Defending Medical Information Systems Against Malicious Software

This Paper was developed by the
Joint NEMA/COCIR/JIRA Security and Privacy Committee (SPC)

The Paper has been approved by:
NEMA (National Electrical Manufacturers Association-USA)
COCIR (European Coordination Committee of the Radiological and Electromedical Industry)
JIRA (Japan Industries Association of Radiological Systems)

December 2003

www.nema.org/medical/spc

Secretariat: NEMA (National Electrical Manufacturers Association) www.nema.org/medical
1300 North 17th Street, Suite 1847, Rosslyn, VA 22209 USA tel: 703-841-3200 fax: 703-841-5900
Secretary: Stephen Vastagh, tel: 703-841-3281 fax:703-841-3381 E-mail ste_vastagh@nema.org

Executive Summary

Medical Information Systems (MedIS[1]) of today are increasingly vulnerable to attacks by malicious software (or malware). Malware, also referred to as a virus or malicious logic, includes such things as Trojan horses, denial of service attacks, trap doors, time bombs, and worms.

This white paper informs both vendors (manufacturers and integrators of MedIS) and users (for example, hospitals and medical practices) about possible malware attacks and suggests ways to protect against them.

Possible attacks make use of exploitable MedIS vulnerabilities. The vulnerability of a MedIS depends on the kind of physical and logical access available to users and on the kind of software running on it.

Vendors and users must cooperate to meet the challenge of safeguarding the security and privacy of data in healthcare. In this white paper, the Joint NEMA/COCIR/JIRA Security and Privacy Committee (SPC)[2] offers a list of recommendations for both vendors and users to make the MedIS they produce and operate more secure.

Vendors should:

- Assure system integrity
- Employ defensive system design philosophies
- Host virus checkers where appropriate
- Respect the need for a proper configuration when offering virus checkers
- Offer security-relevant updates and technical assistance
- Respect regulatory and technological imperatives and restrictions

Users should:

- Use technical network defenses
- Prepare policies, procedures, and user training
- Restrict physical access whenever possible
- Reduce logical interconnections to the minimum
- Establish secure remote access for servicing
- Keep close contact with the vendor
- Implement the Defense in Depth philosophy

[1] MedIS generally includes all information systems directly employed in delivering health care. Examples include, but are not limited to: HIS (hospital information system), RIS (radiology information system), PACS (picture archiving and communication systems), imaging modalities, radiation therapy systems, cardiology information systems, and patient monitoring systems.

[2] NEMA is headquartered in the United States and is a trade association representing medical device and systems manufacturers, COCIR is the European, and JIRA is the Japanese trade association of such manufacturers.

1. Purpose and Scope

The current white paper discusses concepts related to protection of Medical Information Systems (MedIS) against *malicious software* (or *malware*), commonly referred to as *viruses.* Its purpose is to show how systems can be designed and provisioned to continue to safeguard patient safety, as well as the confidentiality, integrity, and availability of health data of patients, in the face of such threats. Another purpose is to identify and discuss the different types of malware beyond those commonly thought of as viruses. To this aim this white paper informs both vendors (manufacturers and integrators of MedIS) and users (for example, hospitals and medical practices) about possible malware attacks and suggests ways to protect against them.

2. Introduction

Today, the delivery of healthcare to patients increasingly relies on MedIS. Such systems rely on modern information technology (IT) to electronically collect, process, distribute, display, and store patient data. MedIS, like other IT-systems, are vulnerable to malware attacks. MedIS owners and operators have a special responsibility to shield their systems from malicious attacks. Vendors can support users in the ways discussed below. These efforts involve technology and procedures that need to be considered during the whole product life cycle by both vendors and users.

MedIS presents additional challenges not usually present in the office IT environment. Medical data must be better protected because it is needed to protect the health of patients. MedIS must also be safe, effective, and in compliance with government mandates, such as safety and quality systems regulations.

3. Malicious Logic

The term “malicious logic” or “malware” will be used in this white paper when referring to unauthorized software included with or injected into MedIS because, considering the myriad of possibilities, there are many more software threats to MedIS than the simple computer virus. Malware attacks are driven by software not supplied by the vendor that “infects“ and runs in the digital computer intended to control the medical equipment. They often interfere with the computer’s intended functionality. Malware typically attacks every computer it may find rather than explicitly target MedIS. It may cause loss or damage to data or authorized software, or even damage hardware components of the MedIS, e.g. turning off a disk drive without parking the read-heads.

Individuals with malicious intent who wish to disrupt the normal behavior of systems and cause embarrassment or harm have developed many kinds of malware. A taxonomy of security threats developed in 1994 by the MITRE Corporation [1] defines the following categories of malicious logic, based on their attack behavior. These categories have not changed in the following years although the techniques for implementation are changing rapidly. For the following overview we have selected the categories caused by malicious logic and excluded others.

3.1 Masqueraders

Masqueraders are software that appears to produce a desired functionality, but that exhibit malicious behavior.

- *Trojan Horse* is malware that appears to be a useful program, but when used it performs unintended and unexpected functions.
- Seemingly *non-executable files* can contain malware when they appear to be ordinary data. However, when accessed by the computer they become executable software that performs unintended and unexpected functions. One example is the use of malicious “macros” – specialized embedded computer instructions – in word processing files that, when accessed by the word processing application, cause unwanted behavior.

- *Unauthorized recipients* that, using malware, masquerade as an authorized recipient of sensitive data, leading to compromise of the data's confidentiality. Related to the concept of identify theft or theft of authentication.

3.2 Incapacitation

Programs in this category disable the target system.

- *Time-bombs* are programs that contain hidden malicious features. They perform an expected intended purpose until triggered by a secret event, such as reaching a significant point in time or receiving a secret message. They then perform their malicious action.
- *Denial of Service (DoS):* Attacks that create circumstances, often external to the target system, that intentionally interferes with normal system operation. DoS attacks often do not require any modification of the targeted system. They may exploit generally known and often uncorrected or unpatched defects in the targeted system by sending specific data known by the attacker to cause a malfunction. Attackers may also simply overload the targeted system by flooding it with data transmissions of a type or at a frequency intended by the attacker to cause the target system to become unavailable for normal operation.

3.3 Corruption

- *Virus.* Malware that, once incorporated, modifies authorized software so that when the authorized software is executed, it also performs the malware's intended malicious actions. Prior to the widespread use of networks, the virus was the dominant form of malware.

3.4 Misuse/Usurpation

- *Worm:* Malware that deliberately installs itself on systems connected to a computer network, repeating the installation of itself on as many targets as it can find, potentially to each and every system on the network. Once installed, the malware can attempt to perform malicious acts on its host, or simply cause DoS by its repeated attempts to install itself on connected computers. This malware attack may exploit legitimate network facilities that were intended for other purposes, or exploit defects in legitimate network services.

3.5 Implementation Techniques

Malware is increasingly a hybrid form that utilizes multiple attack methods. As computer system defenses against malware have improved, the malware itself has also been improved so as to thwart these defenses. For example, the so-called Code Red worm of 2001 utilized several worm techniques that targeted flaws in several different network services, and it also exploited a flaw in a common email program. If any of these attack methods worked, it then used a viral attack to install itself as a system service so that it could propagate itself further, leading to DoS failures.

Some malware is intended by its creator for later use to harm other targeted systems. It may utilize any of the above techniques to attack and install itself on non-target systems, intending to avoid interfering with the operation of these non-targets. Then, it sits in readiness for commands to attack other targeted systems.

4. Potential Vulnerabilities

Systems become vulnerable to malicious logic when they are placed in an environment that allows an attacker access. Access can be achieved during several phases in the life cycle of modern MedIS, including manufacture, normal operation, and service. The most invulnerable MedIS would have proprietary software, running only one dedicated application, isolated from other systems, afforded perfect physical access control, developed in a malware-sterile factory, requiring no service. Every deviation from this impossible hypothetical system results in the risks and vulnerabilities outlined in this section.

4.1 Physical Access to MedIS

A knowledgeable attacker with physical access to a system including media access, e.g., floppy disk or CD-ROM drives, may be able to infect it with malware. Highly mobile systems increase the difficulty of controlling physical access to them. This increases the likelihood of unauthorized use and modification.

4.2 Connectivity

Vulnerabilities appear when MedIS come into contact with the outside world. This may happen via direct serial port connection, modem, or network connections.

4.2.1 Stand-Alone Systems

Stand-alone systems without media access (i.e., no floppy drives, CD-ROM drives, nor network) are at the least risk of attack. They remain vulnerable to:

- Infected service tools used on-site
- Malicious or inappropriate actions by service technicians
- Malicious or inappropriate actions by vendors or suppliers during manufacture
- Malicious or inappropriate actions by users

4.2.2 Media Access

In the past media was the predominant means of interconnecting IT systems with each other. Malware-infected media can cause infection of systems that access it. So infected media was a common vector for attacking systems with malicious logic. Though still an issue, infected media is losing importance as MedIS is increasingly becoming electronically interconnected.

4.2.3 Networked Systems

Networked devices are increasingly replacing stand-alone systems to improve workflow and reduce administrative costs. They share the above risks. In addition, they also are subject to wider ranges of malicious logic that can traverse the network from one machine to the next and therefore, are vulnerable to:

- Internal forms of malicious logic, which can also be propagated from one system to another, e.g., worms
- External forms of malicious logic that operate from outside of the MedIS, e.g., malware-induced DoS.

Malware propagates between interconnected MedIS using the same technical mechanisms intended for normal communication. Networked systems may require specific services, e.g., http, ftp, SSL, and others, and correlated preassigned ports, depending on the intended use. Some services with common vulnerabilities in networked systems include the following:

- Database services and components, e.g. SQL servers
- Web services, e.g. IIS and Apache
- Directory-services, e.g. LDAP and Active Directory
- E-mail-services
- File sharing, e.g. Samba and ftp
- Printing-services, e.g. postscript and lpr
- Remote control services: e.g. SNMP

Attackers typically desire to affect the greatest number of systems they can, so most frame their service-related attacks on the most common implementations of a service. The more of these services are on a system, the more likely that system will be affected by a successful attack. Likewise, the greater the interconnectivity of devices, the more opportunity an attacker may have to gain access.

The more potential attackers, the greater the risk. Devices connected directly to the Internet have the greatest risk. It is not the kinds of vulnerabilities that change, but the potential number of attackers that increases.

4.3 Software Related Vulnerabilities

4.3.1 MedIS Using Common Software Platforms

Malware attacks often attack commonly used platforms because they are easy to find, weaknesses are known, and they have the highest impact. General purpose systems, which are based on common standards and protocols, are therefore more vulnerable than specialized systems. Despite the increased risk, the healthcare enterprise has benefited greatly from using common software platforms.

4.3.2 Device-Specific Application Software

The software intended to accomplish the dedicated task of a specific type or model of medical equipment can be termed device-specific. It likely will be written to take advantage of common protocols and to operate with a standard operating system such as Unix or Windows, but is specific and functional only with particular MedIS. Such device-specific software is not general-purpose and, as compared to other application software – like office applications for word processing or spreadsheets – is distributed in narrow communities under strict licensing and version controls. Since software of this type is less available to potential attackers, its security vulnerabilities, if any, are less likely to be known outside of the narrow vendor and user communities in which it is used.

4.3.3 Shared Use Systems

MedIS installed onto a shared use general purpose IT system remains vulnerable to all of the relevant media and network access threats of a dedicated system. In addition, it becomes vulnerable to all the preexisting or subsequent malware infections of the host system.
The MedIS vulnerability further increases when the host system also includes E-mail, enables Internet access, or offers services such as FTP, NFS, and RPC.
Additional complications arise because of restrictions on protective steps that can be taken. The dedicated use system can make extensive usage restrictions and remove unnecessary system services. The shared use system cannot make as many changes because the system must support the needs of all the different uses.

5. Defenses Against Malicious Logic for MedIS Vendors

The definition of administrative and technical measures should be started with an intended use risk and threat analysis, so that resources are utilized where most beneficial. It should consider the following points.

5.1 System Integrity Assurance

Integrity assurance can prevent or at least detect modification of the software installed in the system. Unintended or unexpected software changes might be due to the introduction of malware anywhere in the design, manufacturing, installation, and service process. We will discuss some technical approaches to assuring integrity of the system in the following sections.

5.1.1 Hardware Protection

Hardware can be used to raise the level of assurance that software has not been changed in an unauthorized way, for example, Read Only Memory (ROM) and key-locked cabinets.

5.1.2 Checksum Calculation

Checksums can be computed and compared to assure that a file is not modified. A checksum is a value calculated from the content of a file that gives the system ability to check its integrity before use. Possible implementations range from a simple parity bit check, as typically used when transmitting data over a serial line, to a 128-bit hash created when using the MD-5 algorithm. In principle, all methods share the common properties of ease in computing and low probability that correct matches between computed and expected values occur with changed data. However, the ease in computing, in terms of CPU load, varies widely with techniques, as does the probability of detecting problems.

5.1.3 Digital Signatures

Digital signatures are an extension of checksums. When a checksum is digitally signed, the probability that the original file has been changed by an unauthorized user or process can be further minimized. We note here that digital signatures require a complex Public Key Infrastructure (PKI) for reliable signature verification. The costs of establishing and operating a PKI and the cost of the threat it mitigates should carefully be balanced.

5.1.4 System Profiles

System profiles are sophisticated check summing systems that go beyond a simple list of checksums. They verify complete directory structures, e.g., including verification of file attributes, presence or absence of files, and many other characteristics of the entire combination of files present. System profiles often employ digitally signed databases and may incorporate file system checks that bypass normal operating system facilities in order to detect the more sophisticated checksum aware malicious logic.

5.1.5 Manufacturing Scan

Using virus detection software, with up-to-date virus signature files, at appropriate stages of the manufacturing process is another way to assure system integrity. A scan by an off-the-shelf virus detection tool could assure the delivery of malware free products and updates. This does not prevent a subsequent infection.

5.2 Defensive System Design

Many attack paths utilize flaws that result from common software development errors that do not introduce problems during normal operation. The most common such mistake is called the “buffer overflow” error already exploited by many malicious attacks. It permits a malicious code to overflow the allocated buffer and take control of the system.
The design methodology used by the engineering staff should help to avoid, detect, and eliminate these flaws. Specific tools and techniques should be used by engineering staff, several of which are discussed within this section.

5.2.1 Developmental Tools

There are development tools and methodologies that can analyze systems to detect and help eliminate flaws. Some of them are formal evaluation methods, such as those found in the Common Criteria (ISO/IEC 15408) [2], as well as code analysis, requirements analysis, and design analysis tools.

5.2.2 Programming Language

Some programming languages, such as Java and C#, incorporate security features that provide protection against some forms of malware attack. There are also support library and compiler features for other languages, such as C and C++, which can be used to reduce vulnerability to some forms of attack.

5.2.3 OS and Hardware Services

Some operating systems and hardware provide security features such as execute protection bits, privilege rings, etc. Applications should run with the lowest privilege practical.

5.2.4 Network Service Restrictions

Many MedIS systems are based upon common computer platforms that incorporate many network features such as logical ports and a suite of available network services. Remove or close all unnecessary features, ports, and services to eliminate potential malware attack points. For example, email or web access facilities should be deliberately removed from MedIS that have no need for these services. Their absence may be noticeable to the users who are accustomed to generic computer platforms but it should be understood as normal and desirable to increase IT security.

5.2.5 Security-focused Engineering Services

Software audits and inspections by independent personnel, including peer reviews and software walkthrough sessions, can further reduce inadvertent errors. These techniques are

commonly used for detection of functional flaws. Their scope should be expanded to include vulnerability reduction.

5.3 Host Virus Checkers

Virus checkers or virus scanning software is a class of application software that searches hard drives, disks, etc. for any viruses known by that software. Virus checkers typically consist of an executable application (scan engine) and a data file of virus patterns containing the information required by the scan engine to detect known viruses. As new viruses frequently but irregularly appear pattern file updates have to be distributed and installed frequently. After detecting a virus the virus checker performs a preconfigured action e.g., making an entry in a log-file, spawning a pop-up window with a warning text, performing an automated attempt to repair the infected file.
Virus scanners have significant drawbacks when used with MedIS. They are not a panacea for virus detection and elimination. They can consume significant resources. They may act inappropriately on false positives. Some common impacts when using virus scanners with MedIS include the following:

- Medical images, e.g., x-rays, can be damaged because the virus scanner consumes too much system resources
- Medical image files can be damaged because the virus scanner attempts to fix what it falsely identified as a virus
- Virus scanning software set to detect system behavior abnormalities can falsely identify medical software as having malicious behavior and shut down the medical software
- Pop-up windows from virus scanners can obscure medical images and medically necessary alerts

In healthcare, a proper configuration is critical for patient safety and reliable operation. Vendor recommendations should be followed when configuring and maintaining virus scanners.

5.4 Behavioral/Administrative Defenses

In addition to selected technologies described above, the vendor should:

- Keep abreast of security-related patches and updates to platform components and make them available to customers when they are deemed to add value to the intended use of the MedIS.
- Stand ready to assist customers with eradicating malware infections and malware-caused damage to their products.

5.5 Specifics and Restrictions for MedIS

Healthcare-specific regulatory and technological requirements further influence the choice of countermeasures used in MedIS as compared to those that might be used with standard office IT:

- MedIS must operate safely and effectively. Protection mechanisms must not interfere with the intended medical use of the equipment.
- When there is a failure, MedIS usually “fails open,” leaving the system usable, because it must provide continued patient care. Non-medical IT equipment usually shuts down upon failure, e.g., Automatic Teller Machines go out-of-service in the event of a problem.
- MedIS must comply with relevant government regulations (e.g. QSR) including release testing – even seemingly small updates like new virus pattern files. This has to be balanced with the desire for rapid response to new threats.

6. Defenses Against Malicious Logic for MedIS Users

Similar to the vendor the customer should start with an enterprise risk and threat analysis to define administrative and technical measures, see Identification and Allocation of Basic Security Rules In Healthcare Imaging Systems [3]. This will help to allocate resources where most beneficial and should consider the following points.

6.1 Typical Network Defenses

Many network routers and other network equipment can be configured to protect against some kinds of malware. Some examples of the kinds of actions that need to be considered in secure network design include the following:

- Operating System and router-level DoS detection and amelioration
- *Connection Authentication:* Secure techniques, whereby the claimed identity of an entity that wishes to connect is reliably authenticated, such as those specified in the Digital Imaging and Communications in Medicine (DICOM) standard and the IHE Basic Security Integration Profile, can be used to manage network access to MedIS.
- *Firewalls* are typically installed on a host or between networks primarily to prevent outsiders from accessing internal services. They are an effective and flexible tool that can perform valuable security service, but only if properly maintained and configured by well-trained people.
- *Network Virus Scanners* inspect incoming and outgoing data for known malicious logic. They may also sort out unwanted e-mails, including spam or malicious logic masqueraded as e-mail or e-mail attachments. Their disadvantages are that they can slow down the clinical workflow and suffer from false positive alerts.
- *Audit Logging & Analysis:* The MedIS will increasingly provide activity-logging information. Network operations management can utilize this information to detect malicious behavior, from both inside as well as outside sources, more rapidly. There are OS audit logs, as well as application-level ones. Both need to be reviewed frequently (based on the risk analysis) and acted upon in accordance with the enterprise security policy.
- *Intrusion Detection Systems (IDS):* Many kinds of IDSs, including “honey pots,” can be installed within healthcare facility networks. They can detect many types of potentially malicious behaviors. An IDS will provide notification, but some damage may have already been done.
- *Demilitarized Zone (DMZ)* is a logical area typically located between a private network and the Internet. Inbound and outbound connections are first intercepted by computers within the DMZ. Proxies and other security-preserving applications scan the traffic and make security and routing decisions. Traffic may be stopped, routed to other computers or proxied.
- *Monoculture Avoidance – MedIS Diversity:* Malicious logic is usually adapted to a very specific weakness of a specific type or family of IT or an IT platform. Therefore avoiding IT monocultures could reduce the number of systems affected by attacks aimed at such systems.

6.2 Behavioral/Administrative Defenses

In addition to the protective measures described above, organizations should consider the following additional processes and technologies:

- *Risk analysis and mitigation planning:* Institutions should identify enterprise specific risks, assess them, and define how to mitigate them.
- *Policies, Procedures, User Training:* Institutions should prepare administrative enterprise-wide security policies and procedures that, among other things, describe the expectations of users of their MedIS and the sanctions available if they are negligent on the one hand, or willful on the other, in their disregard of them.
- *Disaster Planning:* Damage control and remediation plans need to be in place so that responders know what to do and how to react if and when a malware attack is discovered. Backup sensitive data and software so they can be restored following a malicious attack.
- *Restrict physical access* to MedIS whenever possible by physically hiding MedIS, closing doors, locking keyboards etc. In addition, logical access to MedIS should be restricted to identifiable members of the workforces of the institution and its service providers.

- *Review all connections* of MedIS to other equipment and networks for necessity and reduce such connections to the absolute minimum. Properly configured routers by trained IT staff can deliver a high level of security.
- *Wireless communications* must receive special attention. For example improperly configured devices could inadvertently connect to an adjacent but unknown network.
- *Secure remote access* for servicing, as a time and cost saving way to receive maintenance services, can be practically and securely achieved as described in the SPC white paper Remote Service Interface Solution (A): IPSec over the Internet Using Digital Certificates [4].
- *Plan on maintaining close contact* with the vendors of your MedIS so they can offer you focused help.

6.3 Defense in Depth

The Defense In Depth concept realizes that protecting the security of an enterprise is best achieved by duplicating controls at multiple locations. A healthcare facility should establish a multi-layered defense against the risks and consequences of malware and other MedIS threats.

It is helpful to provide defenses at different layers, such as

- firewalls,
- intrusion detection systems
- virus protection,
- auditing,
- authentication
- checks and balances within the application.

In this way, if an attacker gets through one network security measure, there are additional security measures to help thwart the attack.

7. Conclusion

A single standardized solution to the issues raised by malicious logic cannot be offered in this white paper. Instead, in Sections 5 and 6, a basic set of reasonable technical and administrative measures for vendors and users has been described. Depending on the local situation each measure by itself may help healthcare providers using MedIS to increase the level of protection against the threats imposed by malicious logic. Some of these measures require in-depth analysis of the impact to safe intended use of the MedIS and thus should be the joint responsibilities of the MedIS vendors and users. Most defenses are well-established common IT tools and may be properly configured by the healthcare provider. The best approach is defense in depth. Users must take special care when defining and configuring their local security concept to avoid implementing measures that weaken the inherent security level of their MedIS.

8. Bibliography

[1] Taxonomy of Threats and Security Services for Information Systems, Gulachenski and Cost, (Working paper: Project No.:8353Z, Contract No.:DAAB07-94-C-H601), MITRE, 1994

[2] Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408)

[3] Identification and Allocation of Basic Security Rules In Healthcare Imaging Systems, Joint NEMA/COCIR/JIRA Security and Privacy Committee, 2002

[4] Remote Service Interface Solution (A): IPSec over the Internet Using Digital Certificates, Joint NEMA/COCIR/JIRA Security and Privacy Committee, 2002