a C++ compiler, and we have not been able to successfully compile or locate one. Essentially the PID program will need to perform network packet inspection and decryption functions, so it must be able to operate as efficiently as possible. For this reason it will use a non-object orientated language and design methodology. We will use Flow, State, Component, System, and Network Diagrams to express the system architecture and design.

Quality Management

We are using the Concurrent Versions System (CVS) to manage our source code and releases. A utility like this is invaluable for teams of any size, since it provides for project specific branches. This isolates one teams work from another and allows small teams to work together without involving the entire engineering organization in the creations of a separate group. It can also be considered as a great form of software project history for reporting on project issues. However these tools may represent a single point of failure if not used properly.

We are using coding techniques that allow for compiling the programs on multiple systems and architectures, however we will primarily only be concerned with the x86 and the ARM Xscale® Linux platforms. No bug reporting mechanism is being used due to the small size of the team (ie. one). Several testing methodologies have been considered. White Box testing will be completed by development. No paper trail will be created for this testing, since it is expected that development is able to complete the application as specified. Random user-based action tests will comprise the Black Box testing portion of Quality Control. A paper trail will be produced for this testing as the Black Box tests will attempt to achieve actual user case scenario results for instance:

1. Can a user configure the server with Timeouts, Connection Limits, Hour Limits, and Customization Allowed settings, and do each of these features work as expected.

2. Can a user configure the server to have device specific settings. Do the device specific settings properly over-ride the general settings for the server.

3. Can a client make use of a specific service and IP address to enable a connection to a firewall hosted device. If so how well do these services perform, and what level of service to they provide to the end user.

4. If a default service is not given can the user successfully specify a custom service to open. If this feature has been disabled is the service then unavailable, and how well is that communicated to the end user.

5. For a device service that requires the device itself to initiate a connection, how well is this service provided. What aesthetic quality does this service provide a user.

6. Does the user manual tell the user everything they need to know in order to use the product successfully and skillfully.

7. What aesthetic quality do the client and server application provide to the end user.

Finally since it is required that this project use a significantly random encryption methodology a brute force testing technique will be used, to ensure that the data packets being sent have a verifiably random nature in their composition. These tests will log DAD initiator packets from various IP address ranges and compare the data bits sent over the wire. Stress testing will be used to ensure the PID application operates well under a heavy volume of DAD requests.

Attack testing will be used to ensure the firewall operates as specified, and that the applications operate well even when under attack (ie. no vulnerabilities.). A buffer overflow in this application might prevent users from using the application as desired if under attack, so Attack testing will be utilized to ensure the code base is not subject to blatant security holes. The following firewall rules will be tested for conformance with the specification of this project:

- Devices used should not be allowed to communicate with any devices other than their local PCN devices, using the appropriate destination ports, services, and communications states.

- No other devices (enterprise, local, or remote) should be able to communicate directly with SCADA devices, except by using the planned communication states, services and devices operating on the PCN (Essentially the system is completely locked up.).

- Designated connections may be permitted between the designated machines and the SCADA devices for short periods of time. Does the PID program provide for this functionality in a secure way?

VII. Milestones

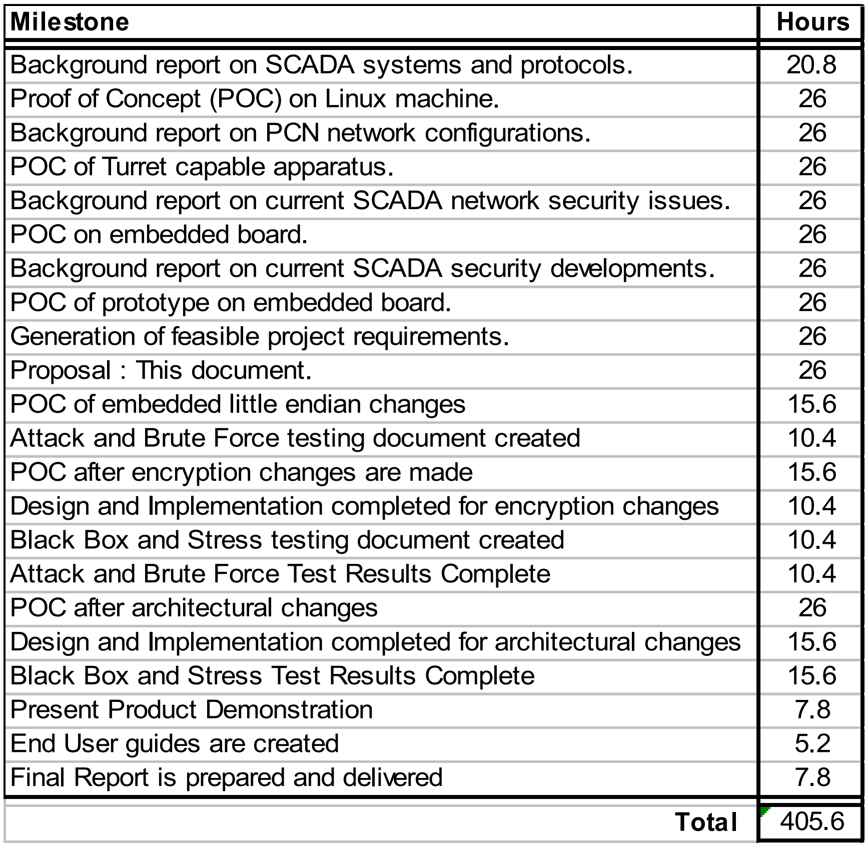
It has been decided that the following project milestones will best facilitate this project and provide potential customers with the best knowledge about project success. These milestones can be categorized into seven categories Background Research [**B**], Configuratory POCs [**C**], Prototyping POCs [**P**], Implementation and Design [**I**], Software Testing [**T**], User Documentation and Demonstration [**U**], and Project Documentation [**D**].

1. Background report on SCADA systems and protocols. **B**
2. Proof of Concept (POC) on Linux machine. **C**
3. Background report on PCN network configurations. **B**
4. POC of Turret capable apparatus. **C**
5. Background report on current SCADA network security issues. **B**
6. POC on embedded board. **C**
7. Background report on current SCADA security developments. **B**
8. POC of prototype on embedded board. **C**
9. Generation of feasible project requirements. **D**
10. Proposal : This proposal document. **D**
11. POC of embedded little endian changes. **P**
12. Attack and Brute Force testing document created. **T**
13. POC after encryption changes are made. **P**
14. Design and Implementation completed for encryption changes. **I**
15. Black Box and Stress testing document created. **T**
16. Attack and Brute Force Test Results complete. **T**
17. POC after architectural changes. **P**
18. Design and Implementation completed for architectural changes. **I**
19. Black Box and Stress Test Results complete. **T**
20. Present Product Demonstration. **U**
21. End User Guides are created. **U**
22. Final Report is prepared and delivered. **D**

VIII. Current Schedule

When we plotted the milestones and associated tasks for this project in a Gantt Chart the end project consisted of a total of 78 days or 624 hours, however this figure is not exact, mostly due to the fact that this project really started before the beginning of the 2006 Fall Term. Quite a few design issues have been the subject of non-school term work and thinking. If we condense this into the mandatory 405 hours of a 9 credit practicum essentially we have over-estimated the project by 35%. This means that each 8 hour work day can be reduced to roughly 5.2 hours per day (or 64.9% of 8 hours). This is not an unfair assumption, and it only makes sense that the hours advertised by BCIT, would include more work than one person could normally perceive, given the nature of a normal BCIT school term. The Gantt chart data we have created for this project appears in Appendix D.

As of November 10th, 2006 four of the seven POCs have been demonstrated as successful. The project has completed a total of 49 days of work or 392 hours. If we apply the over-budget formula described above, the project has completed a total of 254.8 hours. Essentially the project is 63% complete and on schedule for completion by the December 15th, 2006 deadline. Using this formula the following shows the work required for each milestone previously defined:



Milestone	Hours
Background report on SCADA systems and protocols.	20.8
Proof of Concept (POC) on Linux machine.	26
Background report on PCN network configurations.	26
POC of Turret capable apparatus.	26
Background report on current SCADA network security issues.	26
POC on embedded board.	26
Background report on current SCADA security developments.	26
POC of prototype on embedded board.	26
Generation of feasible project requirements.	26
Proposal : This document.	26
POC of embedded little endian changes	15.6
Attack and Brute Force testing document created	10.4
POC after encryption changes are made	15.6
Design and Implementation completed for encryption changes	10.4
Black Box and Stress testing document created	10.4
Attack and Brute Force Test Results Complete	10.4
POC after architectural changes	26
Design and Implementation completed for architectural changes	15.6
Black Box and Stress Test Results Complete	15.6
Present Product Demonstration	7.8
End User guides are created	5.2
Final Report is prepared and delivered	7.8
Total	405.6

IX. Deliverables

There are numerous deliverables involved in completing this project. The three main deliverables are this proposal, a demonstration of the final product, and a report detail each aspect of the project. This section details any portions of this work which might not be obvious if not otherwise explained.

Project Proposal

This project proposal has entailed a considerable amount of research, however this can be expected of a project which attempts to solve computer engineering aspects that are not widely known in the industry. While the report itself is the only tangible deliverable, here are the components deliverables provided in this proposal:

1. Background report on SCADA systems and protocols.
2. Background report on SCADA network configurations
3. Background report on current SCADA network security issues.
4. Background report on current SCADA security developments.
5. Description of the Proof of Concepts already conducted.
6. Description of the general feasibility of the project.
7. Description of the product scope and end deliverables.
8. Description of possible future projects and the nCase™ suite (family of products).

Demonstration

The project should ideally be demonstrated to an audience so that someone at BCIT can confer the project has achieved its end goals, and a pass or fail mark ledgered. This presentation will include:

1. Presentation slides: That detail what the project has attempted to accomplish.
2. Actual demonstration: Certain activities are carried out before nCase™ is applied, and after. The effect should produce a desire in the observer to buy the nCase™ name brand product and use it in production, or to at least consider donating to the cause of securing critical infrastructure systems.
3. Physical device configuration: This configuration will be used to demonstrate the nCase™ product in operation.

Final Report

It is required the project end in a meaningful way. Essentially this means that a report must be produced to document all the various stages of the project, and provide the design work, testing status, and any other information of importance. According to the above proposal and our witnesses this report should contain the following:

-

1. Background report on SCADA systems and protocols.
2. Background report on SCADA network configurations
3. Background report on current SCADA network security issues.
4. Background report on current SCADA security developments.
5. Description of the Proof of Concepts already conducted.
6. Description of possible future projects and the nCase™ suite (family of products).
7. Description of the general feasibility of the project.
8. Description of the end product scope and end deliverables.
9. Detailed product design work and necessary changes.
10. Testing plans, reports, and results.
11. Description of the how successful the methodologies achieved their end goals
12. Description of the demonstrations already conducted.
13. A conclusion regarding the project including any industry connections gained.

Physical Deliverables

It has been decided that printed paper alone cannot complete a project, a project must have concrete deliverables. Accompanying the end report the completion of this project should provide:

1. Detailed design work: Showing the entire network, systems, and programs design.
2. User Guide: Detailing how a user should install, administrate, and use the system.
3. Embedded kernel package: A flash ROM file for burning to the Cerfboard device.
4. Embedded programs package: A flash ROM file for burning to the Cerfboard device.
5. Client application: An x86 compiled program that can access the embedded firewall.
6. Source Code: In printed form all code and configuration files necessary for installation
7. DVD copy: A binary copy of all of the above should accompany the printed volumes.

X. Project Benefits

It is expected that Designated Access Devices (DAD) will allow corporations using SCADA systems and PCNs to provide for the proper protection of their equipment even during their shift toward technologies that are moving closer and closer to the Internet. After performing the general background research we are convinced that this is a real problem facing the world today. While the solution is not exactly a simple one, firewalls in general are complex configurations to maintain. Corporations seeking to maintain current insurance premiums may in the future be required to follow specific security procedures in order to pass security checks and insurance requirements. Additionally organizations may be required to undergo security checks in order to ensure their SCADA and automation systems are not vulnerable to attacks. In this situation the nCase™ solution will provide these corporations an added tool which allows the organization to pass security checks, and potentially still provide remote access to support organizations, which are in the business of working on PLC equipment and devices remotely. In general the end goals of the project are to provide a safer world for everyone.

XI. References

Brooks, F. P. *No Silver Bullet - essence and accident in software Engineering,* Proceedings of the IFIP Tenth World Computing Conference, pp. 1069-1076. 1986. Retrieved October 10, 2006 from the World Wide Web: http://inst.eecs.berkeley.edu/~maratb/readings/NoSilverBullet.html

Brown, W. Stanley. and Holbrook, Bernard D. *A History of Computing Research at Bell Laboratories (1937-1975)*, Computing Science Technical Report No. 99. AT&T Bell Laboratories. 1982. Retrieved October 10, 2006 from the World Wide Web: http://cm.bell-labs.com/cm/cs/cstr/cstr99.html

Byres, Eric. *The Need for Security Testing of Devices,* Process Control Systems Forum (PCSF). 2006 Spring Meeting. June 5 - 7, 2006. La Jolla, California. Retrieved October 10, 2006 from the World Wide Web: https://www.pcsforum.org/events/2006/spring/

Byres, Eric and Lowe, Justin. *Real World Cyber Security Risks for Industrial Control Systems,* The Industrial Ethernet Book. Issue 22. September 2004. Retrieved October 10, 2006 from the World Wide Web: http://ethernet.industrial-networking.com/articles/articledisplay.asp?id=206

Chu, Bei-Tseng. and Wang, Yongge. *sSCADA: Securing SCADA Infrastructure Communications,* Software and Information Systems (SIS). University of North Carolina. August 5, 2004. Retrieved October 10, 2006 from the World Wide Web: http://eprint.iacr.org/2004/265.pdf

Cunningham, R. Paxson, V. Staniford, S. and Weaver, N. *A taxonomy of computer worms,* In Proceedings of the 2003 ACM Workshop on Rapid Malcode. Washington, DC. USA. October 27 - 27, 2003. WORM '03. ACM Press, New York, NY, 11-18. http://0-doi.acm.org.innopac.lib.bcit.ca:80/10.1145/948187.948190

digitalbond.com. *Blog: On the Need for Free (and Fee) Open SCADA Vuln Research,* Mattew Franz. previous employee of Cisco CIAG. October 3, 2006. Retrieved October 10, 2006 from the World Wide Web: http://www.digitalbond.com/SCADA_Blog/2006/10/on-need-for-free-and-fee-open-scada.html

Fernandez, Andres E. and Fernandez, John D. *SCADA Systems: Vulnerabilities and Remediation,* Researchers from Texas A&M University. Consortium for Computing Sciences in Colleges Journal. pp160-168. April 2005. Retrieved October 10, 2006 from the World Wide Web: http://portal.acm.org/citation.cfm?id=1047846.1047872

Franz, Matthew. and Pothamsetty, Venkat. *SCADA HoneyNet Project: Building Honeypots for Industrial Networks,* Research Project. Critical Infrastructure Assurance Group (CIAG). Cisco Systems, Inc. Retrieved October 10, 2006 from the World Wide Web: http://scadahoneynet.sourceforge.net/

GAIT. *Good Practice Guide on Firewall Deployment for SCADA and Process Control Networks,* National Infrastructure Security Coordination Centre. February 23, 2005. Retrieved October 10, 2006 from the World Wide Web: http://www.niscc.gov.uk/niscc/docs/re-20050223-00157.pdf

GAO-04-628T. *Critical infrastructure protection: challenges and efforts to secure control systems,* Testimony Before the Subcommittee on Technology Information Policy, Intergovernmental Relations and the Census. House Committee on Government Reform. March 30, 2004. Retrieved October 10, 2006 from the World Wide Web: http://www.gao.gov/new.items/d04628t.pdf

greatachievements.org. *A Historical Timeline of the Internet,* Greatest Engineering Achievements of the Twentieth Century project. National Academy of Engineering. 2006. Retrieved October 10, 2006 from the World Wide Web: http://greatachievements.org/?id=3736

Hamadeh, I. Hart, J. Kesidis, G. and Pothamsetty, V. *A Preliminary Simulation of the Effect of Scanning Worm Activity on Multicast,* In Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation (June 01 - 03, 2005). Workshop on Parallel and Distributed Simulation. IEEE Computer Society, Washington, DC, 191-198. http://dx.doi.org/10.1109/PADS.2005.2

Kirstein, Peter T. *Early Experiences with the ARPANET and INTERNET in the UK,* Department of Computer Science. University College London. IEEE Annals of Computing, 21, 1, 1999. Retrieved October 10, 2006 from the World Wide Web: http://www.cs.ucl.ac.uk/staff/jon/arpa/internet-history.html

Leon-Garcia, Alberto and Widjaja, Indra. *Communications Networks: Fundamental Concepts and Key Architectures,* University of Toronto. Prentice Hall. 2001.

members.shaw.ca/yarrowwd. *Yarrow Waterworks District SCADA Logs and Photos,* Homepage. Yarrow Waterworks District. Retrieved October 10, 2006 from the World Wide Web: http://members.shaw.ca/yarrowwd/

McConnell, Steve. *Rapid Development: Taming Wild Software Schedules,* Microsoft Press. Redmond, Washington USA. 1996.

modbus-ida.org. *Object Messaging Specification for the MODBUS/TCP Protocol,* Version 1.1. Modbus Organization, Inc. November, 8, 2004. Retrieved October 10, 2006 from the World Wide Web: http://www.modbus-ida.org/docs/Object_Messaging_Protocol_ExtensionsVers1.1.doc mtl-inst.com. *Industrial Ethernet Networks get an Intrinsically Secure™ solution,* Press Release. The MTL Instruments Group plc. October 2006. Retrieved Oct. 2006 from the World Wide Web: http://www.mtl-inst.com/newsroom/press_releases/pr456.htm

NERC. *SQL Slammer Worm Lessons Learned for Consideration by the Electricity Sector,* North American Electric Reliability Council. June 20, 2003. Retrieved October 10, 2006 from the World Wide Web: http://www.esisac.com/publicdocs/SQL_Slammer_2003.pdf

netfilter.org. About the netfilter/iptables project, The Netfilter Core Team. Open Source Distribution. Retrieved October 10, 2006 from the World Wide Web: http://www.netfilter.org/about.html

NISCC. *Good Practice Guide: Process Control and SCADA Security,* National Infrastructure Security Coordination Centre. October 25, 2005. Retrieved October 10, 2006 from the World Wide Web: http://www.niscc.gov.uk/niscc/docs/re-20051025-00940.pdf

npl.co.uk. *Packet Switching, Donald Davies and X.25,* National Physical Laboratory. United Kingdom. Retrieved October 10, 2006 from the World Wide Web:
http://www.npl.co.uk/about/famous_names/donald_davies.html
http://en.wikipedia.org/wiki/Packet_switched_network
http://en.wikipedia.org/wiki/X.25

Paller, Allan. *Cyber Attacks Against SCADA and Control System,* Special Webcast. The SANS Institute. September 7, 2006. Retrieved October 10, 2006 from the World Wide Web:
http://www.infoworld.com/article/06/09/11/37NMmain_1.html
https://www.sans.org/webcasts/show.php?webcastid=90748

Peterson, Dale. *Blog: Byres - - Tofino,* Lead Network Security Consultant. Digital Bond, Inc. October 17, 2006. Retrieved October 25, 2006 from the World Wide Web: http://www.digitalbond.com/SCADA_Blog/2006/10/byres-tofino.html

Pothamsetty, Venkat. Where Security Education is Lacking, In Proceedings of the 2nd Annual Conference on information Security Curriculum Development. September 23-24, 2005. Kennesaw GA, USA. InfoSecCD '05. ACM Press. New York, NY. Retrieved October 10, 2006 from the World Wide Web: http://doi.acm.org/10.1145/1107622.1107635

profibus.com. *PROFINET Security Guideline,* Version 1.0. PROFIBUS International. March 2005. Retrieved October 10, 2006 from the World Wide Web: http://www.profibus.com/pall/meta/downloads/article/00341

rockwellautomation.com. *EtherNet/IP: Industrial Protocol White Paper,* Logix/NetLinx Technology Adoption. Rockwell Automation. Institute of Electrical and Electronic Engineers. EFTA October 2001. Retrieved October 10, 2006 from the World Wide Web: literature.rockwellautomation.com/idc/groups/literature/documents/wp/enet-wp001_-en-p.pdf

Schiffer, Viktor. *The CIP Family of Fieldbus Protocols and its Newest Member – EtherNet/IP,* Engineering Manager. European Technology Development Unit Rockwell Automation. Institute of Electrical and Electronic Engineers. EFTA 2001. Retrieved October 10, 2006 from the World Wide Web:
literature.rockwellautomation.com/idc/groups/literature/documents/wp/nets-wp003_-en-p.pdf

Schneier, Bruce. *The Secret Story of Non-Secret Encryption*, Crypto-gram Newsletter. Counterpane Internet Security Inc. Retrieved October 10, 2006 from the World Wide Web:
http://www.schneier.com/crypto-gram-9805.html

Shannon, Claude E. *A symbolic analysis of relay and switching circuits*, Masters Thesis. Massachusetts Institute of Technology. Dept. of Electrical Engineering. 1940. Retrieved October 10, 2006 from the World Wide Web:
http://hdl.handle.net/1721.1/11173

Stibitz, George. *Links that credit the Stibitz-Shannon Team with remote computation,* Retrieved October 10, 2006 from the World Wide Web:
http://www.bell-labs.com/history/unix/blcontributions.html
http://www.denison.edu/mathsci/stibitz/bio.html
http://ei.cs.vt.edu/~history/VonNeumann.html
http://www.libsci.sc.edu/bob/ISP/bell.htm
http://www.kerryr.net/pioneers/stibitz.htm
http://www.maxmon.com/1937ad.htm

theregister.co.uk. *Hacker jailed for revenge sewage attacks,* Software News Category. The Register. 2001. Retrieved October 10, 2006 from the World Wide Web:
http://www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_sewage/

Turing, Alan M. *The Biography of Alan Mathison Turing*, School of Mathematics and Statistics. University of St Andrews. Scotland. 2003. Retrieved Oct. 10, 2006 from the World Wide Web:
http://www-groups.dcs.st-and.ac.uk/~history/Printonly/Turing.html

vovida.org. *Your Source for Open Source Communication*, Cisco Systems, Inc. Retrieved October 10, 2006 from the World Wide Web:
http://www.vovida.org/protocols/

wikipedia.org. *One-time Pad,* The Wikimedia Foundation. Retrieved October 10, 2006 from the World Wide Web:
http://en.wikipedia.org/wiki/One_time_pads

XII. Appendix

Appendix A – Glossary

Note all the definitions below not originating in this proposal were taken from the Wikimedia Foundation, Inc. website called Wikipedia®.

ASP *Application Service Provider* : is a business that provides computer-based services to customers over a network. Software offered using an ASP model is also sometimes called On-Demand software. Whole application suites can be offered or only single applications.

DAC *Designated Access Control* : refers to the access control rules of a specific nCase™ installation. Each installation may be different depending on the access control rules of that configuration. These access control rules are implicit in the configuration file which governs the device accesses. The rules are not based on user access, but predefined general and device specific settings (ie. number of accesses per hour etc.).

DAD *Designated Access Device* : refers to devices that are only accessible over the network in a designated (controlled) fashion. In this paper it references a device protected by the nCase™ solution, which provides access to machines for which a proper client request is received.

DMZ *Demilitarized Zone* : is a network area (a subnetwork) that sits between an organization's internal network and an external network, usually the Internet (In the case of this document we mean a DMZ between the corporate network and the local PCN.). The point of a DMZ is that connections from the internal and the external network to the DMZ are permitted, whereas connections from the DMZ are only permitted to the external network -- hosts in the DMZ may not connect to the internal network. This allows the DMZ's hosts to provide services to the external network while protecting the internal network in case intruders compromise a host in the DMZ. For someone on the external network who wants to illegally connect to the internal network, the DMZ is a dead end.

PCN *Process Control Network* : is a communications network that is used to transmit instructions and data between control and measurement units and Supervisory Control and Data Acquisition (SCADA) equipment. These networks have, over the years, used many of the technologies and topologies utilized in other network applications. However, Process Control Networks (PCNs) have several special requirements that must be met in order for the solution to be acceptable to the industry.

PLC *Programmable Logic Controller* : is a small computer used for automation of real-world processes, such as control of machinery on factory assembly lines. The PLC usually uses a microprocessor. The program can often control complex sequencing and is often written by engineers. The program is stored in battery- backed memory and/or EPROMs. Unlike general-purpose computers, the PLC is packaged and designed for extended temperature ranges, dirty or dusty conditions, immunity to electrical noise, and is mechanically more rugged and resistant to vibration and impact.

PID *Packet Inspection Daemon* : This is a program that inspects network packets received by a network device. Various uses for such an application can be designed. In this paper it references the BeachHead™ program which analyzes packets to determine if they are requesting a DAD connection or not. If so the packet is processed accordingly.

SaaS *Software as a Service* : a model of software delivery where the software company provides maintenance, daily technical operation, and support for the software provided to their client. SaaS is a model of software delivery rather than a market segment; software can be delivered using this method to any market segment including home consumers, small business, medium and large business. A common example of this is salesforce.com which houses the entire operation, even including the actual data (ie. storing the data on company hardware for a service purchasing entity.).

SCADA *Supervisory Control And Data Acquisition* : SCADA may be called Human-Machine Interface (HMI) in Europe. The term refers to a large-scale, distributed measurement (and control) system. SCADA systems are used to monitor or to control chemical, physical or transport processes.

SOA Service Orientated Architecture : expresses a perspective of software architecture that defines the use of loosely coupled software services to support the requirements of business processes and software users. In an SOA environment, resources on a network are made available as independent services that can be accessed without knowledge of their underlying platform implementation. A service-oriented architecture is not tied to a specific technology and may be implemented using a wide range of interoperability standards including RPC, DCOM, ORB or WSDL.

Appendix B – Family of Potential Programs

SecureTransit™

The SecureTransit™ service makes use of an Apache Web Server hosted in the DMZ of the PCNs tri-homed firewall. As mentioned before the ultimate security of such a solution is not immediately verifiable. It could be that placing a web server at this position is not worth the security risk. However this idea has been conceived as a catch all solution for providing remote access (ie. No one needs to download the application in order to use the service, instead everyone who can connect to the Web Server has access to the application if the Access Control Lists (ACL) permit it.). The following diagram depicts one way this product might be configured:

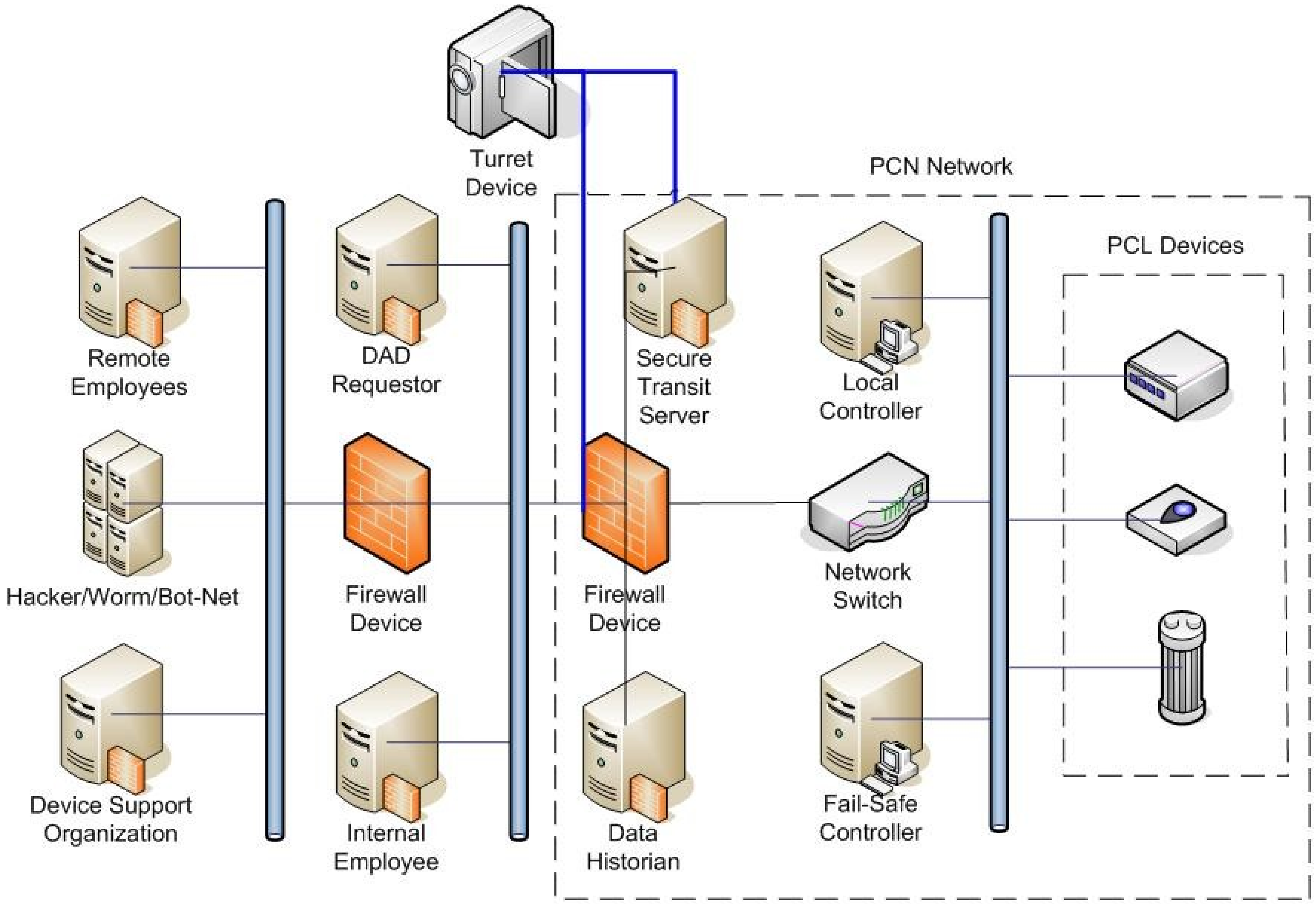


Illustration 25: SecureTransit™ service resides in the DMZ of the PCN firewall with Data Historian.

Notice that this implementation requires the use of a Turret™ encryption system. All DAD requests for PLC device support organizations are established from a specific machine (with a robust host-based firewall) within the enterprise network. This limits the possibility of security breaches in this scenario.

Turret™ Encryption

The Turret™ encryption system makes use of various optical and sensory data captured using various devices to create One-Time Pad (OTP) data, for use in secure data communications between two network computers. The method is considerably effective in situations where the machines are in close proximity, since the data can be communicated using a non-public (social) communication network. In this situation it also very important that the machines themselves are secured from possible data interception. The following diagram illustrates one possible implementation of the encryption system:

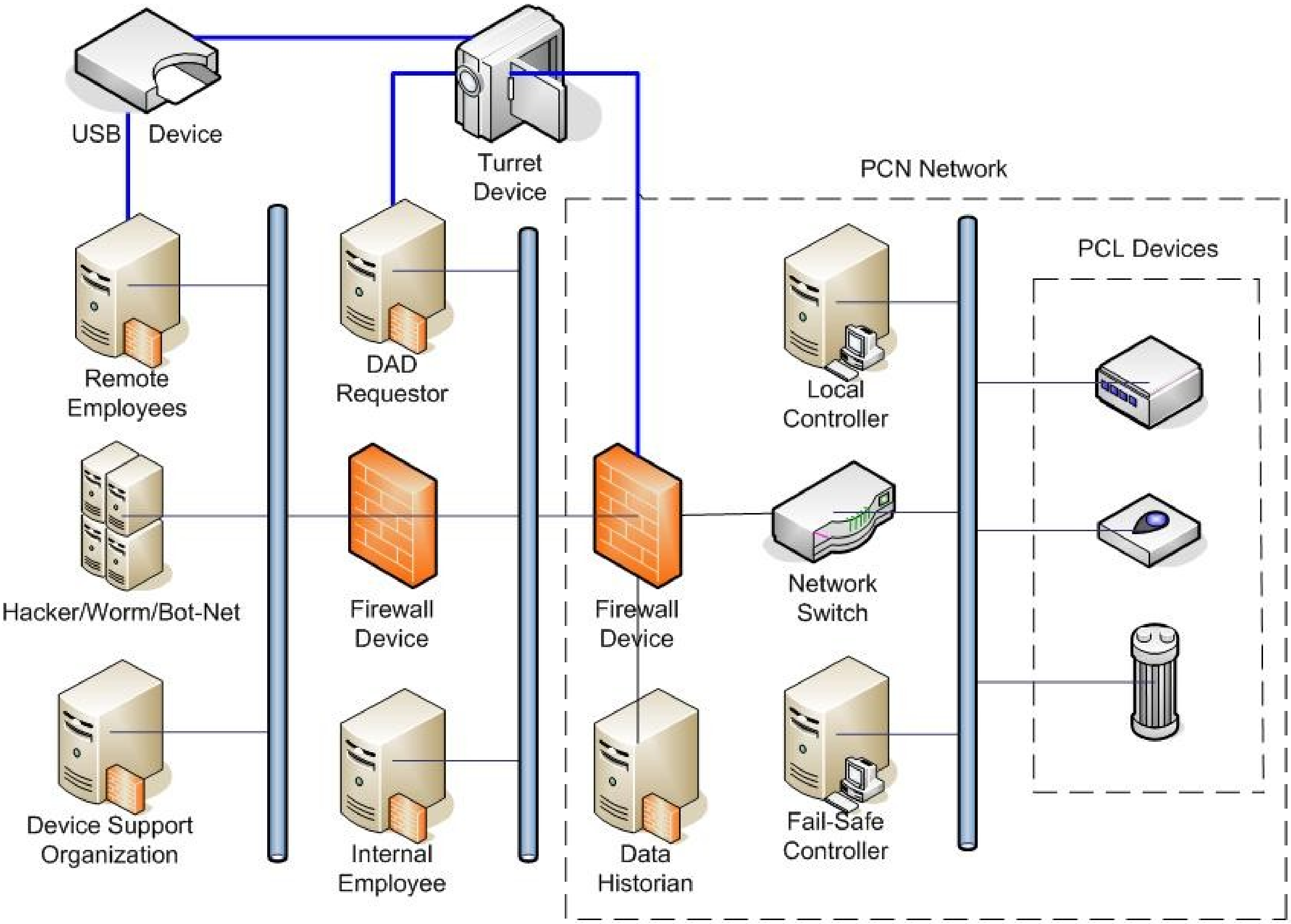


Illustration 26: In the Turret™ system the OTP data is sent to a machine via. a hidden method.

The user first authenticates using a VPN to enable access past the corporate firewall. Then using a specific machine in the enterprise network, a certain PCN device/machine can be accessed. In large installations it would be recommended that DAD connections occur from specific machines inside the enterprise network (To allow for static tested firewalls rules.). As can be seen in the above illustration it is also conceivable that a remote user gain access directly to a PCN device, using OTP data stored on a USB device. In large enterprise networks using a USB device for remote connections is not advisable, since there is already enough hardware within the local enterprise network to host the DAD connections.

In remote situations however USB data might prove invaluable. The main problematic issue in this case is the transportation of such data for convenience as well as security (ie. what if the USB device is lost). This might be particularly useful for SCADA systems that are remotely operated (Such as oil pipe lines etc.). The user authenticates with the external firewall host using the predefined USB contained OTP data. The firewall machine then authenticates the user for a VPN connection to the PCN. So why add double security for something like a VPN connection? Because a SCADA system or PCN is not just any old network, more likely they have the potential to do even more bad than they do good. Any vulnerability in a system like this should be minimized as much as possible. The following diagram depicts this scenario:

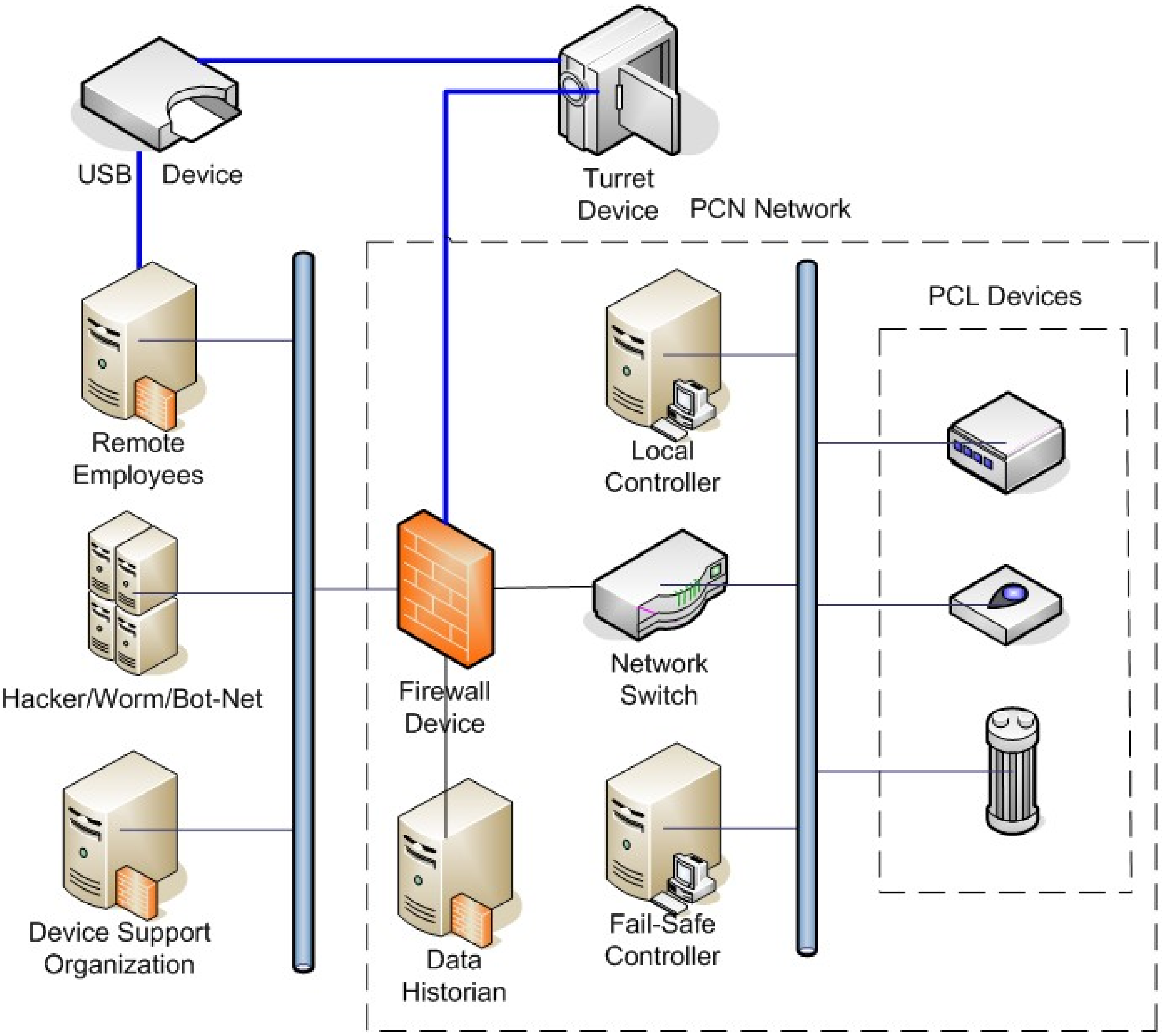


Illustration 27: A Turret™ system providing a more secure form of network access.

Appendix C – nCase^{TM} Project Schedule

ID	Task_Name	Duration	Predecessors
1	Find relevant SCADA documents	1 day	
2	Read material	1 day	1
3	Write report summary	1 day	2
4	Combine material and complete Report	1 day	3
5	Report on SCADA systems and protocols	0 days	4
6	Install dual-homed Firewall	1 day	5
7	Install basic PID program	1 day	6
8	Install Encryption Libraries	1 day	7
9	Conduct Proof of Concept	2 days	8
10	Basic POC on Linux machine.	0 days	9
11	Find relevant PCN documents	1 day	10
12	Read material	1 day	11
13	Write report summary	1 day	12
14	Combine material and complete Report	2 days	13
15	Report on PCN configurations	0 days	14
16	Locate WebCam drivers and utility programs	1 day	15
17	Install WebCam drivers and utility programs	1 day	16
18	Test whether utility program are automatable.	1 day	17
19	Conduct Proof of Concept	2 days	18
20	POC of Turret capable apparatus.	0 days	19
21	Find relevant SCADA security documents	1 day	20
22	Read material	1 day	21
23	Write report summary	1 day	22
24	Combine material and complete Report	2 days	23
25	Report on current SCADA network security issues.	0 days	24
26	Install and test embedded device tools and compilers	1 day	25
27	Create customized iptables embedded kernel	1 day	26
28	Cross-compile the PID program	1 day	27
29	Conduct Proof of Concept	2 days	28
30	POC of embedded device project usability	0 days	29

31	Find relevant current SCADA security developments documents	1 day	30
32	Read material	1 day	31
33	Write report summary	1 day	32
34	Combine material and complete Report	2 days	33
35	Background report on current SCADA security developments	0 days	34
36	Create customized drivers and application packages	1 day	35
37	Attempt to create a C++ cross-compiler for embedded developments	1 day	36
38	Install and configure kernel and application packages	1 day	37
39	Conduct Proof of Concept	2 days	38
40	POC of embedded device with working prototype	0 days	39
41	Combine technological report and POC findings	1 day	40
42	Create the problem statement	1 day	41
43	Create the project requirements and scope	1 day	42
44	Combine material and complete Report	2 days	43
45	Report on Feasible Project Requirements	0 days	44
46	Specify deliverables, innovation explanation, and project benefits	1 day	45
47	Explain methodologies and software development process	1 day	46
48	Update project schedule and describe Future Projects and create descriptions	1 day	47
49	Combine material and complete Proposal document	2 days	48
50	Present Proposal	0 days	49
51	Analyze code for sections which need endian corrections	1 day	50
52	Make endian based changes	1 day	51
53	Conduct Proof of Concept	1 day	52
54	POC of embedded little endian changes	0 days	53
55	Define ways the firewall could be defeated or not perform as expected	0.5 days	54
56	Define ways the PID have overflow or DOS issues	0.5 days	54
57	Combine material and complete test document	1 day	56,55
58	Attack and Brute Force testing document created	0 days	57
59	Define a modular encryption design	1 day	58
60	Implement encryption stubs	1 day	59

61	Conduct Proof of Concept	1 day	60
62	POC after encryption changes are made	0 days	61
63	Integrate encryption stubs to main source	1 day	62
64	Conduct Attack and Brute Force Testing	0.5 days	63
65	Combine design material and complete document	0.5 days	63
66	Design and Implementation for encryption changes	0 days	64,65
67	Make Black Box Tests from State Diagrams and Use Case Scenarios	0.5 days	66
68	Make Stress Tests from Network and System Diagrams	0.5 days	66
69	Combine material and complete test document	1 day	67,68
70	Black Box and Stress testing document created	0 days	69
71	Review Attack and Brute Force Test Results	0.5 days	64,70
72	Draw conclusion from test results	0.5 days	71
73	Complete test results document	1 day	71,72
74	Attack and Brute Force Test Results Complete	0 days	73
75	Define architectural changes necessary	1 day	74
76	Implement configuration file processor stubs	1 day	75
77	Implement configuration run-time checking and processing stubs	2 days	76
78	Conduct Proof of Concept	1 day	77
79	POC after architectural changes	0 days	78
80	Integrate configuration file processing to main source	0.5 days	79
81	Integrate configuration run-time checking and processing to main source	0.5 days	79
82	Conduct Black Box and Stress Testing	1 day	80,81
83	Combine design material and complete document	1 day	82
84	Design and Implementation for architectural changes	0 days	83
85	Revisit Attack and Brute Force Tests to check changes	0.5 days	84
86	Complete Black Box and Stress Testing	1.5 days	84
87	Complete test results document	1 day	86
88	Attack and Brute Force Test Results Complete	0 days	87
89	Describe solution using simple but direct language	0.25 days	88
90	Create solution demonstration steps	0.5 days	88
91	Complete and practice demonstration using distributable	0.25 days	90

	package		
92	Present the Solution to an audience	0.5 days	91
93	Present Product Demonstration	0 days	92
94	Compile a list of questions received from presentation	0.5 days	92
95	Create end user guide based on audience attendance	0.5 days	94
96	End User guides are created	0 days	95
97	Compile questions from proposal which need to be answered.	0.25 days	50
98	Compile POC information and results	0.25 days	79, 10, 30, 40, 54, 20, 62
99	Compile list of Test Results and documents	0.25 days	74, 88
100	Compile a list of Design documents	0.25 days	66, 84
101	Prepare report conclusions and print	0.5 days	97, 98, 99, 100
102	Final Report is prepared and delivered	0 days	101
	Total Days	**78 days**	**624 hours**