Industrial information system security Part 2

Malware protection for industrial automation systems

Martin Naedele, Rolf Vahldieck

The previous part of this three-part tutorial on information system security in industrial networks introduced the basic terminology and explained the need for security measures. This need was, in part, motivated by a number of reported incidents where worms infected automation systems. There is no doubt that worm infections pose a real threat for networked automation systems today.

This part of the tutorial explains the different types of malware and suggests how an automation system can be defended against them. A case study is presented where some of the suggested mechanisms are applied in a real-world automation system.

Protecting their networked automation systems against viruses, worms, and related threats (so-called malware) is one of the foremost security concerns many enterprises must deal with. As the incidents discussed in the first part of this tutorial [1] have shown, this is indeed a serious issue. Damage caused by malware may include:

- The obstruction of communication between hosts on the LAN because of high loads on the network.
- Host operating system crashes and forced shutdowns.
- Permanent damage to application and data files stored on the hard disks of those hosts.
- Malware can also send out information from the system to outside destinations or open a backdoor into the system, allowing remote control of this host by an attacker.

Any of these symptoms of a malware infection are unacceptable in a manufacturing and process control environment.

This article looks at different strategies that can be used on technical and procedural levels to defend against such threats.

What types of malware are there? And how do they infect a system?

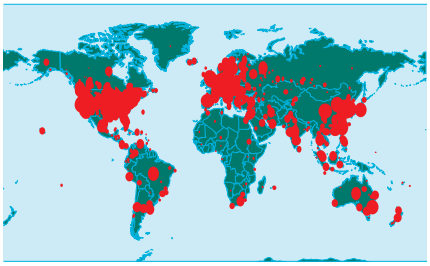
There are four main types of malware, though actual implementations may have characteristics of multiple types [2].

A *virus* is a program that infects a host as part of a legitimate data transfer, and is generally attached to another message or file. In the pre-network days of computing, hosts were infected with viruses via floppy disks. In particular, the virus was often situated in the part of a disk that is read and executed whenever the computer was started (so called boot sector viruses). Nowadays, viruses spread mostly via executables or documents containing so-called active content (macros, scripts) sent over the network, for example as attachments to emails, instant messaging, or as part of web pages. Infections via portable storage media do happen, but more rarely.

The action a virus executes when it is invoked is called its payload. A virus payload may display a message on the screen or it may completely take over and manipulate the display. It may change or delete random files or even the entire file system, or it may crash the system. Its most characteristic action is that, in addition to other malicious actions, it spreads by attaching itself to other programs and documents on the host.

A virus relies on one or even multiple user actions, ie, starting a program, surfing to the web page or inserting a portable storage medium to be propagated and/or activated. Of course, in most cases the user is not aware that such actions will activate the virus.

1 The geographical spread of Slammer within 30 minutes of its release. The diameter of each circle is a function of the logarithm of the number of infected machines, so large circles visually under-represent the number of infected cases in order to minimize overlap with adjacent locations. For some machines, we can determine only the country of origin rather than a specific city. [Figure and caption taken from [3]].



A *worm* is very similar to a virus but the major difference is that it does not require a user action for propagation. A worm can be regarded as a self-propagating virus, copying itself from host to host via the network. Typical worms use the e-mail system to propagate – they mail themselves from the infected host to recipients taken from the user's e-mail address book, or they directly scan other hosts on the LAN or network addresses on the Internet for potential entry points and then open a connection to complete the transfer.

As worms are not activated by user actions they are often located in the host's memory and can be removed by rebooting. To date, few worms have had a payload that has caused permanent damage to the infected hosts, though, of course, this is not a guarantee that future worms will not be destructive. Worms often overload the networks around the infected host as they spread to other destinations 1 and Table 1.

A *Trojan* (like the proverbial horse) is an application that camouflages itself as a harmless piece of software but in fact has an additional malicious functionality. A Trojan can be regarded as a virus without a propagation component. A typical function of a Trojan is to provide a backdoor into a system for remote control by an external attacker. For this purpose, the Trojan wants to remain undetected for long periods and therefore has no interest

Table 1 Major worm/virus epidemics in recent years [4]

Name	First seen	Means of propagation	Size of epidemic and damage type
Sasser	30 April 2004	Vulnerability in Windows LSA service	>= 1,000,000 hosts infected (repeated shutdown/reboot)
Blaster	11 August 2003	Vulnerability in Windows DCOM	>= 500,000 hosts infected
Slammer	25 January 2003	Buffer overflow in SQL Server	>= 75,000 hosts infected (network saturation)
CodeRed	19 July 2001	Buffer overflow in IIS	>= 350,000 hosts infected (network congestion)
ILOVEYOU	4 May 2000	VB script attachment processed by Outlook	>> 10,000,000 hosts infected
Melissa	26 March 1999	Mail attachment; on opening sending itself to Outlook addressbook entries	>= 100,000 hosts infected

in causing obvious malicious effects noticeable by a user.

In fact certain Trojans, so called root kits, modify significant parts of the operating system to prevent detection. They may replace, for example, operating system utilities that list files on the hard disk or show open network connections by versions that hide the Trojan activity.

A special kind of Trojan is spyware. Spyware applications are active in the background and transmit certain information from the infected host, such as software applications installed on the host, or web pages visited, to outside recipients. Like Trojans, *spyware* tries to stay undetected over extended periods of time.

How to prevent an infection?
Generically speaking, malware infections can be prevented by limiting their means for propagation and communication.

To prevent virus infections, the different ways a virus infected file enters the system must be controlled. Fortunately for automation systems, it is possible to eliminate most of these attack paths [5]:

E-mail or instant messaging (IM/IRC) attachments:
there should be no incoming e-mail or IM/IRC traffic to hosts on the automation system network. This can be enforced by not installing any e-mail or IM client applications on the computers, or disabling those that are part of the operating system. Additionally the corresponding protocols should be blocked at the firewall(s) between the automation network and outside networks should be blocked. If there are operational reasons for receiving e-mail and using IM facilities in the control room, a separate computer can be used which is not connected to the automation network but directly to the enterprise intranet. This computer may also have access to outside communication networks, for instance via dial-up modem or ADSL. Also on this computer the e-mail or IM client applications should be configured so as not to execute any active content. In addition filtering applications should be used to remove attachments, scripts and other dangerous ingredients from any incoming message.

Web browsing:
like incoming e-mail, web browsing to arbitrary sites on the Internet from any host in the automation network should be prohibited. As de-installing a web browser is often not seen as a realistic option, filtering at the firewall and VPN setup can be used to ensure only a small number of pre-approved websites – such as reference documentation on the enterprise intranet or a vendor support site – can be reached. This also mitigates the threat that users will work around the email client restrictions mentioned in the previous section by using web based email clients. Again, the use of separate hosts not connected to the automation network is suggested for web access. All downloaded content should be filtered for active and potentially malicious elements. Ideally, only pure HTML should reach the client application.

Malware infections can be prevented by limiting their means for propagation and communication.

Drive sharing:
drive sharing between hosts inside and outside of the automation networks should generally be prohibited. This can be enforced by firewalls. More secure protocols exist for transporting data in and out of the automation network 2.

Portable media (CD, DVD, disk, memory stick, etc.):
processes, procedures, and supporting physical means (eg, drive locks) in the plant should ensure that all portable media are scanned for malware with up-to-date scanning engines and signatures before they are brought into contact with a host on the control network. Extra protection might be provided by multiple scans using different anti-virus products. Of course the host used for malware scanning should not be connected to the control network.

Data transfer applications (FTP, PC Anywhere, etc):
direct transfer of files into the control system by any of those means should be restricted to a minimum. An intermediate staging server should be used to scan all incoming data for malware. Digital signatures can help verify that the file really originates from the assumed, and trusted sender, and that it wasn't modified between the virus scan and import into the control system.

Portable computers:
every portable computer should have the most up-to-date operating system patches installed as well as a current malware scanner running. Recently, several security vendors have started to offer solutions to verify a computer fulfills certain requirements before it can connect to the network.

Externally accessible service:
every host should be "hardened". This means that unrequired services and user accounts should be removed/disabled, and user access rights and policies are set such that every user (including system accounts) only has the minimum level of privileges necessary for operations.

The dataflow architecture of the system should ideally be designed in such a way that no requests are needed from external PC clients to the control network. In this case the firewall can be configured to block all connections from the outside, including worm scans. If ports must be opened, there are several options to protect them against malware:

- If the enterprise can enforce application settings on both the client and the server hosts, ports can be configured differently than the well-known numbers targeted by worms.
- VPNs can restrict the set of permitted client hosts.
- Within the control network further firewalls and personal firewalls can be used to create containment zones.
- Non-essential services with known weaknesses should only be offered on one of several redundant servers to reduce the effect of an infection.

An additional measure to avoid attacks through an externally accessible

service is to install all updates that remove known security vulnerabilities in that service. This is not necessarily the best method because it requires a considerable effort to install each patch. Also, it is advisable to defer installation until the automation system supplier has tested and qualified the patches. Despite vendor testing at a reference installation, it can not be guaranteed that the patch will not interfere with a specific system. Most importantly, patching can not prevent "zero-day attacks", ie, attacks via vulnerabilities which are unknown until the attack occurs. Because of this, any anti-malware policy should not rely on patching alone.

How to detect an infection?

Even before it affects the proper functioning of the automation system, it may be possible to discover the infection and thus initiate countermeasures before any damage is done. Available means of doing this include host-based anti-virus tools, network and host intrusion detection systems, general network health monitoring tools that are perhaps even integrated into the control system HMI, 3 [6], as well as relatively recent technologies to detect specific worm activity in a network [7]. However, certain hosts, directories, or applications may have to be excluded from virus scanning and intrusion monitoring for performance reasons.

In any case, such mechanisms will only be useful if there is somebody who continuously monitors these systems and is trained and equipped with tools to respond immediately to contain the infection and recover from it.

How to recover from an infection?

To plan and execute an appropriate response to an infection, certain information must be collected:

- Which functions/hosts are needed most urgently and have to be recovered with highest priority?
- What down time is acceptable for the different system functions?
- Which persons have to be alerted? How quickly are they available? Who may replace whom?

Based on information about functions, interdependencies, and data flows, it may be possible to design the system architecture such that it has predefined isolation points. If the network is cut at those points, the remaining islands are at least partially functional.

Redundant functions can be distributed over several of these containment zones.

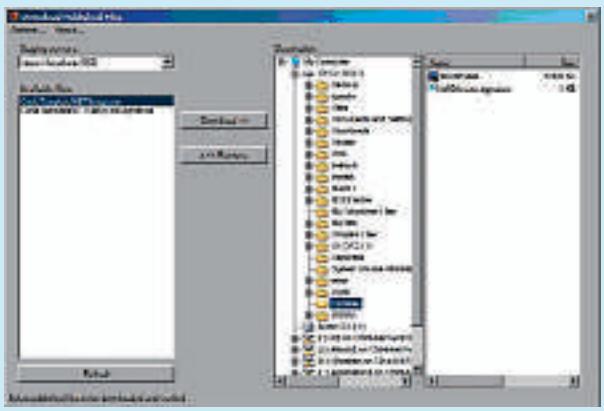
The following technical means support the rapid recovery of plant control:

- A warm standby backup host isolated from the network for functions that allow basically no outage.
- Recent and complete system backups on swappable and bootable hard disks for the immediate restart of infected systems after the infected hard disk has been removed.
- Recent and complete system backups on other media ("ghost images") for the immediate reinstallation of affected hosts.
- Local, non networked backup control mechanisms, such as manual valves as well as procedures on how to man and operate these.
- Communication means for emergency staff that are independent of the IT network, e.g. two-way radio.
- Internet access from a separate host independent of the enterprise network, eg, via ADSL or dial-up to a different provider, so that information and updates from the Internet can be obtained even while the main network is down.

Once these technical measures are in place, the following sequence of steps should be adhered to when an infection occurs:

- Isolate hosts that are known to be infected.
- Separate the different containment zones at the prepared isolation points.
- Identify and remove the vector of the infection (eg, connection to outside network).
- Once the spread of malware is stopped, connect standby backup hosts that were kept offline until that point.
- Identify the malware.
- Restore the infected hosts according to the prepared strategy (hard disk, ghost image, rebuild from the original media, disinfection). Follow expert guidance on how to remove the

2 Prototype of a tool developed in ABB Corporate Research for securely importing files into a control network.



3 Prototype of a tool developed in ABB Corporate Research to alert the process control system operator to anomalies in network behavior, e.g. caused by a worm.

identified malware. If in doubt, it is better to rebuild the system – to ensure that all pieces of the infection are really removed – rather than to rely on malware removal tools.
- Take backup hosts offline again.
- Reconnect restored hosts.
- Take measures, like changing certain procedures, to prevent the reoccurrence of the same infection.
- Reconnect to the outside network.

Case study

This case study describes the malware protection strategy for a slightly simplified version of a real ABB customer automation system. The plant control system consists of an HMI and a performance data collection and reporting subsystem.

The customer wanted access to the reporting server from PC clients in his intranet, which is also connected to the Internet. The main security concern for the customer is the threat of worm infections, especially via the reporting service which must be exposed through the firewall.

The following description highlights some of the more interesting technical malware protection mechanisms suggested for this situation 4. For brevity, procedural and standard technical measures are not described.

The operations part of the control system is isolated from the rest of the network. Any files imported into this part via portable media must first be scanned at the anti-virus station. The anti-virus station is located in the control room and is only connected to the intranet. It has a personal firewall and can only "pull" updates from the net into the anti-virus station. The reporting system is connected to the controller network via its own server. The controller network uses a non-TCP/IP/Ethernet fieldbus and is therefore unlikely to be a worm vector.

A firewall is used to block requests from the intranet to all ports on the reporting server except those needed to obtain performance data. To further reduce the attack surface, only a subset of the PCs on the intranet – those that are supposed to run the performance reporting client application – are allowed to send data to the open ports in the firewall. This is achieved using an IPSec based VPN between these hosts and the firewall. In this case, the VPN is mainly used to ensure host authentication, not data confidentiality.

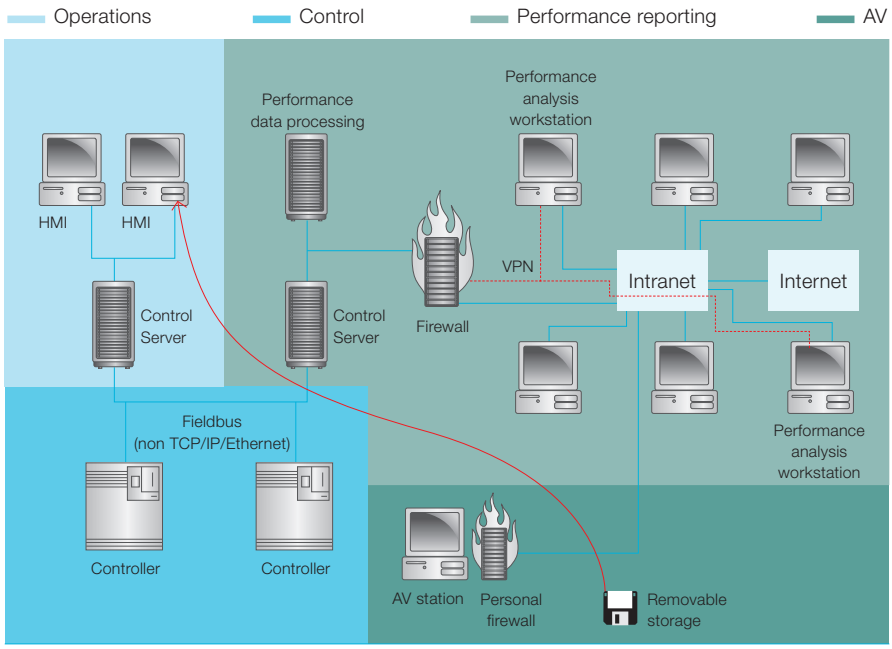
Summary

Protection against malware is mostly a "people issue". Clear policies explaining what is permitted, procedures to ensure that permitted actions are executed correctly, and training to make users aware of policies, procedures, and the reasons behind them are important factors that help protect a system against malware infections. A policy with respect to data flows between the automation system and the outside, for example, should be defined such that many of the infection vectors previously mentioned are automatically disabled, even if this causes some inconvenience for users. Technical means such as firewalls, virus scanners, content filters, and intrusion detection systems are available to support and enforce procedures.

Martin Naedele
ABB Switzerland, Corporate Research
martin.naedele@ch.abb.com

Rolf Vahldieck
ABB Automation Products, Germany
rolf.vahldieck@de.abb.com

4 Network diagram of a malware-protected automation system.



References

[1] Naedele, M. , Dzung, D.: IT security in industrial plants – an introduction, ABB Review 2/2005.
[2] T. Chen, J-M. Robert, "Worm epidemics in high-speed networks," IEEE Computer, June 2004.
[3] M. Naedele: Sicherheitsstrategien für automatisierte Produktionssysteme, in: D. Burgartz, R. Röhrig [Eds.] Information Security Management, TÜV Verlag, to be published in 2005.
[4] Naedele, M., Biderbost, O.: Human-Assisted Intrusion Detection for Process Control Systems, 2nd Int. Conf. on Applied Cryptography and Network Security (ACNS), Tunxi/Huangshan, China, June 2004.
[5] Riordan, J., Zamboni, D.: Billy Goat Detects Worms and Viruses, ERCIM News No. 56 January 2004, http://www.ercim.org/publication/Ercim_News/enw56/riordan.html.
[6] N. Weaver, V. Paxson, S. Staniford and R. Cunningham, A Taxonomy of Computer Worms, Proc. ACM CCS Workshop on Rapid Malcode, October 2003.
[7] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford and N. Weaver, Inside the Slammer Worm, Security and Privacy, July/August 2003.

Further reading

Richard Harrison: The Antivirus Defense-in-Depth Guide, Microsoft, 2004