Case Study Series: Vol 1.2

An Undirected Attack Against Critical Infrastructure

A Case Study for Improving Your Control System Security

Troy Nash
Vulnerability & Risk Assessment Program (VRAP)
Lawrence Livermore National Laboratory
UCRL-MI-217620

September 2005

An Undirected Attack Against Critical Infrastructure

A Case Study for Improving Your Control System Security

Contents

Sponsor:

U.S. Dept. of Homeland Security
US-CERT Control Systems 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

Computer virus incidents cost companies billions of dollars every year. While antivirus technologies for detection and containment are attempting to keep pace, the threat is constantly evolving. The attack vector is no longer simply an infected executable on a floppy disk. Email, websites, macro-enabled documents, instant messages, peer-to-peer networks, cell phones, and other interconnected systems are all potential entry points onto our networks for a wide range of *malware* [1]. Our ability to successfully defend these entry points, as well as recover in the event of a given contamination, needs improvement.

Such is the situation for the water treatment facility featured in this case study, where systems on its networks were repeatedly compromised by malware over the span of a couple days. Symptoms of this infection are first noted when network performance degrades significantly on several systems, but the actual compromise is not recognized until the Internet Service Provider (ISP) of the facility relays a message regarding a suspected worm outbreak emanating from the facility's network. The offending systems are eventually identified, taken off-line, scanned, and disinfected. Unfortunately, the source carrier (a mobile laptop) of the worm is not identified and cleaned during the initial recovery process. Even though steps were being taken to address the vulnerability issues in the environment, the day after restoring operations, systems on the network are once again infected, further compounding the overall incident. Unable to effectively defend against and respond to the outbreak results in a loss of data, disruption in operation, and ultimately substantial financial impacts.

Background

Overview

The water treatment facility is owned and operated by a public utility agency of a mid-size city. Its primary function is to supply treated surface and ground water for human consumption, fire control, industrial, and commercial usage to the region. A Supervisory Control and Data Acquisition (SCADA) system monitors processes taking place within the facility, as well as the storage and distribution components of the system to the surrounding communities. Specific functions of the SCADA include process visualization, control, alarming, and historical data logging for analysis and trending.

Architecture

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

Mobile Systems – The laptops used by employees of the facility for typical business purposes (email, remote access, analysis, project planning, scheduling) while on travel or working from home.

Business LAN – The network for the conduct of business operations, including Internet access, Intranet services (employee web portal, electronic mail, file sharing, printing, databases), and other application infrastructure for common business functions such as finances, human resources, 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 (e.g., gateways, concentrators, multiplexers).

Secondary Operations LAN – A secondary network where minimal control system functionality and infrastructure reside. Specifically, this network is used for simulation, testing, and development.

Control Point – The infrastructure located at specific control points (tower, booster station, well, or storage tank). This is where the monitoring and control equipment resides, including the sensors and actuators (flow and pressure controllers, gauges, analyzers, valves, and pumps) for the specific mechanism being monitored/controlled. In this case, there are control points at the local facility as well as at remote stations in the water storage and distribution system throughout the community.

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 fiber, leased-lines, and microwave radio as the transmission medium between remote sites.
- The facility uses private (non-routable) IP addresses for systems on its internal, trusted networks. Specifically, the Business LAN uses `172.16.x.x` (Class B network), the Operations LAN uses `192.168.0.x` (Class C network), and the Secondary Operations LAN uses `192.168.1.x` (Class C network). *Network Address Translation* (NAT) is used to provide the necessary mapping between the internal private address space and external networks (i.e., the public IP addresses provided by the ISP).
- All systems in the environment (laptops, desktops, servers) including the SCADA system use a Windows-based operating system.

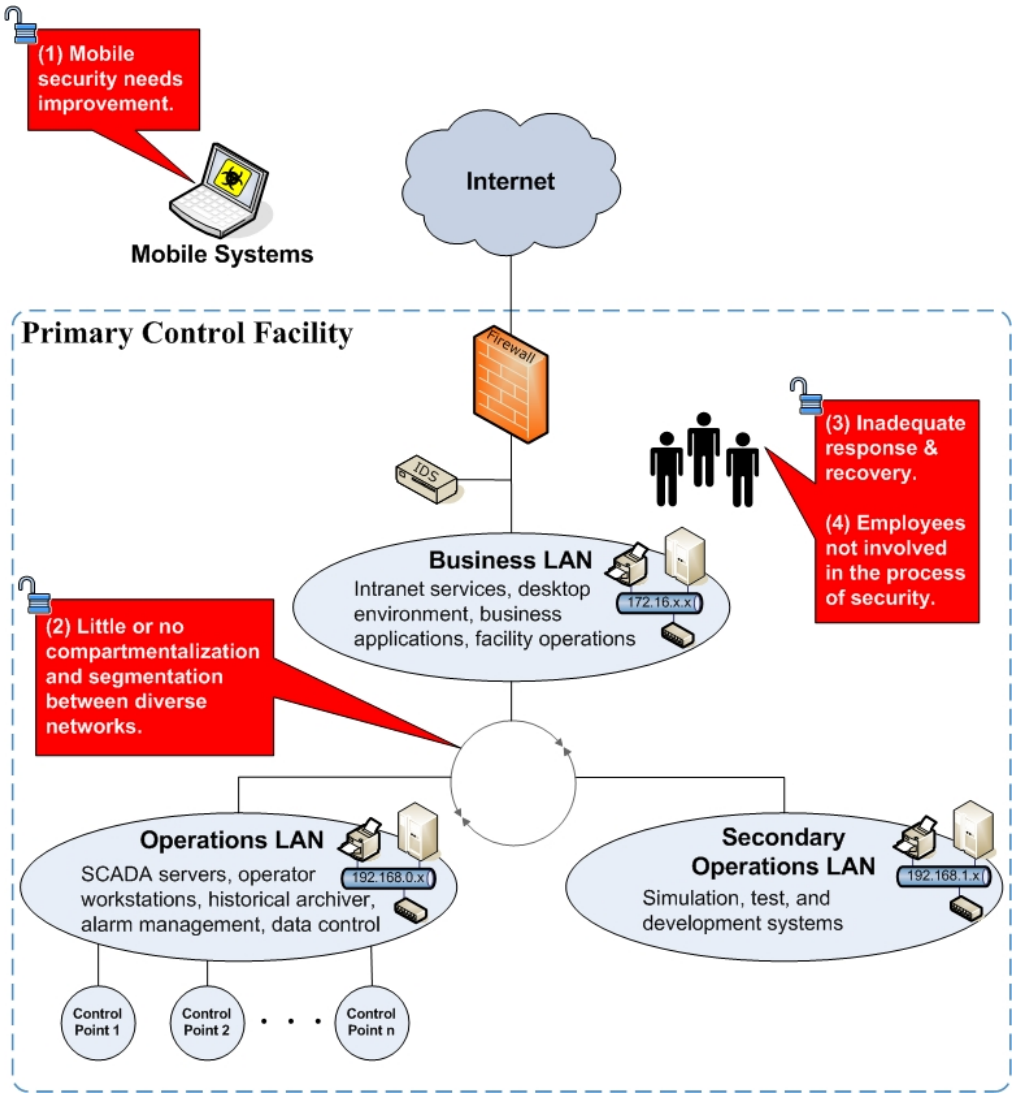

Figure 1

- There is a firewall and intrusion detection at the Internet perimeter between the Business LAN and the Internet. Virus protection is on individual desktops, but it is not centrally managed. There is no virus protection at the network gateways or on the mail server.
- Laptops in the environment have virus protection, but no other security controls. Their security configuration is verified upon deployment, but rarely (if ever) after that. Patching, virus signature updates, and virus scans are the responsibility of the end-user.
- There is very little security segmentation or compartmentalization between systems and networks in general – everything behind the firewall is considered trusted.

Threat

Threat is defined for this case 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 things like an earthquake, a harmful biological agent, or a terrorist cell intent on disrupting operations. In this case, the threat is computer malware. Specifically, a worm that exploits vulnerable systems via network connections, disabling security controls, and replicating in such a way as to significantly impact network resources.

The actual "attack" is *undirected.* The author of the worm does not target the facility specifically, but seeks to impact as many hosts across the Internet as possible. This worm's primary purpose is simply to mass replicate and the water treatment facility is just a casualty of that event. However, other more malicious scenarios are achievable. The malware could just as easily have been designed to collect and exfiltrate sensitive information, delete or corrupt file systems, or create a backdoor on infected systems, allowing an adversary direct access to resources, data, applications, and control.

Attack Sequence

Even though the water facility was not explicitly targeted, it became one of the incidental victims of the deliberate, widespread attack by the worm's creator. Nonetheless, specific circumstances and events were necessary in order for the attack to infiltrate the particular facility. These are summarized below by the following attack sequence:

Attack Sequence	
1	A malicious programmer develops a worm that exploits a specific vulnerability and can self-propagate itself over networks with vulnerable systems. The programmer deploys it on the open Internet without any particular target in mind, the only goal being maximum infection.
2	An employee of the water treatment facility uses a company-owned laptop outside of the workplace (at an airport, hotel, or from home). In particular, the employee frequently connects to a wireless network hotspot at a local coffee shop before work to read daily news blogs and check email. The morning of the outbreak, the worm (from an infected host connected to the wireless network) scans the local coffee shop subnet for vulnerable systems, propagating itself to the laptop in a matter of seconds. Elements of the worm's attack are as follows:

Infection	1. On the infected host, the worm creates a mutex [2] so that only one instance of itself runs on the system at a time.
	2. It copies itself to `%Windir%\` (by default `C:\Windows` or `C:\Winnt`)
	3. It adds the name of its executable to the following registry keys so that it will run each time Windows starts:
	`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`
	`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices`
	`HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run`
	4. It attempts to disable security controls by ending processes associated with common virus protection and firewall software.
	5. It begins the process of distribution.

Attack Sequence — *Continued*

2

Distribution

1. The worm determines the IP address of the host system (the base number from which the attack spreads). IP addresses of subsequent targets are generated using various address pools:
   - First, the local subnet (by sequentially incrementing/decrementing the last octet of the base IP address)
   - Then variations of the base IP address (replacing the last 2 or 3 octets with a random number)
   - Finally, any potential IP address across the Internet (all four octets are replaced with randomly generated numbers)

   An infected host will not scan the same system twice, but the attack is not coordinated so multiple infected systems can end up replicating scans depending on the random numbers that are produced.

2. For each new IP address, the worm sends packets to a target port. The target port is the port where the vulnerable service is listening for a connection. For example, infamous worms like:
   - Blaster — targets TCP port **135** (DCOM RPC)
   - Sasser — targets TCP port **445** (Microsoft-ds)
   - SQL Slammer — targets UDP port **1434** (SQL Server Resolution Service)

   For the attack to succeed, the port must be open, accept a network connection from the infected host (i.e., no firewall or filter is blocking), and the service must be vulnerable to the adversary's exploit code.

3. The exploit code is crafted by the adversary to take advantage of a given vulnerability or bug in the service. If the exploit succeeds, the payload of the worm is delivered and executed over the connection (e.g., code can be run with Local System privileges, programs can be installed, system settings can be modified, etc.). In this case, the worm copies itself to the system and repeats the cycle of infection/distribution.

3

Later that morning, the employee briefly connects the laptop to the Secondary Operations LAN at the facility to download simulation data for work-related purposes. Within minutes, several systems are infected on the **192.168.1.x** network. The laptop is disconnected and unused for the remainder of the day, but the damage is already done. The worm attempts to spread to other local subnets in the Class C address ranges of **192.168.x.x**, between **192.168.0.x** and **192.168.255.x** (in random order). The only other valid IP address range at this facility is the primary control network (**192.168.0.x**), which eventually is scanned and vulnerable systems are exploited. At some point, infected systems begin to propagate the worm outside of the facility by scanning thousands of random IP addresses across the Internet, increasing overall network traffic several fold. As a result of these activities, network performance degrades significantly in the control system environment.

4

A technician at the ISP for the facility is aware of outbreaks of this particular worm and begins to check traffic logs for possible signs of contamination. An email alert is sent out to customers that appear to be infected, before complaints are generated by other network providers. The water treatment facility receives one of those emails, but it does not get routed to the appropriate cyber security personnel until mid-afternoon.

5

Once the attack is realized, it takes some time to contain and recover the environment at the water treatment facility. Control system administrators scour the Web to understand the problem and its solutions before beginning the process of identifying infected systems, taking them off-line, scanning them, disinfecting them, and ultimately restoring them back to stable operation. They are able to determine that the outbreak is isolated to the control networks (**192.168.0.x** and **192.168.1.x**), but the recovery effort is not without its own problems. During the process, control data is lost, aspects of the SCADA operation are forced to manual overrides, and a general loss of functionality in key services is felt across the entire facility. By late evening, the environment is restored and steps are taken to patch vulnerable systems and install the latest virus definitions on remaining mission-critical systems on the control networks that were not affected by the outbreak.

6

The employee who inadvertently transmitted the worm is not aware of the specific problem (only knows that the networks were operating sluggishly and technicians were troubleshooting the issue well into the evening). The next morning, the employee connects the laptop to the Business LAN to synchronize data between the mobile system and a desktop system. Since the Business LAN was not infected the first day and therefore not part of the response process, the scenario is repeated as the worm successfully infects multiple systems in the **172.16.x.x** address range before spreading out to random hosts across the Internet.

Discussion

Beyond the particular system vulnerability that allowed for the worm to compromise each 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:
Mobile security needs improvement.

Mobile devices (such as laptops, PDAs, smartphones, and tablet PCs) extend the boundary of our network perimeters. Unlike the immobile network architecture at our facilities, these devices transport elements of our company to the outside world every time they leave those facilities, whether they connect to a network or not.

As the outermost edges of our network perimeter, these assets carry sensitive company information that is susceptible to loss, theft, and damage while abroad. But, the risks are not isolated to the devices themselves or the information they contain. When they are on external, untrusted networks (such as the one at the coffee shop) they are highly exposed to a variety of attacks and situations where they can become easily compromised (by either physical or cyber access). This makes them a potential attack vector and backdoor onto our networks. Instead of considering these as untrusted hosts on our border perimeter, more often than not, we allow them to connect back onto our internal networks, unchecked and unchallenged, with the privileges of a trusted host – bypassing most if not all of our security controls in the process.

Securing these devices is a challenge. Because mobile devices (by their nature) spend a portion of their time disconnected from any network or swinging between multiple networks, they are harder to monitor and manage via centralized or automated processes. Scheduled updates (for patches, virus definitions, and configuration settings) may be ignored or cancelled because the device is only used briefly to check email or visit a website. When they are not onsite and connected to our internal networks, they may be missed during network scans, security audits, and penetration tests. Thus, the burden to maintain the device's security configuration generally falls to the user.

In this case, the laptop was not sufficiently hardened to withstand any form of attack. The extended perimeter was vulnerable. And, while the device may have been secured upon its initial de-

Mobile Security Mini Self-Assessment

- ❑ What are the procedures and requirements for hardening/securing your mobile devices?
- ❑ Is the security of your mobile devices ever tested (vulnerability scans, penetration tests, assessments)? How often?
- ❑ Is software installed on mobile devices that can be used to control or access your control system? What are the security procedures? Are there any security controls in place (such as encryption of the communication channel)?
- ❑ Is operational information (system configuration files, network diagrams, training manuals) or archive data stored on the devices? How is this protected from unauthorized access?
- ❑ How do you enforce the security policies relating to your mobile devices?
- ❑ Where possible, are passwords required for mobile devices, including handheld devices like PDAs or cell phones?
- ❑ Is electronic asset identification, tracking, or recovery software used (for lost or stolen devices)?
- ❑ Is the data on mobile systems backed-up? How often? Where is it stored? How easy is it to recover?
- ❑ Are employees allowed to sync mobile devices to both home and business computers?
- ❑ Can mobile systems be used for personal use? Are users allowed to access company network resources with personal mobile devices?
- ❑ Are mobile devices connected to the internal operations LAN when onsite? Are modems plugged in and active? Are wireless network adapters enabled?
- ❑ Is the mobile computing environment centrally managed – e.g. purchases, configuration, security?
- ❑ What would be the impact of a lost or stolen device? How many devices are lost or stolen per year? What is the procedure for reporting a lost or stolen device?
- ❑ What is the procedure for password recovery (i.e. replacing forgotten passwords) off-site?
- ❑ Are there laptops that function as servers when off-site – e.g. ftp, web, file-shares?
- ❑ Are devices with camera, video, or audio-recording features allowed in the operations environment?
- ❑ Has there ever been a security-related incident involving mobile devices? How was it handled? How did it affect operations? How could have it affected operations (worst case)?

These questions can be used to help understand and characterize mobile devices in your control system environment in order to develop and refine the appropriate controls, policy, and response with regards to your security-related decisions.

ployment, its security configuration had degraded over time as new applications were installed, as files and settings were modified, and through normal day-to-day use.

Specific issues included:

- virus definitions were not up-to-date
- a full system scan for viruses was rarely (if ever) performed
- no host-based firewall was installed and activated
- unnecessary ports and services were left open
- the system did not have the latest operating system security patches installed
- the employee had no way of knowing the system was compromised

RECOMMENDATIONS

1. ***Reduce the attack surface of your mobile devices*** – The overall attack surface of a given system is defined by the way that system is configured, the services it provides, the resources that are available, and the security controls that are in place. These factors contribute to the likelihood of a successful attack, more so than any specific vulnerability. Minimizing this attack surface on individual systems helps improve security for your entire environment. See sidebar for suggestions for hardening systems against cyber attacks.

   Administrative tools (like security configuration templates, checklists, and standard operating procedures), technical tools (like security scanners and scripting used to automate tasks), and managed environments (like Active Directory) can aid in the process of hardening systems.

2. ***Don't trust your mobile devices*** – For devices that frequently connect to external, untrusted networks for any significant amount of time, it's best to assume that they have been attacked (and even successfully compromised) while on those networks.

   If they don't need to reconnect back onto your trusted networks, consider ways to keep them separate and isolated. One way to do this is to use non-direct data transfers between untrusted and trusted systems (via email or flash drive for example) that pass through malware scanners, filters, and other appropriate security controls, limiting the potential attack vector to files as opposed to entire mobile systems.

   If they must reconnect back onto your trusted networks, consider placing them in a more restricted/monitored subnet within the network or develop a procedure for "re-entry" (i.e., scan them, verify security logs don't contain any suspicious entries, and so on) as a precaution before admitting them back into the trusted environment.

3. ***Develop policies and procedures for securing mobile devices*** – These should be relevant to the mobile environment as a whole. Examples include:

   - Initial deployment (What is the baseline security configuration?)
   - Connectivity (Where/How do these devices connect both externally and internally? What security controls are used – i.e., VPN, SSL, file encryption?)
   - Maintenance (What are the procedures for patching, virus protection, vulnerability scans, etc.? How frequent? Who is responsible? Do they have adequate training?)
   - Security incidents (How will you handle lost or stolen devices or suspected compromises?)
   - How to handle forgotten passwords on the road (Does your IT help desk provide admin passwords or reset procedures over the phone without authentication?)

Tips for Hardening Your Mobile Systems

1. Remove or disable unnecessary ports and services.
2. Remove or disable sample and unused code (scripts, plug-ins) and applications.
3. Remove or disable unnecessary accounts (users and groups).
4. Change default parameters, especially identifiers and passwords.
5. Use strong passwords (on all user accounts, applications, and documents that are password-protected).
6. Utilize access permissions on files and directories for specific users/groups.
7. Define and use appropriate security settings (system security policy, web browser, email, and other application settings).
8. Implement additional security controls where applicable (two-factor authentication, host-based firewall, intrusion detection, virus protection, security event logging, file and communication encryption).

- End-of-life (How is a system sanitized or destroyed when it is no longer used?)

If procedures are too complicated or cumbersome, they won't be adopted by your employees effectively. Easily accessible and current documentation (operational manuals, "how-to" guides, quick-tips, brochures, posters, etc.) and training should be used to support your security procedures wherever possible.

4. ***Assure security configurations do not degrade over time*** – Do this systematically so as not to overlook systems. In particular, verify the following regularly:

   - A full-system virus scan has been conducted recently.
   - All security patches are current.
   - Event and security logs don't have any abnormal or suspicious entries.
   - Security controls (like virus protection and firewalls) are active on startup.

   More thorough evaluations can be performed less frequently such as comprehensive vulnerability scans and security assessments that confirm the security configuration for these devices is current and viable.

OBSERVATION #2:
There was no compartmentalization between networks at the facility.

Whether by design or not, your network and routing architecture may allow systems from other networks to communicate with systems on your control network – in other words, someone in the accounting department may have unrestricted network access to mission-critical systems on the control network. Some reasons for these pathways include:

- *Administration* – Control system administrators that use systems in their desktop environment (in most cases the business LAN) to administer systems on the control network.
- *Information Sharing* – Control information (like system telemetry and status, data that is archived or backed-up, and other information used for markets, customers, and regulatory agencies) that is pushed/pulled from the operation environment to hosts on a DMZ or external network.
- *Fail-over and Testing* – a backup operations center or simulation test bed may have interconnections to the primary control network.

While elements of this functionality may be necessary for the operation of the business, there are often no barriers or control mechanisms in place to restrict these communication and connection flows between the distinct internal environments. Often, many of these communication pathways are unnecessary. In this case, the disparate networks are blended due to a lack of specific design and for ease of administration, not because it is required by operational workflow. Without network segmentation or compartmentalization, it can be harder to address security breaches — to detect , contain, and recover from them. If the internal networks are interconnected and not segmented properly, a successful compromise on one host can easily spread, exposing more hosts than necessary to the attack.

RECOMMENDATIONS

1. ***Utilize separation and compartmentalization in your network architecture design*** – Develop a topology for your network architecture that is resistant to attack propagation and limits unrestricted access privileges across the environment. For example, hosts in the employee lounge have no legitimate reason to connect to mission-critical systems on the Control LAN. Containment (in the event of a breach) and separation of critical functions are your primary goals. If a host is compromised, it should be difficult to propagate that attack beyond the immediate network segment. Consider the various functions, services, applications in use, physical topologies, and security "zones" that exist or are required and build your security around those. Establish gateways and choke points between each environment for easier access control. Network diagrams can be useful aids in the visualization, planning, and maintenance of your design.
2. ***Use security controls to separate distinct network environments*** – Beyond physical hard-wiring, network switching and routing, or air-gapping, consider enforcing your segmentation design with technical controls such as Virtual Local Area Networks (VLANs), firewalls, filtering, and access control mechanisms in routers and switches. Also, just as important as controlling what can or cannot get into these networks, one should understand and control what is leaking or broadcasting out. In other words, make sure your security controls handle both *ingress* and *egress* of communication flows.

3. ***Evaluate network connectivity across the environments*** – Test connection pathways from the various networks you establish – e.g., can a desktop on your business LAN *ping* the SCADA server? Can hosts in the control room connect to Intranet services in the HR (Human Resource) department? Can a print station in Engineering map file shares on operator terminals? What packet traffic does a remote facility see on the network? In general, verify that security controls work as expected and the desired segmentation topology is enforced.

OBSERVATION #3:
The response and recovery capability needs improvement.

The ability and mechanisms to deal with a known (or suspected) compromise are not always well established. This is an area of concern, since the impact of even a simple compromise can be exacerbated without adequate response and recovery mechanisms in place.

It is inevitable that some attacks will succeed. Instead of trying to defend against all possible attacks, we should consider putting some resources into developing solid response and recovery mechanisms so that we can effectively restore operations should an event occur. In this case, the facility was not able to deal with the worm outbreak, either at initial response or at various stages during recovery.

RECOMMENDATIONS

1. ***Develop and implement an incident response and recovery program*** – This includes the policies, procedures, and general capability required to:
   - Stop and contain the attack — with minimal impact to continued operation.
   - Notify the appropriate personnel, mobilize a response team
   - Assess the damage — How far did the attacker or malware penetrate? Did the attacker or malware change anything or leave anything behind? Did the attacker or malware exfiltrate any information that could lead to another attack?
   - Triage — sort/rank the operational and system recovery priorities.
   - Identify and fix the source of the problem that allowed for the compromise
   - Notify proper authorities and communicate with the public (if necessary)
   - Restore operations — recover lost data or systems if necessary.
   - Verify the integrity of the restored environment
   - Take legal action (if necessary)

2. ***Integrate backup and fail-over capabilities*** – Ideally, mission-critical systems will have redundant fail-over capability, including diversity for interdependent support infrastructures like power, water, environmental control, and communications. They will also be backed up in such a way that they can recover (or be replaced) to previous operational capability with minimal loss of data and functionality. But just as important as having these capabilities is the need for them to be seamlessly integrated into the recovery effort. Some issues to consider:
   - Are spares, back-ups, redundant equipment, or repair technicians available on-hand?

Is your web browser the next entry point for malware onto your control systems?

Web browsers are increasingly utilized on control system networks as clients for web-based applications and even Internet access. Browsing web sites, downloading files from the Web, and vulnerabilities within web browsers themselves are all potential entry points that expose systems on control networks to malware infections. In practice, these connections are allowed through firewalls and intrusion detection systems uncontrolled.

A comprehensive security posture surrounding control system networks needs to address this potential pathway. Some considerations include:

- Deployment of web content filters, monitors, and blockers designed to detect/filter malicious content or control/prevent undesirable connections not related to the company's mission.
- Centralized virus protection at key chokepoints (e.g., gateways and mail servers).
- Utilizing host-based software (dedicated applications or expanded virus protection capability) that detect or prevent the installation of adware, spyware, hijackware, and other browser-based malware that infect systems just by visiting a web site.
- Hardening web browser security configurations (e.g., limiting active content and scripts that can run/install automatically and restricting cookies to trusted sites, content controls, current security patches are up-to-date, etc.)
- Educating users on how to identify and deal with a suspected compromise, including where to find information and the appropriate personnel to resolve the problem.

Onsite or offsite? How are they acquired/activated/ mobilized?

- Are there replacements or contingencies for proprietary and custom equipment, software, applications, and communication pathways?
- Are data backups located offsite? How long does it take to retrieve them? Is the recovery process tested? Is there a process for ensuring the integrity of backups before restoration (such as checksums)?
- Are there contracts or agreements in place that would allow the organization to quickly procure or borrow equipment from vendors or similar organizations if no spare is available?

3. ***Test your response capability*** – Use table-top reviews and training exercises to periodically rehearse probable scenarios. This serves as a verification/validation check of your procedures – Do they make sense? Are they achievable? Are there missing elements or areas for improvement? This also reinforces familiarity and fluency in preparation for actual events.

OBSERVATION #4: Employees were not involved in the process of security.

Employees can unwittingly be the vector for an attack during the course of normal business operations. This is especially relevant if they do not have adequate security awareness education or knowledge of incidents in progress. A simple inadvertent action can circumvent even the most rigorous security measures. While human error is unavoidable, it can be reduced through education and training.

A cyber security awareness program provides a reinforcing layer to existing procedures and technical controls. Even a minimal security education and training program can help to establish and promote a security-minded culture within the control system environment.

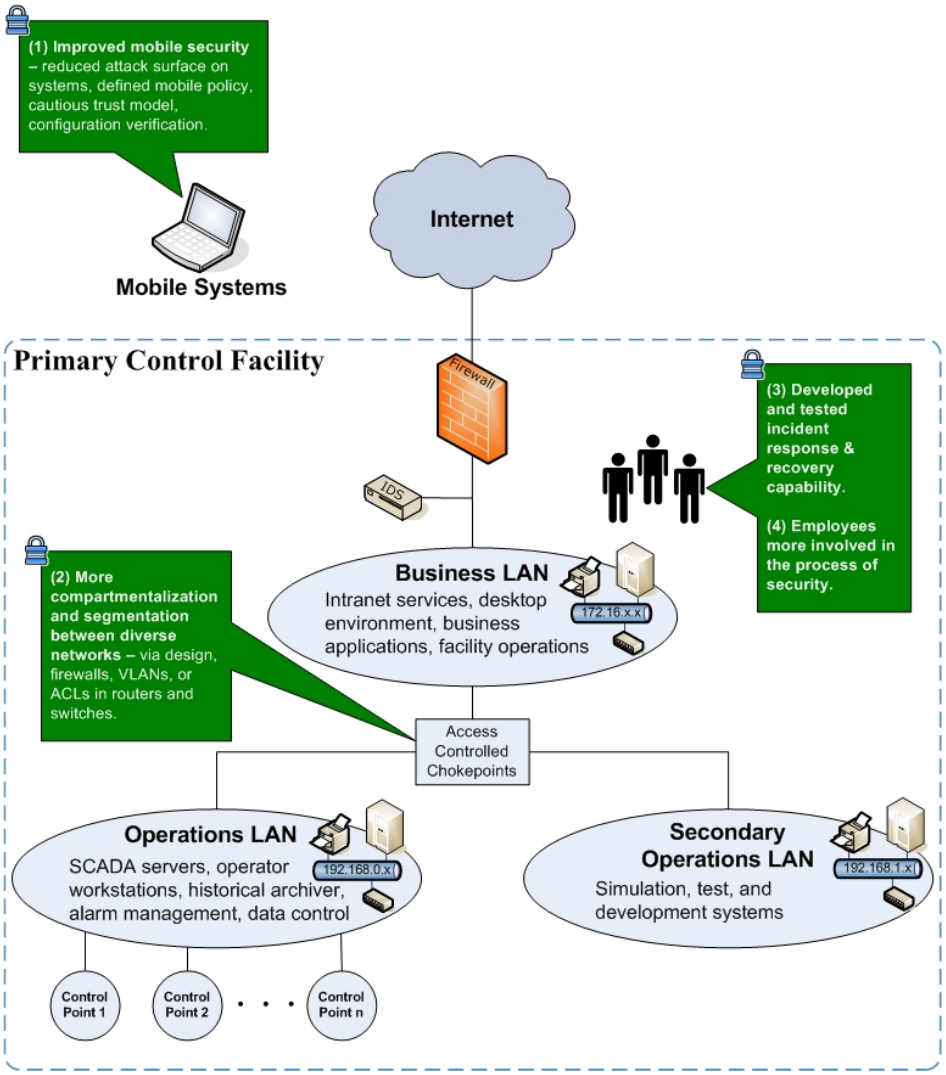

Figure 2

In this case, investing in employee security awareness (with regards to mobile security issues and responsibilities) and notifying them of the outbreak (including ways to mitigate it), may have improved the outcome of the situation in this case.

RECOMMENDATIONS

1. ***Involve employees in the process of security*** – Education and involvement should be focused on getting actionable and relevant information to the appropriate person(s) at the right time. There are many possible delivery mechanisms ranging from periodic "all-hands" meetings to prominently featured bulletins and online resource materials. For example, a timely memo discussing the dangers and proper response to *phishing*[3] or other social-engineering attacks can be a helpful limiting factor during an outbreak. Frequency (without belaboring the point) combined with relevance to the user's day-to-day activities will improve reception and application of the information received.

Conclusion

Malware can affect any network operation. Yet, beyond the infection of malware, there were several factors that contributed to the opportunity and success of this undirected attack against the water treatment facility. The ability to successfully defend entry points, as well as recover in the event of a given contamination, were two primary areas needing improvement.

Figure 2 illustrates some of the recommendations from this document, applied to the environment presented in Figure 1.

Primary recommended mitigations included:

- Improving mobile security
- Utilizing segmentation and compartmentalization in the network architecture design
- Developing an adequate response and recovery capability
- Involving employees in the process of security

The first focuses on prevention and defense at one specific entry point – the mobile environment. The others provide recommendations that help mitigate impact and proliferation to other systems given a successful attack. The consideration of these, as well as the overall issues presented by this case study, can serve as a starting point for developing a more comprehensive, multi-layer security posture across a variety of sectors that face similar scenarios.

Definitions

[1] **Malware**. (mal´wãr) (n.) Short for ***mal****icious soft****ware***. Programming (code, scripts, active content, and other software) designed to disrupt or deny operation, gather information that leads to loss of privacy or exploitation, gain unauthorized access to system resources, and other abusive behavior. Examples include various forms of adware, dialers, hijackware, slag code (logic bombs), spyware, Trojan horses, viruses, web bugs, and worms.

[2] "**Mutex**. Short for ***mut****ual* ***ex****clusion object*. In computer programming, a mutex is a program object that allows multiple program threads to share the same resource, such as file access, but not simultaneously. When a program is started, a mutex is created with a unique name. After this stage, any thread that needs the resource must lock the mutex from other threads while it is using the resource. The mutex is set to unlock when the data is no longer needed or the routine is finished." Definition obtained from *Webopedia*, http://www.webopedia.com/TERM/m/mutex.html, September 2005.

[3] "**Phishing**. (fish´ing) (n.) The act of sending an e-mail to a user falsely claiming to be an established legitimate enterprise in an attempt to scam the user into surrendering private information that will be used for identity theft. The e-mail directs the user to visit a Web site where they are asked to update personal information, such as passwords and credit card, social security, and bank account numbers, that the legitimate organization already has. The Web site, however, is bogus and set up only to steal the user's information." Definition obtained from *Webopedia*, http://www.webopedia.com/TERM/p/phishing.html, September 2005.