Guidance and Performance Impact Testing to Support the use of Antivirus Software on SCADA and Industrial Control Systems

(ISA EXPO 2005, Chicago, IL, October 25-27, 2005)

Joe Falco
Mechanical Engineer
National Institute of
Standards and Technology
Gaithersburg, MD 20899

Michael Lochner
Summer Intern
University of Maryland
College Park, MD

David Teumim
Teumim Technical, LLC
Allentown, PA
Consultant for
Sandia National Laboratories

KEYWORDS

Antivirus Software, SCADA, Industrial Control Systems, Guidelines, Performance Testing, Computer Security

1 ABSTRACT

End-users and vendors of control systems used in Supervisory Control and Data Acquisition (SCADA) and Industrial Control System(s) (ICS) have expressed concerns that the deployment of antivirus software may interfere with the operation of time-critical control processes. This paper describes an effort to establish a set of guidelines and a test methodology for industry to help minimize performance degradation when deploying commercial off-the-shelf antivirus products with ICS. The effort is being performed for industry through a collaborative effort between the National Institute of Standards and Technology, and the Department of Energy's National SCADA Test Bed at Sandia National Laboratories. A survey was conducted of end-users and vendors who are currently using or recommending the use of antivirus software with their ICS. Information gathered from industry includes system configurations, needs and priorities for performance, as well as current practices and problems using antivirus software on control system workstations and servers. Antivirus software vendors are also providing input to this study. Parallel to the survey, NIST is conducting a series of performance impact tests using commercially available antivirus software packages and control software within its Industrial Control Security Testbed. The results of the survey and testbed work are being compiled into a documented set of guidelines and a test methodology for industry. The test methodology will be presented as a general set of test procedures to be used by industry as a starting point when developing control system specific performance impact tests. A set of laboratory-based tests that demonstrate use of the test methodology and provide example performance data is also being developed in support of this effort.

INTRODUCTION

End-users and vendors of control systems used in Supervisory Control and Data Acquisition (SCADA) and Industrial Control System(s) (ICS) have expressed concerns that the deployment of antivirus software may interfere with the operation of time-critical control processes. These concerns are one of the reasons that antivirus software has not been more widely implemented in these industries. This document describes an effort to develop guidelines and a test methodology for industry to assess performance impacts when deploying commercial off-the-shelf (COTS) antivirus products with their industrial control systems.

The effort is being performed for industry through a collaborative effort between the National Institute of Standards and Technology, and the Department of Energy's National SCADA Test Bed at Sandia National Laboratories. NIST has several ongoing efforts to address industrial control system security; one of these efforts focused on studying the effects of IT security on the real-time performance requirements of industrial control systems [2][3][4].

A survey was conducted of end-users and vendors who are currently using or recommending the use of antivirus software with their ICS. Information gathered includes system configurations, needs and priorities for performance, as well as current practices and problems using antivirus software on control system workstations and servers. Antivirus software vendors are also providing input to this study. The survey information collected along with a study conducted within the NIST Industrial Control Security Testbed is the basis for the development of these guidelines and testing methodology.

This guidance and test methodology is intended for use by individuals responsible for installing, configuring, and maintaining antivirus software on control system workstations and servers within the following company perspectives:

- End users who have never implemented antivirus on their systems for fear that it may disrupt their production
- End users implementing a different antivirus application than that specified and certified by the control system vendor
- End users concerned with the effects that antivirus software may have on the performance of their control system
- Control system vendors implementing, specifying, or certifying the use of antivirus software on their systems

The test methodology is presented as a general set of test procedures based upon the different operational modes of antivirus software. Since control systems are implementation-specific, these test procedures should be used only as a starting point when developing performance tests for specific control system integrations and must be expanded to support the specific configuration of the system under test. A set of test cases will accompany the test methodology. Test cases are being used to

develop the test methodology and present generic data to demonstrate use of the procedures and some of the adverse effects caused by using certain antivirus software configurations and practices.

The current draft guidance and test methodology document [5] includes two test cases for deploying antivirus software with operator interface software. The first test case was implemented on the NIST Industrial Control Security Testbed using a Human Machine Interface (HMI) software package typically used in a manufacturing environment. The second test case was developed at Pacific Northwest National Laboratory (PNL) on a SCADA System HMI. Future test cases will be developed for software based Programmable Logic Controllers (PLCs), data historians, and SCADA servers prior to the final release of the guidance and test methodology documentation.

A Functional Overview of Antivirus Software

The execution of a malicious virus can cause damage to the operation of a computer [6] including deletion or intentional corruption of files, corruption of system areas to prevent reboots, and the degradation of computer performance by theft of memory, disk space, clock cycles, and/or system modifications. In contrast to viruses, a worm is a self-replicating piece of software designed to exploit networks and security vulnerabilities. A Trojan horse is a malicious computer program disguised as legitimate software. Both worms and Trojan horses are often used to deliver viruses. Throughout this document, the term "virus" refers to any viruses, worms or Trojan horses as handled by antivirus software.

Antivirus software packages are designed to discover both known and unknown viruses, disable them, and if possible, reverse any damage caused by them. Most commercially available antivirus software packages offer two methods of virus detection; signature matching and heuristic analysis.

Signature matching uses a predefined set of virus signatures for thousands of known viruses. The scanning engine in the antivirus software compares the program code running on the computer against virus patterns stored in the virus signature database. The signature database must be updated frequently to ensure all newly discovered viruses are detectable. This process is commonly referred to as a virus definition update. Updates can be automatically downloaded from the Internet, or manually installed using portable media or local files. Heuristic analysis techniques use behavioral characteristics to find new, uncataloged or viruses that are designed to automatically change during propagation.

Active (also called on-access or real-time) scanning examines files, diskettes, and system areas for actions or changes including copying and saving files, collecting network-based data, and starting applications and their associated Dynamic Link Libraries (DLLs). If actions or changes are detected, the associated object is scanned for viruses using signature matching and heuristic analysis techniques.

Manual (also called on-demand) scans detect viruses in selected groups of files and/or directories using signature matching and are initiated by a user or automatic scheduler. Once an antivirus application detects a virus, there are typically several methods available for dealing with it: attempt to clean the

virus from the infected file, quarantine the file to a reserved location on disk, delete the file, or simply log the fact that a virus was detected.

Most antivirus applications are configurable by the end user. The following list describes some configuration settings commonly found in antivirus software:

- Actions for handling an infected file (clean, quarantine, delete, leave alone)
- Amount of CPU time dedicated to the antivirus application (often referred to as throttling)
- Enabling or disabling heuristic scanning
- File exclusion list (a listing of files that will not be scanned for viruses)

These settings can be locally configured, or set and administered using a centralized server. Centralized administration software is commonly available that allows remote installation, configuration and administration of antivirus software running on networked computers.

The execution of antivirus applications on a system uses processing and memory resources. Although antivirus vendors strive to keep the resources use within acceptable limits, performance effects, as well as general compatibility with ICS core applications should be assessed. One must understand all the configurable options of antivirus software (such as heuristic analysis and signature updating) and the possible effects they may have on system performance. Performance problems or system malfunctions can also be associated with scanning critical operational control files. In addition, SCADA and ICS are typically comprised of many workstations and servers working together to control a process. Performance degradation on one system could affect the performance of the overall system.

ESTABLISHING GUIDELINES

Information is being gathered from a survey of end-users and vendors currently integrating antivirus software into their ICS. This information and experience gained from testbed work are being compiled into a set of general guidelines that address the planning, installation, configuration, operation and maintenance of antivirus software. Below are some guidance extracts from the guidance and test methodology document [5], still under development.

- Check to see if the vendor of your control system software recommends and/or supports the use of a particular brand of antivirus. In some cases, control system vendors may have performed regression testing across their product line for supported versions of antivirus software.
- Obtain all available installation and configuration guidance from the control system vendor if they support a particular brand of antivirus software. Also, obtain general antivirus application installation and configuration documentation from the antivirus vendor. **Any vendor guidance specific to your control system and antivirus software compatibility should be given precedence over this generic guidance.**
- Determine which control application files should not be scanned during production time because of possible control system malfunction or performance degradation and add these files to the antivirus file exclusion list. Some control system vendors that support antivirus software will specify in their user documentation particular control files that should not be scanned.

- Manual scanning and virus definition update operations may negatively impact control system performance. To minimize performance impacts: stagger manual scans, schedule scans and virus definition updates during non-peak hours, and scan redundant control servers at different times.
- An antivirus application throttling feature is often available for manual scanning. As this configuration setting is increased, more of the workstation or server CPU time is given to the antivirus software process, giving less time to concurrently running control processes. This may result in serious performance problems. It has also been noted that these throttling settings are not always linear and may shift the majority of CPU processing time to the antivirus process at mid-level settings. These throttling settings should be minimized in the event that manual scanning is scheduled during control system operation.
- Active scanning when a control program is accessing data from a historian may cause significant performance effects since the excessive traffic in turn causes a large degree of scanning.

TEST METHODOLOGY

The test methodology is presented as a general set of test procedures for use by industry as a starting point when developing specific control system performance impact test procedures. Although Windows-based platforms are being used for development, the test methodology should be general enough to be applied to other platforms. Once specific test procedures are developed, testing should first be performed on non-production systems, including corporate IT stations, development systems, off-line laboratory systems, production system simulations and production systems in down time. End users must decide to what extent they will follow the test methodology. The following conditions warrant testing before implementation on a production control system:

- First time integration of an antivirus software package
- Upgrades to antivirus software such as the installation of a new scan engine
- Updates to virus definitions
- Reconfiguration of the control system.

The test methodology should be implemented over the entire operational range of a control system. This range would include steps such as start-up, operating capacities and shutdowns (both standard and emergency).

The major steps in each procedure are given below. More detailed procedures can be found in [5].

1 Compatibility and functionality testing
- 1.1 Lock down computer and turn off all unnecessary processes.
- 1.2 Install antivirus software or perform upgrade per manufacturers instructions.
- 1.3 Set antivirus software configurations for your control server
- 1.4 Perform health checks on concurrently running antivirus software and control system software.
- 1.5 Perform functionality testing on concurrently running antivirus and control system software.

2 Establishing a performance baseline of the control system
- 2.1 Identify system variables to monitor such as processor, network and memory and production system performance variables and select utilities used to monitor the variables.
- 2.2 Run the control system applications, the antivirus software, and the monitoring tools. Identify computer processes stemming from both the antivirus software and the control system process under test. Run applications through all modes of operation in order to identify all processes associated with it.
- 2.3 Turn off all features of the antivirus software. Check the server task manager for any antivirus application processes. Note that active (on-access) scanning may need to be turned off since it is often activated in the default configuration. Also, deactivate any automatic enabler options.
- 2.4 Collect data over typical control system usage and workload conditions over a period of time that insures all major control system processing events are captured. It is advisable to monitor resources remotely when possible in order to eliminate the additional CPU usage due to the addition of these applications on the workstation or server being tested.
- 2.5 Document baseline parameters with usage and workload conditions.

3 Performance impact test: active scanning
- 3.1 Duplicate the data collection scenario used in establishing a performance baseline.
- 3.2 Enable active scanning.
- 3.3 Test peripheral drive operations by dragging contents to the hard drive to trigger active (on-access) scanning
- 3.4 Start all secondary applications that are used periodically during control system operation to force scanning of executable program files, as well as any associated DLLs.
- 3.5 Maximize network communication traffic of data coming onto the control server under test.
- 3.6 Collect data during steps 3.3, 3.4, and 3.5 above.
- 3.7 Compare performance results with baseline performance data.

4 Performance impact test: manual scanning
   4.1 Disable active scanning.
   4.2 Duplicate the data collection scenario used in establishing a performance baseline.
   4.3 Configure a manual scanning operation on a group of directories and files located on the server hard drive and on peripheral drives.
      4.3.1 Total size of files to be scanned should be selected based on desired test time.
      4.3.2 Use many small files such as browser caches to force file open and read operations.
      4.3.3 Insert copies of a test virus throughout the directories of files to be scanned.
   4.4 Trigger data monitoring and the manual scanning operation.
   4.5 Collect data over the entire scanning operation.
   4.6 Compare performance results with baseline performance data.

5 Performance impact test: virus definition updates
   5.1 Duplicate the data collection scenario used in establishing a performance baseline.
   5.2 Perform a virus definition update via the some planned mechanism for distribution and installation.
   5.3 Collect data during the virus definition update operation.
   5.4 Compare performance results with baseline performance data.

A TEST CASE

The current draft guidance and test methodology document includes a test case for deploying antivirus software with operator interface software using HMI software typically used in a manufacturing environment. This example test case provides data for running a popular commercial antivirus software package on a PC that is also hosting HMI and IO server software. It is anticipated that similar test cases will be developed for other types of control system applications such as SCADA servers and HMIs, PLC software based controllers, and data historians and a second antivirus software package.

This test case was developed on the NIST Industrial Control Security Testbed. The testbed, designed to serve as a platform developing performance and conformance test methods for industrial control security, is comprised of several implementations of typical industrial control and networking equipment as well as relevant sensors and actuators. The testbed, shown in Figure 1, includes a water distribution implementation and factory control implementation. This test case was developed using the water distribution implementation portion of the testbed.

The test setup for this test case is shown in Figure 2. It measures the performance of commercially available HMI and I/O Server software running on a Windows 2000 PC platform with an antivirus software package. Tests were conducted on three separate PC workstations, each running duplicate software setups. Each integration of the HMI application was configured with enough tag points to bring the total baseline processor load of each of the PC systems to between 40 % to 60 %. Ethereal, an open source, packet monitoring software package, and Performance Monitor, a software tool (included with Windows 2000 and XP) to monitor system resources, were run on a separate computer to capture performance data of the servers. The three servers used were a Pentium III 1000 MHz processor with 256 MB RAM, a Pentium II 450 MHz processor with 256 MB RAM, and a Pentium II

266 MHz processor with 64 MB RAM. These computers were chosen based on industry recommendations to address the typical lower end PCs considered the most vulnerable to performance impacts caused by the inclusion of antivirus software.

Test procedures were designed for this setup using the documented test methodology. The European Institute of Computer Anti-virus Research (EICAR) test virus [7] was injected through different test vectors, and communication packets were monitored between the HMI and the PLC. The time between consecutive packets for monitoring a single sensor device was calculated and used as a measure of polling latency. CPU utilization data was also collected as a measure of performance for this test configuration.

A manual scan test procedure was designed where a 100 MB folder was created consisting of one thousand small files including copies of the EICAR virus. The folder was placed on the hard drive of each testbed and a manual scan was executed only on the test folder. This procedure was carried out several times while adjusting the throttling setting on the antivirus software. This allowed us to observe the range of effects a manual scan could have on the processor time of a control system relative to these settings. Results from this manual scanning procedure at the antivirus software's lowest and highest throttling settings are shown in figures 3 and 4 respectively. Regardless of the setting, once the manual scan began, the total processor time often reached the 80 % to 100 % range. As suspected, the antivirus software dominated the CPU time when set to the high setting. This affected the performance of other processes like those belonging to the HMI software, which were reduced from the 40 % to 60 % CPU utilizations observed during baseline operations. When set low, the manual scanning process used the remaining processor time instead of taking precedence over the HMI software, keeping the antivirus software from hindering the performance of other processes.

There is a relationship between the antivirus software's throttling setting and the time required to run a manual scan. By increasing a throttling setting in the antivirus software, the time required to perform a scan was decreased. The scan duration was reduced by approximately 20 % to 50 % depending on the testbed that was being examined. The Pentium II 266 MHz testbed displayed a 20 % decrease while the Pentium III 1000 MHz testbed displayed a 50 % decrease. The highest throttling setting on antivirus software produced a significant impact on HMI performance. The lowest throttling setting produced a lower more stable latency that mimicked the baseline data, while highest setting produced a more chaotic latency pattern usually with several relatively large spikes occurring during the manual scan. The average latency level of the high setting tends to be slightly greater than that of the low setting. This trend is most observed in the high priority data gathered from the Pentium II 266 MHz testbed. There were no performance degradations from detecting and quarantining occurrences of the EICAR test virus.

Two different procedures were chosen to examine the effects of antivirus software actively scanning files in our control system. The first procedure involved transferring files from a removable hard disk to the hard drive and the second procedure involved transferring files from a floppy disk to a hard drive. There was a 100 MB directory on the removable hard disk and a 1.2 MB directory on the floppy disk, each containing many small files. Transferring data from these media to the test system hard drive enabled the active scanning process for the antivirus software. Data from the removable hard disk active scanning procedure is shown in figure 5. The primary difference between the total and base

processor utilizations was due to the large numbers of file transfers associated with these active scanning procedures. Analysis of individual process CPU utilization data determined that the antivirus processes dedicated to actively scanning files have a minimal impact on performance. This included any detection and quarantining of the EICAR virus. The time duration of the tests was primarily dependent on the transfer rates between the media drive and the hard drive. The effects on the latency within the control system's network while actively scanning accessed files were negligible. Latency appeared to average between 1.5 and 2 seconds for all test bed CPUs and all data media.

The virus definition update procedure was conducted using a local update file since there is no Internet connectivity from the NIST testbed. An executable file was used to update the virus definitions that could be manually placed on the control system computer. The executable file was obtained by downloading it off of the antivirus software vendor's website. While the virus definition updater was executed, the computers performance data was monitored and recorded. Data from this procedure can be found in figure 6. Total and base processor times were basically the same except when update utility was carrying out its operation. The update utility consumed relatively large amounts of processor time and caused the processor to reach its maximum operational capacity for the majority of the update. This occurred on all three platforms. Since the update utility dominates processor time, there is noticeable performance loss with the HMI processes and any other processes that might be running. This is most apparent on the Pentium II 266 MHz system since its test duration was the longest. The latency was observed to remain in the base latency range of 1.5 to 2.5 seconds except while the update is being performed. The latency jumps to the 4.5 to 5 second range while the antivirus update process is running. The Largest latency spikes were observed on the Pentium II 266 MHz platform.

SUMMARY

This paper describes an effort to establish a set of guidelines and a test methodology for industry to use when deploying commercial off-the-shelf antivirus products with industrial control systems. The effort is being performed for industry through a collaborative effort between the National Institute of Standards and Technology, and the Department of Energy's National SCADA Test Bed at Sandia National Laboratories. A subset of the guidance being developed as well as an overview of the test methodology is presented. Also presented is the first of several laboratory-based tests cases that will be used to demonstrate use of the test methodology and provide example performance data. More detailed information can be found by downloading the draft guidance and methodology documentation [5].

ACKNOWLEDGMENTS

NIST and Sandia would like to thank the many end-users and vendors who are supporting this project by providing valuable input and document reviews. We also would like to thank Pacific Northwest National Laboratory (PNL) for providing pilot testing of the test methodology on their laboratory SCADA system in addition to participating in the document review process. NIST would also like to acknowledge Sandia National Laboratories for partially funding these efforts.

REFERENCES

[1] Department of Energy, National SCADA Testbed, http://www.ea.doe.gov/pdfs/nstb_factsheet.pdf

[2] Falco, Stouffer, Gilsinn, IT Security for Industrial Control Systems: Requirements Specification and Performance Testing, 2004 NDIA Homeland Security Symposium & Exhibition, Crystal City, VA, 2004, http://www.isd.mel.nist.gov/documents/falco/NDIA.pdf

[3] Process Control Security Requirements Forum (PCSRF), http://www.isd.mel.nist.gov/projects/processcontrol/

[4] PCSRF System Protection Profile for Industrial Control Systems (SPP-ICS), http://www.isd.mel.nist.gov/projects/processcontrol/SPP-ICSv1.0.doc

[5] Falco, Teumim, Using Antivirus Software on SCADA and Industrial Control Systems: Integration Guidance and a Test Methodology for Assessing Performance Impacts, Draft Document

[6] Harley, Slade, Gattiker, Viruses Revealed, McGraw-Hill Companies, 2001, pg. 151-160

[7] The European Institute of Computer Anti-virus Research (EICAR), http://www.eicar.org

DISCLAIMERS

1. Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

2. Always consult vendors of control system software and antivirus software for any available guidance. Any vendor guidance specific to control system and antivirus software compatibility should be given precedence over this generic guidance and test methodology.

3. Commercial equipment and materials are identified in order to adequately specify certain systems. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, Sandia National Laboratories, or the Department of Energy, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

4. Products undergo testing to determine the validity of the testing procedures and the range of results that can be expected with commercial systems. Any results will be reported either in aggregate, or with any vendor identifying information removed.

FIGURES

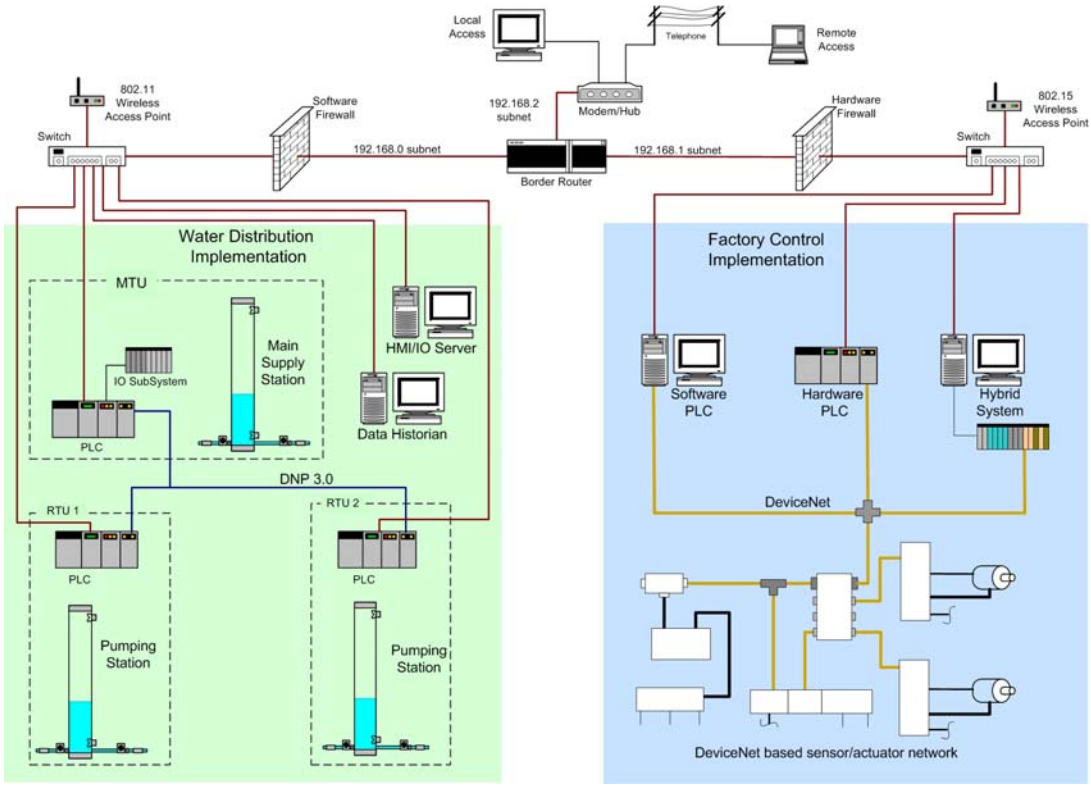


FIGURE 1: NIST INDUSTRIAL CONTROL SECURITY TESTBED ARCHITECTURE

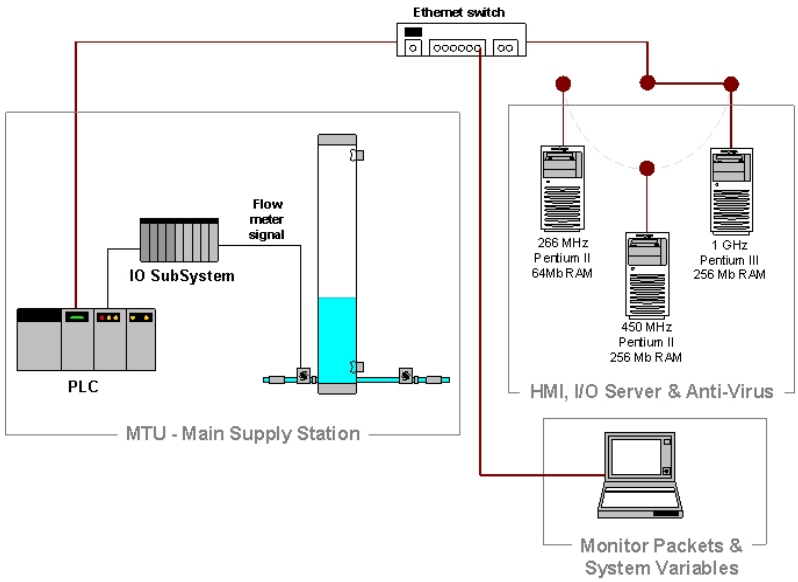
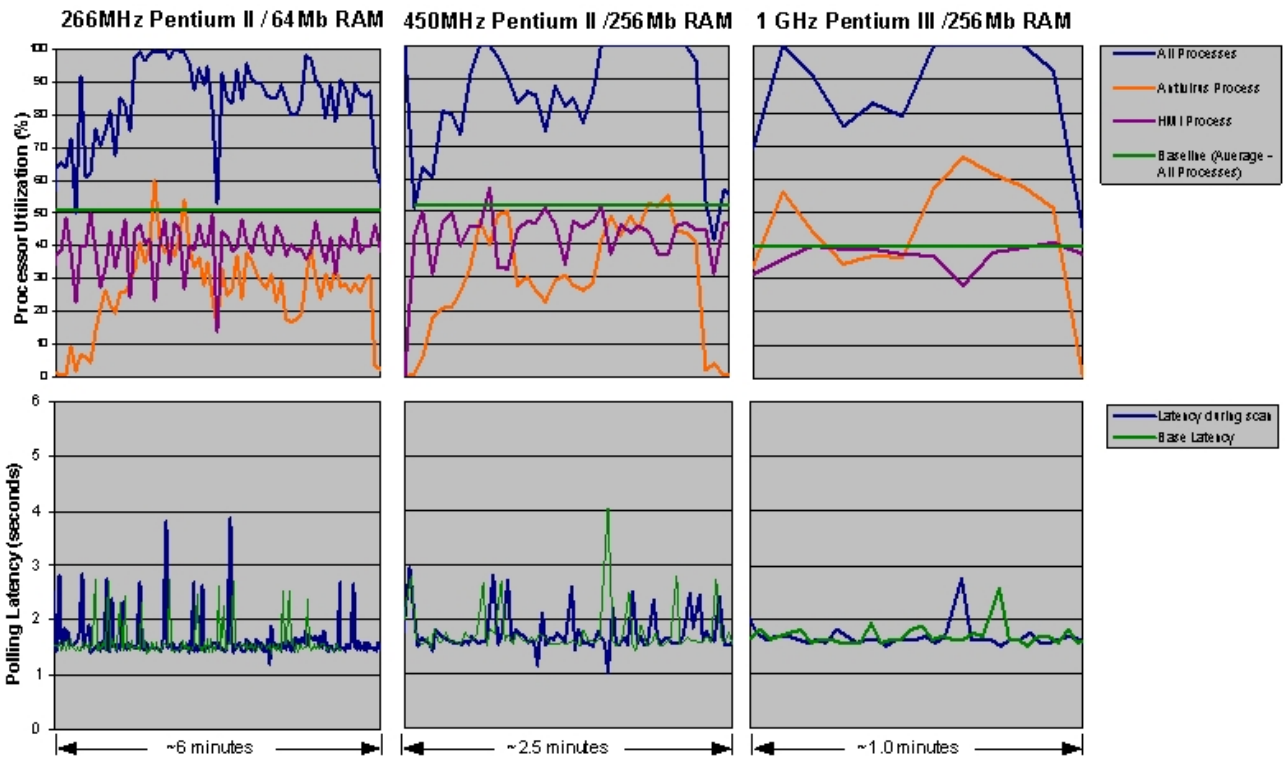


FIGURE 2: TESTBED SETUP ANTIVIRUS VS. HMI & I/O SERVER



6

FIGURE 3: TEST CASE DATA – MANUAL SCAN (LOW PRIORITY)

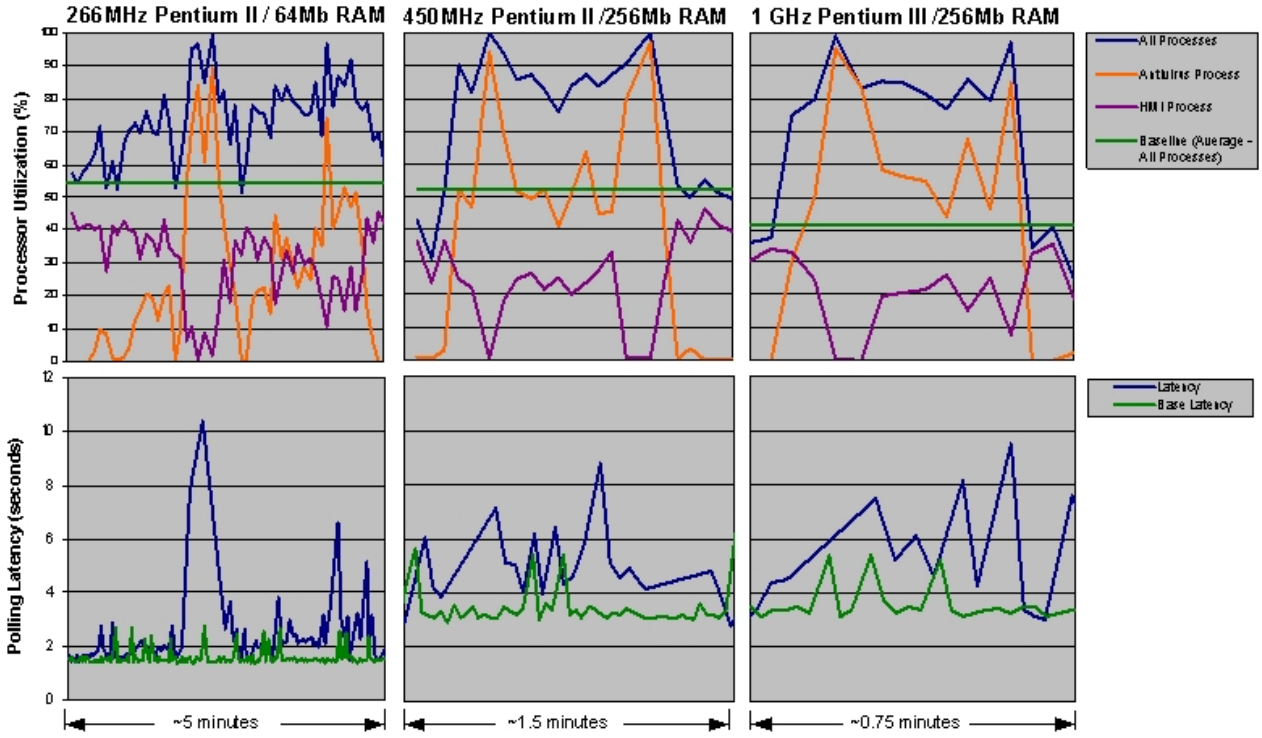


FIGURE 4: TEST CASE DATA – MANUAL SCAN (HIGH PRIORITY)

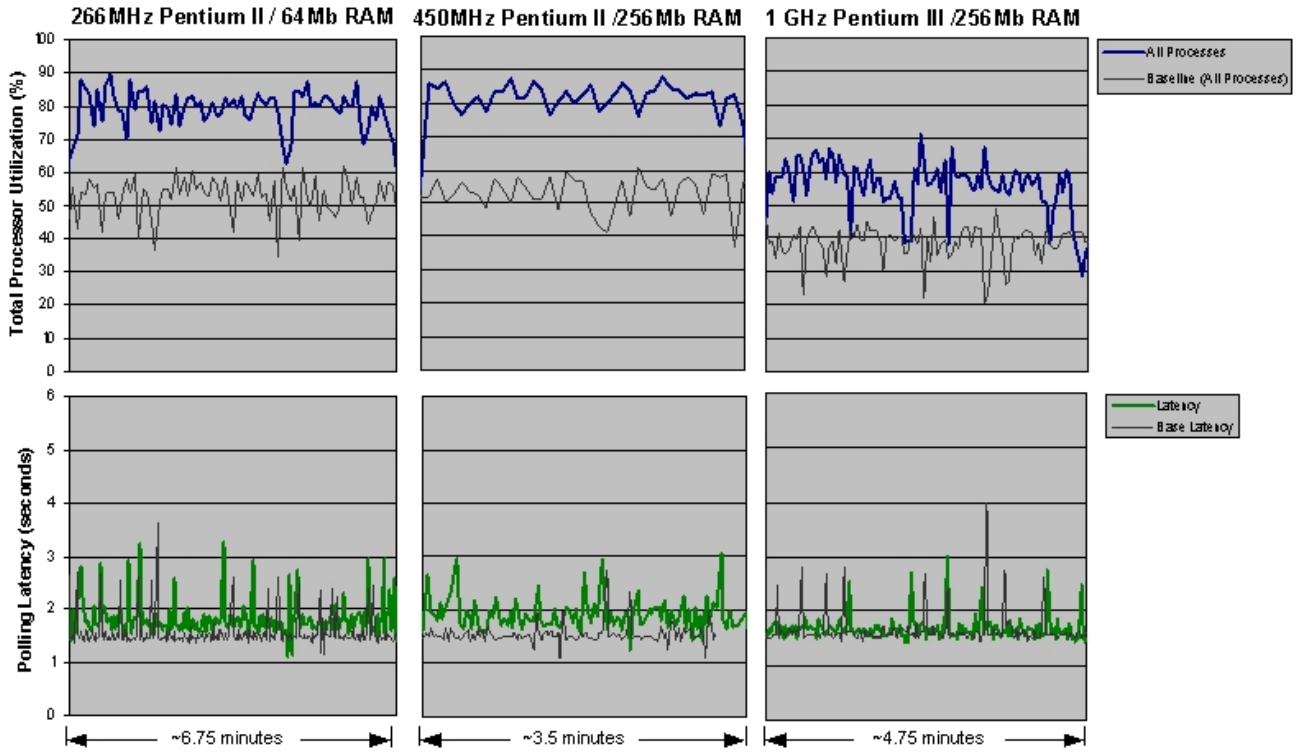


FIGURE 5: TEST CASE DATA – ACTIVE SCAN

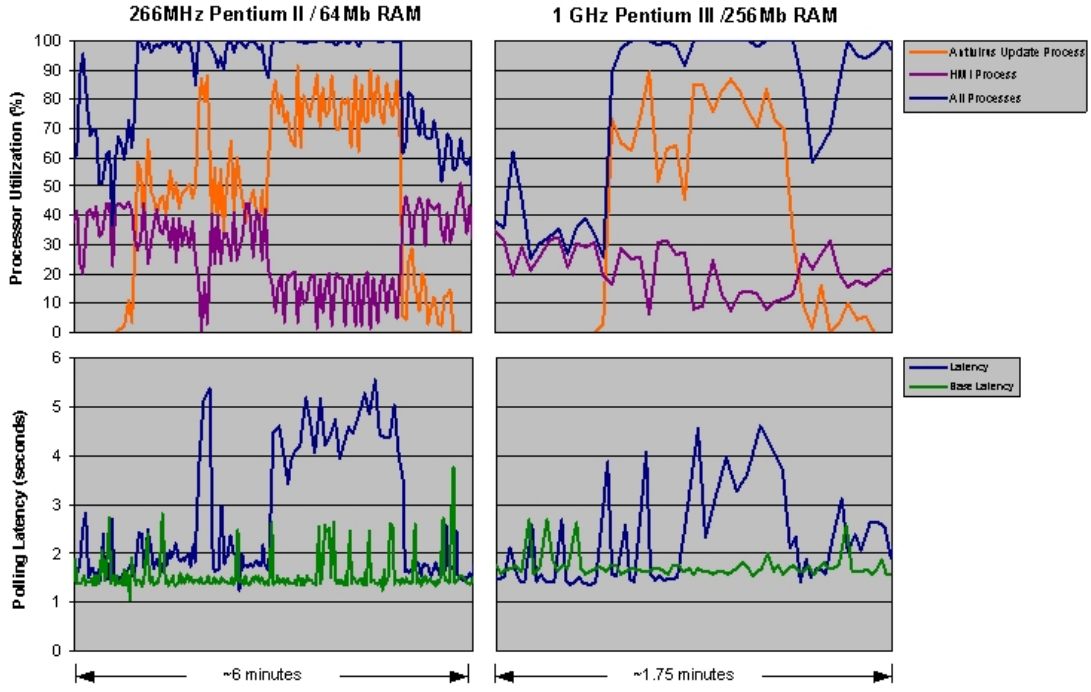


FIGURE 6: TEST CASE DATA – VIRUS DEFINITION UPDATE