Case Study Series: Vol 1.1

Backdoors and Holes in Network Perimeters

A Case Study for Improving Your Control System Security

Troy Nash
Vulnerability & Risk Assessment Program (VRAP)
Lawrence Livermore National Laboratory
UCRL-MI-215398

August 2005

Backdoors and Holes in Network Perimeters

A Case Study for Improving Your Control System Security

Contents

Sponsor:

U.S. Dept. of Homeland Security
Control System Security Center

Developed by:

University of California
Lawrence Livermore National Laboratory

Note: This case study is fictional with composite elements from real-world examples and open source information. The goal of this series is to provide a neutral platform for the discussion of critical infrastructure security issues across a variety of sectors.

Introduction

The Supervisory Control and Data Acquisition (SCADA) system of a natural gas utility was compromised resulting in a reduction of operation. The breach was discovered when operator interfaces became unresponsive and the system was no longer acquiring data. As a result, the system was disconnected from the network and a combination of manual operation overrides and limited fail-over to a backup server went into effect until the environment could be restored. Technicians troubleshooting the incident identified the deletion of several core application files on the primary control server as the source of the problem.

Background

Overview

The SCADA system is operated by a natural gas company serving customers (residential, industrial, and some commercial) in several communities spread across a geographically diverse region. The company handles all aspects of distribution, storage, transportation, and customer service (installation, billing, meter reading) of the natural gas which it purchases from interstate suppliers. The primary purpose of the system is to monitor and control pressure, volume, temperature, and general operating status of the various pipeline facilities, including underground storage reservoirs and unmanned compressor stations at locations throughout the service area.

Architecture

Figure 1 illustrates the network environment at a conceptual level, including the following core elements:

DMZ – A less restrictive network used for public access services like Web and FTP. In this case, the target company hosts a website for Internet presence, customer service, as well as providing some system data to industrial clients. Other systems provide applications for the company's operations. All of the hosts on the DMZ are on a separate subnet (with public, Internet-addressable IP addresses) behind the primary firewall.

Business LAN – The network used for the conduct of business operations, including Internet access, Intranet services (web, electronic mail, file sharing, printing, databases), and other application infrastructure for common business functions such as finances, human resources, market monitoring and operations, the employee desktop environment, and facility operations.

Operations LAN – The primary network where the SCADA system resides. Includes components such as the servers, operator workstations, historical archiver, alarm management, and data control (gateways, concentrators, multiplexers).

Remote Stations – The infrastructure located at the specific control point (e.g., compressor station). This is where the monitoring and control equipment resides, including the sensors and actuators (meters, valves, pressure controller, odorant injection) for the specific mechanism being monitored/controlled.

In addition, the following attributes of the overall environment are worth noting:

- The communications infrastructure is Ethernet and TCP/IP-based using a combination of leased-lines and microwave radio as the transmission medium between remote sites.
- The SCADA utilizes Unix-based systems, while all other systems (desktops, laptops, business servers) in the environment are Windows-based.

- There is a firewall and intrusion detection at the Internet perimeter between the Business LAN and the Internet. However, there are no intrusion sensors on the Operation LAN itself. Additionally, scanning activity from the Internet is ignored (no critical alerts are generated, no action is taken).
- The system includes several gas applications for analysis, data warehousing, and customer use, some of which are interconnected to systems external to the Operations LAN, but with very little security segmentation or compartmentalization between systems and networks in general.
- 802.11b wireless is used at the remote compressor stations. This allows field technicians easy access to the control network for diagnostics and maintenance purposes using their portable laptops.

Threat

Threat is defined for this case study as: *a source of danger (whether intentional, accidental, or natural) with the capability to cause harm, damage, or other operational impact to an asset (persons, property, data) by exploiting vulnerability.* Threats are dynamic, can change with time and opportunity, and are influenced by both internal and external events.

Specific threats may include an earthquake, a harmful biological agent, or an individual intent on disrupting operations. In this case, the threat is construed to be a human adversary such as a terrorist, hacker, activist, or disgruntled employee. In the following discussion, the threat will be referred to as the *adversary*.

The adversary in this case chooses to utilize a remote, cyber-based attack that does not require physical access to control system resources. While the attack described here is the deletion of files leading to a denial of service, other potential scenarios are possible, including more covert tactics such as capturing and exfiltrating data or controlling set points and operational parameters of the SCADA system itself. The attack does not necessarily have to be isolated to the specific SCADA system but could be used in support of a coordinated "swarming" attack[1] using multiple exploits (both physical and cyber) in order to maximize the impact of the attack and further complicate recovery and response efforts.

Example Attack Sequences

In the case of our target company, we will focus on two attack sequences for achieving compromise of the environment. The first represents a *backdoor* that completely circumvents perimeter defenses while the second involves a *hole* that penetrates through perimeter defenses.

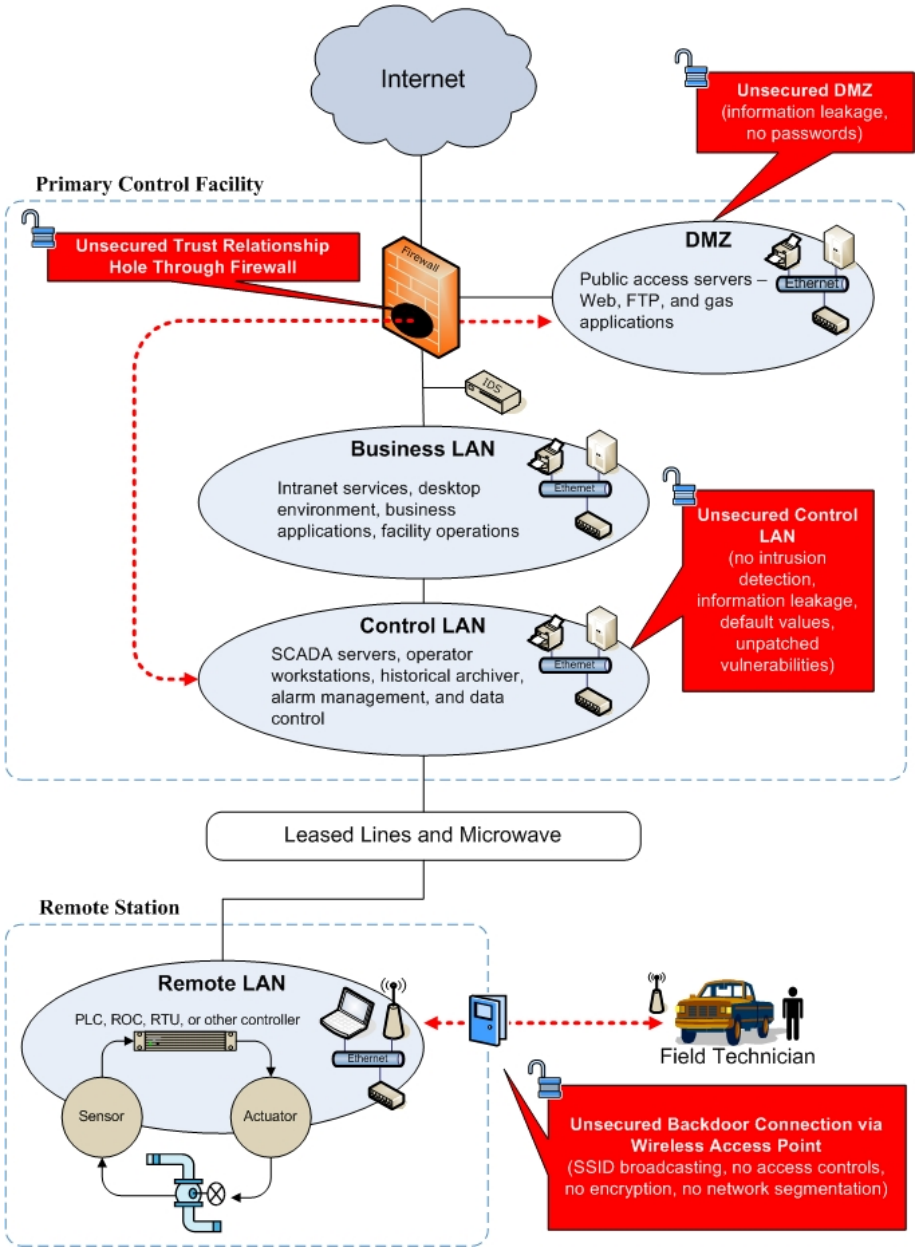

Figure 1

Attack Sequence #1 – *Infiltration through the Wireless Access Point*	
STEP 1	The adversary becomes aware of the wireless access point at the remote facility (through reconnaissance, social engineering, insider knowledge, or wardriving). From a parked vehicle outside the property fence line, the adversary uses a standard mobile rig (laptop, 802.11 wireless network interface card, range-extending antenna, and discovery software) to determine the specifics of the wireless network. Signal strength, WEP usage, and the MAC address and SSID used by the wireless access point are obtained in a matter of seconds.
STEP 2	Using the SSID, the adversary attempts to gain access onto the wireless network. Since there are no security measures (authentication, access control, or encryption) in place, the adversary is able to associate with the access point unchallenged. Additionally, a Dynamic Host Configuration Protocol (DHCP) server is active on the network, assigning a dynamic IP address to the adversary's laptop, and thereby completing the connection to the wireless network.
STEP 3	Once connected, the adversary is able to probe the network and its systems. First, host discovery techniques are used to discover active systems on the network. Where possible, the specific network infrastructure (the switches, routers, and firewalls) is identified as well.
STEP 4	For those systems that are found to be live, a fast port scan is conducted to discover what ports they have open, as well as identify the operating system and applications in use. The adversary focuses on a small subset of ports that are typically associated with common exploits or specific control system environments such as port **21**(FTP), **23**(Telnet), **25**(SMTP), **80**(HTTP), **102**(ICCP), **161**(SNMP), **502**(Modbus TCP), **1433/1434**(MSSQL), and **20000**(DNP).
STEP 5	SNMP (Simple Network Management Protocol) is found to be running on several systems. Using a SNMP utility and the default community string "`public`", the adversary connects to the open SNMP port and retrieves system information. The `system.sysDescr.0` field for one of the hosts is ***SCADA-01***. The vendor and version of the operating system is determined as well.
STEP 6	SCADA-01 is moved to the top of the adversary's list of potential target systems. Further probing identifies a vendor specific vulnerability in the operating system. An exploit is acquired from a well known hacker site and then attempted with success. The attacker gains root privileges and a command shell on the system then proceeds to recon the system for several hours before deleting the files which cause the denial of service.

Attack Sequence #2 – *Infiltration through the DMZ*	
STEP 1	From a remote system on the Internet, the adversary performs reconnaissance of the company using keywords and custom searches to identify information that can be used to support the attack. In one document retrieved from the company's website, the adversary finds the hostname and IP address of the SCADA system. The adversary cannot connect directly to the system remotely because a firewall is blocking access from hosts on the Internet.
STEP 2	The adversary then proceeds to identify all of the public IP address ranges associated with the target company using the *American Registry for Internet Numbers* (www.arin.net) and then begins to perform various scans against those addresses to identify open ports and potential vulnerabilities.
STEP 3	A Windows system is discovered on the perimeter that has TCP port 139 (NetBIOS Session Service) open, used for connecting to file shares. Access to the port is not blocked by a firewall. Additionally, system accounts are not using strong passwords (a null administrator password can be used to remotely map the system drive). Once connected, the adversary is able to read, write, and delete files on the primary file system.
STEP 4	Before attempting to attack the SCADA system the adversary first recons the compromised box. The backup *SAM* file is acquired (to run a password cracker on) and the system (logs, caches, histories, bookmarks, scripts, batch files, archives, trash bin, etc.) is searched for information that can be used to propagate the current compromise to other systems on the network. It is discovered that the host uses SSH (Secure Shell) to connect to the SCADA server.
STEP 5	The adversary can successfully *ping* the SCADA system (using the IP address obtained from the document found on the Web in Step 1) from the compromised host. While the firewall is providing limited protection to hosts on the DMZ it is not blocking the DMZ network from making connections to systems on its trusted interface. In other words, a *trust relationship* exists between the hosts on the DMZ and hosts on the protected network. With an available access pathway, all that is required to attack the SCADA is more interactive control and a vulnerability. Virus protection software on the Windows machine prevents the uploading of known Trojans onto the system, but it does not prevent the installation of a remote access tool. The adversary escalates their control of the system by installing *rconsole*, giving them more freedom and options to remotely use the resources of the compromised host (or install their own) as if they were running the tools locally at that system.
STEP 6	From the compromised host, the adversary identifies that the SCADA system is using a vulnerable version of SSH. An exploit is crafted and then attempted with success. The attacker gains root privileges and a command shell on the system then proceeds to recon the system for several hours before deleting the files which cause the denial of service.

Discussion

Beyond the specific system vulnerabilities that allowed for a compromise of the SCADA host, four general observations can be made with respect to vulnerabilities in the overall environment that contributed to the success of the attack.

OBSERVATION #1:
Perimeter security is incomplete.

Modern process control environments face significant security challenges. SCADA or other DCS (Distributed Control Systems) that operate in these environments are distributed by nature and are not concentrated in a single area that is easy to delineate and defend.

The boundaries (both physical and logical) of these systems vary. Some are localized to a specific facility, while others span large geographical regions with multiple, interconnected sites. Given the dispersed environment, the perimeter—the outermost edges, border, interfaces, interconnections—that surrounds the control system is somewhat blurred and difficult to manage from a security viewpoint. This is especially true of the cyber components of the control system, as opposed to the physical apparatus which is easier to visualize and protect — it's a piece of hardware inside a room, within a building, behind a fence, on private property, and so on. But the cyber perimeter is less tangible, and unsecured backdoors and other holes in the network perimeter are not uncommon.

Consider the wireless access point. While the physical hardware may be locked inside a secure building, the network perimeter is not just the remote station anymore, but everything within wireless range of the access point, including the hosts that connect to it. Even though access from the Internet may be heavily monitored and guarded, this connection circumvents those security controls – it's the unlocked backdoor that puts the control system at risk as long as pathways such as these remain unsecured.

RECOMMENDATIONS

1. ***Know your perimeter*** – What is the boundary of your *network* perimeter? Is it simply the border gateway that separates your control system from other external networks? Is it at the firewall? What about a modem that connects directly to the SCADA system or the field technician's laptop that gets connected to both the control network and untrusted networks (e.g., at home, hotel, or airport)? To better understand your network perimeter, consider the following:

   - Take a complete inventory of all access points, remote connections, and other ways onto your networks. Consider all relevant mediums (satellite, microwave, radio, telecommunications, wireless 802.11, Bluetooth) and locations (remote stations, vendors, customers), not just the Ethernet pathway from the Internet.
   - Develop and maintain network or system-level diagrams that inventory and illustrate these connections and the security controls that are in place.
   - Develop a process for periodically verifying and modifying the inventory as the perimeter expands or shrinks.

2. ***Defend your perimeter*** – Appropriate security controls should be added to all entry points onto your network, not just the Internet connection. In this specific case, security should be added to the wireless network connection (see sidebar for suggestions) and the trust relationship on the DMZ should be broken.

3. ***Test your perimeter*** – Table-top review, assessments, wardialing, wardriving, scanning, and penetration testing will help identify backdoors and holes, as well as uncover potential vulnerabilities in perimeter defenses.

Tips for Improving Wireless Access Point (AP) Security

Change default parameters on your AP such as the administrator password and the SSID used for the network. Changes should be performed periodically, not just the first time the device is deployed.

Turn off SSID broadcasting on all non-public APs or single AP environments that have a pre-defined set of users.

Control access to the network. At a minimum, enable MAC address filtering and use WEP encryption keys to control access to the network. For a more secure approach, consider a dedicated authentication server.

Set up the AP on its own dedicated subnet. Establish separation and security controls between the wireless subnet and the wired network(s) that it connects to using a firewall or Access Control Lists (ACLs) on the router.

Use encryption for communications. Enable WEP (preferably with TKIP or other similar enhancement). Use the largest encryption key possible and change the key frequently (if applicable). Dynamic or session-based WEP keys offer the best protection. In addition, use higher-level encryption mechanisms like VPN, SSH, and SSL for connections between hosts.

Know your network. Maintain inventories and diagrams of systems and devices on your wireless local area network (WLAN). Enable logging on systems and devices and check logs regularly. Consider deploying a wireless intrusion detection system on the WLAN.

Conduct periodic assessments. Establish a practice of testing existing wireless environments to discover new vulnerabilities and rogue devices as well as to verify that the security posture is maintained over time.

IMPORTANT: Simple security measures (like disabling SSID broadcasting, enabling WEP, or using MAC address filtering) in and of themselves will not provide adequate security against a determined adversary. However, when used in combination as reinforcing layers in a "defense-in-depth" strategy, a more comprehensive security posture is established, raising the level of sophistication and effort required for a successful attack and increasing the opportunity to detect that attack.

OBSERVATION #2:
Intrusion detection coverage is limited.

While the infiltration is seamless in both attack sequences (the attacker looks like an ordinary user, accessing system resources by ordinary means), the network reconnaissance and subsequent exploit is very **loud**, generating suspicious traffic on the network, both outside the perimeter and on the interior networks. However, this is not discovered in either scenario because there are no intrusion sensors on the control system network and traffic from the host on the DMZ is given a regrettable pass because of the trust relationship.

RECOMMENDATIONS

1. ***Verify intrusion detection coverage*** – Consider all the potential access points to each of your networks, whether they are from the Internet, a remote station, or an Ethernet jack in a public lobby or conference room. Consider key choke points and mission-critical systems. These all become potential candidates for intrusion sensors and should be considered in the overall deployment of an intrusion detection system.

2. ***Develop an intrusion detection capability*** – Beyond hardware/software controls, establish a capability (people + tools + process) to monitor and react to suspected network and system-level intrusions, as well as to maintain and tune the specific detection rulesets and logging requirements for your organization.

3. ***Evaluate the detection capability*** – Perform regular tests at all perimeter entry points, key choke points, and from random systems on the networks. Confirm that intrusion detection is working as expected – i.e., suspicious activity (like scanning) and relevant exploit signatures are flagged and the appropriate response (email or page, for example) is generated and routed correctly.

4. ***Report suspicious activity*** – Communicate with Internet Service Providers (ISPs) regarding IP addresses within their range that are being used to conduct scans against your networks and notify law enforcement of exploit attempts. Also, consider reporting incident activity to external organizations (e.g., DShield.org or US-CERT) that track such information. Forming cooperative partnerships in an effort to share information (best practices, lessons learned) and identify trends and common issues is another effective strategy. While this will not stop an adversary, it will foster an image that your organization takes violations against your security seriously and are willing to act on them.

OBSERVATION #3:
Nonexistent and default passwords were in use in the environment on both mission-critical and perimeter systems.

The use of passwords for authentication (and subsequent access to systems) is a potential area of vulnerability in every security environment. The security issues relating to password authentication (i.e., the use of weak, default, or nonexistent passwords) has consistently remained among SANS *Most Critical Internet Security Vulnerabilities*[2] since the inception of the list. The creation, distribution, usage, revocation, and other aspects of managing and protecting the keys to our network systems is an unending challenge, with many opportunities for failure.

On a control system network, the problem is exacerbated due to its mission-critical nature and the requirement for real-time operation. Operators need instant access to systems (getting locked out for mistyping a password in a crisis situation is not tolerable) and passwords often go unchanged simply because technicians do not want to risk bringing down a system that is stable. As such, shared, default, weak, or blank passwords are not uncommon in these environments. In this case, the use of nonexistent and default passwords contributed to the success of the attack sequences described here. Specifically, the following observations are worth noting:

1. There was no password required for access to the wireless network.

2. There was no password required to access the file share on the perimeter system.

3. The SCADA server used the default SNMP community string (the protocol password) "*public*".

In each case, a stronger password would not have adversely affected the operation of the environment, while significantly improving security.

RECOMMENDATIONS

1. ***Change default and nonexistent passwords*** – This requires a comprehensive look at **all** of the default and non-existent passwords used in the environment, including:

   - User accounts (administrator, root, service, temporary, guest)
   - Application passwords (SCADA, FTP, SNMP, database, web, mail, file shares)
   - Scripts & source code (Web-applications, utilities, plug-ins)
   - Network devices (access points, routers, switches, printers, firewalls)
   - Control equipment (RTUs, PLCs, IEDs, ROCs)

2. ***Develop and implement policy and procedures*** – Establish the minimum requirements for creating strong passwords, such as: length, aging, reuse, character set to be used, as well as general principles —the password shouldn't be found in a dictionary (English or foreign) or utilize personal information (such as name, birth date, or SSN).

The policy should also handle changing passwords after suspected compromise or when an untrusted user such as a vendor or technician is allowed temporary access to mission critical systems and devices. Finally, educate users regarding the policy and best practices for the security and overall usage of passwords.

2. ***Assess the environment*** – Periodically audit the passwords used in the environment to ensure that they meet policy requirements. At a minimum, systematically check mission-critical systems on a regular schedule.

3. ***Wrap additional layers of security around the exceptions*** – If a system absolutely must have a weak, blank, default, or shared password then it becomes important to add additional layers of security around that system. For example:

   - Deny remote login (only allow physical login at console/ device).
   - Use a firewall or access control list to restrict network access to a given system. In other words, the user must use System X to remotely connect to System Y (the one with the weak, default, or nonexistent password). No other system is allowed access to System Y, regardless if the password is known or not.
   - Use more robust system event logging. Determine what the normal behavior is and is not and then flag those events that are suspicious — in order to identify brute-force guessing at login prompts, access to password files, and unusual command or data patterns.

4. ***Consider alternative methods of authentication*** – Where applicable, two-factor authentication (using smartcards, tokens, or biometrics) should be considered as alternatives to using simple passwords by themselves. The advantage of two-factor authentication is that in order to access the system the user must provide something they have (smartcard, token, or fingerprint) and something they know (a PIN or Password). An adversary must acquire (or circumvent) both for the attack to succeed.

OBSERVATION #4: Sources of information leakage were present in the environment.

Unless the adversary is an insider or has otherwise acquired insider knowledge (through social engineering, coercion, blackmail, or bribery) the specifics of the network and systems prior to the attack are unknown. In the early stages of a cyber attack, the adversary operates somewhat blindly and must first discover the information, targets, and vulnerabilities necessary to execute the attack. In other words, adversaries do not magically know where your SCADA system is or what systems are vulnerable. They must discover this information through various techniques of scanning, probing, information searches, etc.

As we observed in both attack sequences, the adversary needed to gain knowledge in order to successfully attack the target. For example:

- The existence of the wireless network
- The SSID of the wireless network
- Live systems, open ports, potential vulnerabilities
- Version, brand, or type information of systems and devices
- The IP address, host name, or MAC address of target systems

If the SCADA system did not contain a descriptive name, or if its IP address was unknown, what system would the adversary attack? All of them? Or randomly, in hopes of identifying the SCADA? The adversary's work is made much easier if information leakage exists, since they may not have the capability to profile a system across a network of hosts to adequately determine if a particular one is a SCADA system or not. Packet capture and analysis or social engineering are valid secondary options, but they involve more time and resources.

The less information you give to the adversary the harder their job becomes and the more likely you will discover their attack. In this case, the attacks succeeded because the adversary was able to easily acquire the information necessary. Finding ways to control and minimize information leakage without affecting operations is the challenge.

RECOMMENDATIONS

1. ***Practice good Operations Security (OPSEC)*** – OPSEC is a process that attempts to deny the adversary information that could be leveraged to improve the opportunity, success, and impact of an attack. Some recommendations for improving OPSEC in this case would be:

   - Not using descriptive names for mission-critical systems. While it may be more convenient for managing those systems, using names like SCADA or FIREWALL or DNS make those systems prime targets in keyword searches and network discovery.
   - Minimize the amount of information regarding vendors, versions, configurations, and applications that you provide (in banners, diagrams, documents, presentations, fact sheets, annual reports, etc.), especially if those resources are accessible via the network. Identify, track, and protect those sources that do contain such information.
   - Develop a review and release process for all information that is accessible via the Web

including webpages, documents, pictures, and other media files.

2. ***Make use of obfuscation techniques where possible*** – Default banners provide the adversary with information (type, version) about the applications in use on a given system. Vulnerability and port scanners often base their findings on the information returned from a standard query. This information can be used for attack planning and exploitation. Modifying these can trick the adversary (or automated tool) into launching the wrong attack as well as increase the opportunity for discovery. Similarly, default installations (directory structures, ports used, or other patterns) can reveal information. Renaming directories (e.g., using "/apps" instead of "/cgi-bin") and using different ports for special services (e.g., using port "9999" instead of a default "8080" for a given admin web service) are examples of obfuscation techniques that can frustrate the adversary's efforts.

Conclusion

Figure 2 illustrates some of the primary recommendations from this document, applied to the environment presented in Figure 1. Primary recommended mitigations included:

- reinforcing all perimeter access points
- improving intrusion detection coverage
- hardening password usage
- minimizing information leakage

These will serve as starting points for a more comprehensive, multi-layer security posture.

While the presence of vulnerabilities on the SCADA server did introduce risk, no single vulnerability was the ultimate cause of the compromise and subsequent denial of service presented in this case study. There were several factors that contributed to the opportunity and success of the attack. The consideration of these factors, as well as the recommendations provided in this document, can help to improve the overall security posture of control system environments across a variety of sectors that face similar issues.

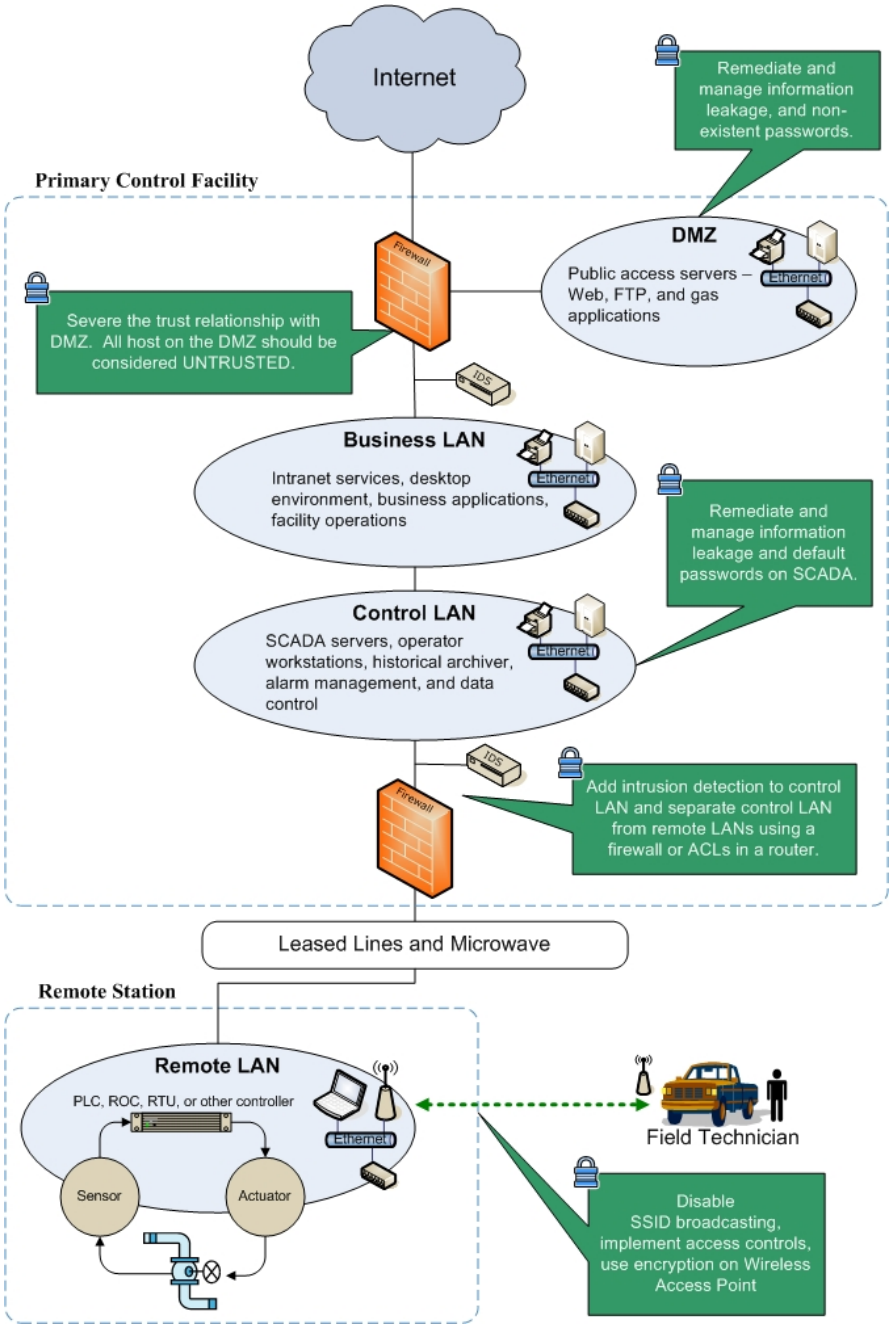

Figure 2

References

[1] *Swarming Attacks: Infrastructure Attacks for Destruction and Disruption*, a whitepaper developed by the National Infrastructure Protection Center (NIPC), July 2002.

[2] *The SANS Top 20 Internet Security Vulnerabilities*, Version 5.0 October 8, 2004 Copyright (C) 2001-2004, SANS Institute, http://www.sans.org/top20/