

WHITE PAPER.

Resisting Cyber Attacks Using the ClearPath MCP Web Server.

Dr. Michael Salsburg and Larry Krablin

Though it is unlikely that the risk of intrusion into an IT infrastructure can ever be totally eliminated, there are many ways to implement security measures that can prevent cyber attacks. It begins with deploying a security-in-depth architecture where concentric rings of security are established to repel various types of attacks. One of the most common system vulnerabilities exploited by cyber attackers today and chief causes of concern is an entry point caused by a buffer overflow. This paper focuses on buffer overflows, the potential introduction and execution of arbitrary executable code and how ClearPath MCP based systems protect against this type of cyber attack.

- > Consulting.
- > Systems Integration.
- > Outsourcing.
- > Infrastructure.
- > Server Technology.

UNISYS

Imagine it • Done •

CONTENTS.

Executive Overview.	1
Introduction.	2
▶ ClearPath MCP repels Code Red.	2
The Current Hostile Environment.	3
▶ Public enemy #1: buffer overflows.	3
How MCP Repels Intrusions.	4
▶ “Tag” architecture prevents buffer overflows.	4
▶ Tags infuse defense throughout the MCP architecture.	5
Web Transaction Server Heritage.	6
▶ MCP provides new defense for Web servers.	6
ClearPath Web Transaction Server Futures.	7
▶ MCP provides new defense for application servers.	7
Conclusions.	9
Biographies.	10
▶ Authors.	10
Appendix A.	11
▶ The CERT Summary Report details.	11

EXECUTIVE OVERVIEW.

The risk of intrusion from cyberspace has escalated from aggravating attempts by hackers to politically motivated cyberterrorism. A recent “worm” infecting the Internet was intended to disrupt a single company due to its activities regarding copyright infringement. The risk of intrusion can never be totally eliminated. Similarly, no single defense will be sufficient to repel all intrusion. Instead, a security-in-depth architecture must be deployed, where concentric rings of security are established to repel various types of attacks.

The most common intrusions reported throughout 2003 were mounted by taking advantage of an application that did not properly restrict the length of input text. By causing a “buffer overflow,” the intruder was able to execute arbitrary instructions, thus carrying out the attack from inside the system itself.

The ClearPath MCP operating system (OS) provides a robust architecture designed to repel this type of attack. Simply stated, it does not allow buffer overflows or the use of data as code. This is unlike any other commercial OS in the sense that every data structure in memory is protected through a careful orchestration that includes the hardware, OS and compilers. This provides a unique distinction over other operating systems, such as LINUX, UNIX, Windows, HP-UX, IBM-AIX, IBM OS/390 and others.

Just as there are architected MCP boundaries between all memory structures, there is an additional mechanism that keeps data isolated and distinct from code. The unique qualities of the ClearPath MCP can be employed to significantly reduce the risk that companies now face from this type of attack.

This paper focuses on buffer overflows and executable code. Unisys will periodically release papers focused on ClearPath security.

INTRODUCTION.

In June of 2001, the Code Red worm was identified. This worm took advantage of the Internet Server Application Programming Interface (ISAPI) extensions installed with most versions of Microsoft Internet Information Server (IIS 4.0 and 5.0). The attacking worm scanned for servers that had a default language of English and were listening on TCP port 80 (the default port for web traffic). The attacker then sent a lengthy string that overflowed a buffer. Once this occurred, the worm could execute its own code in the overflow on the attacked host. In this particular case, threads were spawned where each thread then looked for other hosts listening on Port 80 and an attack was directed at those hosts. In this way, the attack grew exponentially among unprotected servers.

By the time Code Red was discovered and the appropriate vulnerabilities were addressed, the cost of damages was estimated to be over 2 billion dollars.

ClearPath MCP repels Code Red.

During this time, a similar attack was made on a ClearPath site running Web Transaction Server for ClearPath MCP. The buffer overflow was immediately trapped through the hardware architecture and the process aborted, blocking the virus from taking advantage of the host to spread the attack. The attempted memory intrusion was analyzed and the software was enhanced so that further attacks would be repelled without any disruption.

This experience points out a generic vulnerability in practically all servers commercially available today. Software cannot check everything. The difference between servers running various flavors of LINUX, UNIX, Windows OS/390, etc., and ClearPath MCP servers is the hardware architecture. The hardware architecture for systems other than ClearPath MCP servers does not always provide intrinsic safeguards against:

- ▶ Buffer overflows, corrupting memory and allowing arbitrary code to be written
- ▶ Introduction and then execution of arbitrary code

These vulnerabilities are at the heart of the most costly intrusions throughout the Internet. This paper discusses why a ClearPath MCP server is more resistant to this type of intrusion. It also discusses the Unisys direction for the future of ClearPath MCP within today's Web and Application Server environments.

THE CURRENT HOSTILE ENVIRONMENT.

The environment in which an enterprise touches the public Internet is indeed hostile. There are a great number of individuals worldwide who enjoy finding and taking advantage of vulnerabilities that allow unauthorized access to systems. In general, there are only a few categories of intrusion. The main one is intrusion into a server. This sort of intrusion can compromise the server and essentially stop all traffic to or from the server. The intrusion can also take advantage of the compromised server to spawn more copies and thus infest an enterprise and the Internet in general.

Another type of intrusion is the one in which an unwitting user executes code surreptitiously slipped into an email or Web page. This type of intrusion can again spawn more copies. It can also set up the workstation as a “zombie” that will come to life some time in the future to generate traffic in a concerted effort to alter the operation of one or more servers.

Various URLs can be found that provide significant detail regarding the techniques used to execute arbitrary code as a result of a buffer overflow: The Carnegie Mellon CERT coordination center is a center of Internet security expertise. Each quarter, a summary of the current issues is published. The reports for 2003 are listed below.

- ▶ November 24, 2003: <http://www.cert.org/summaries/CS-2003-04.html>
- ▶ September 8, 2003: <http://www.cert.org/summaries/CS-2003-03.html>
- ▶ June 3, 2003: <http://www.cert.org/summaries/CS-2003-02.html>
- ▶ March 21, 2003: <http://www.cert.org/summaries/CS-2003-01.html>

Appendix A at the end of the paper contains the major entries from these four reports. CERT is focused on all aspects of vulnerabilities, from workstations to servers and compromises within network devices. In some cases, these vulnerabilities were revealed by a specific attack. In other cases, the vulnerability was identified before any actual attack took place.

Public enemy #1: buffer overflows.

The table in the appendix indicates the type of vulnerability, along with the mechanism used to take advantage of the vulnerability. Of the 26 entries, 17 took advantage of either buffer or integer overflows. Nineteen of these entries indicate that the vulnerability exposes the site so that the attacker can execute arbitrary code.

In general, vulnerabilities on Web servers are exploited through carefully crafted overflows in which unauthorized areas of memory are seeded with code that is then executed. This was the mechanism used by “Code Red.” It is also precisely the type of intrusion that is defended against by the MCP architecture. The MCP architecture also protects the system from allowing data to be executed as though it were code. Code segments are unique and their “fingerprint” in memory is unalterable. Similarly, code files are uniquely identified in the file system. A hostile attack cannot install a code file, nor can the attacker execute in an area of memory that is not clearly “tagged” as an authorized code segment.

This is key to ClearPath Web Transaction Server’s ability to resist the “Code Red” attack.

HOW MCP REPELS INTRUSIONS.

Typically, an enterprise lowers the risk of intrusion through a set of different mechanisms. Think of server security in terms of a castle or fortress with a number of different defenses, forming concentric circles of defense against intrusion. The first defense is to achieve a high vantage point. The next defense may be a moat, the next impenetrable walls and so on. A similar approach is used by an enterprise to lower risk.

Here are some typical layers for preventing intrusion:

- ▶ Firewalls in front of servers
- ▶ Network intrusion detection
- ▶ Virus detection
- ▶ Authorization/policy enforcement
- ▶ Careful checking within applications for messages that are received
- ▶ Operating system security mechanisms
- ▶ Hardware architecture

Each layer repels a certain percentage of attacks, while the next layer focuses on what penetrates the previous layer. The ultimate goal of a hacker is to infiltrate all layers and execute code on the system. ClearPath MCP systems provide two security features at the innermost level that are not available in today's "open" operating systems. First, it is architected to disallow the execution of data as code. And second, it delineates memory area boundaries. In front of this wall of defense, it marshals an integrated hardware/software/compiler design to immediately detect buffer overflows and block attempts to trespass the memory bounds of a data structure.

"Tag" architecture prevents buffer overflows.

The ClearPath MCP system instruction set uses a "tag" on each word in memory to indicate the purpose (and thus constrain usage) of each word. All explicit data referencing is done through specially tagged words called descriptors. These are created by the hardware and the operating system using instruction sequences unavailable to ordinary user code. Every reference to memory through a descriptor is checked both for the validity of the descriptor and that the reference is within the bounds of the memory area described. Thus conditions such as buffer overflow are detected by the hardware and blocked before the overflow even occurs.



During execution, object code is placed in memory and tagged so that it is not accessible (read or write) as data to the program. That is, the program is not given any descriptor it can use to reference the area holding either its own code or that of another program. It also has no mechanism to force execution of data as code. These gambits require that a rogue program generate code to be able to forge descriptors, but the descriptors available to the program code do not allow code generation. So, the rogue is blocked, caught and reported.

Additionally, because code and data are kept in different “containers,” there is no notion of “adjacency” that could be exploited to overwrite code, even if the hardware bounds checking allowed it. That is, there is no relationship at all between the memory addresses of a data buffer and an area containing executable code.

The machine instructions or instruction sequences that are able to construct and/or modify control words such as descriptors are designated “restricted” and are not generated for any user program. All compilers that Unisys certifies for use on ClearPath MCP servers guarantee this. Any compiler not supplied by Unisys must be certified by the system administration (human), who then takes responsibility for its behavior.

Tags infuse defense throughout the MCP architecture.

The concept of tagging is even extended to the MCP file system. MCP files are each tagged with a “FileKind” that is only accessible through the MCP. Through tags and other file attributes, files are classified and marked regarding type and allowable use, as well as regular security attributes.

Program code files that invoke restricted instructions or sequences (unsafe programs) are identified by the compiler as such, with enough information for the MCP to determine the circumstances in which they may be executed. Unsafe programs may be designated by system administration as system libraries. Here again, the system administration must take responsibility for certifying the behavior of the library.

Compilers guarantee that code files behave properly or mark them as “executable only by administrative intervention.” The MCP guarantees that only properly certified files may be executed, that only properly authorized compilers can create code files and that “unsafe” code files are only executable with additional administrative authorization. The instruction set guarantees that programs using it properly, as certified by the compiler, cannot damage or take over the system. The MCP itself is loaded “administratively.”

All other operating systems except MCP work at the level of virtual memory pages to manage memory and enforce the security architecture. On the ClearPath MCP, each data structure is treated as a virtual segment that is managed and protected independently. This approach to computing provides the necessary protection from buffer overflows and the execution of rogue code.

WEB TRANSACTION SERVER HERITAGE.

The Web Transaction Server for ClearPath MCP is a native MCP Web Server written in NEWP (a dialect of Algol), and it includes a modified version of “Tomcat” to support Servlets and Java Server Pages. This HTTP server, along with Tomcat, provides a full implementation of the J2EE Web container, which includes support for both static and dynamic web pages. Tomcat is an Open Source reference model for the Web Container. It is currently maintained as part of the Jakarta project, which in turn is part of the Apache Software Foundation. See: <http://apache.org>

Many are familiar with the common term, Apache Web Server. This is actually the HTTP server that is supported by the Apache Software Foundation. Within the Apache Foundation, the Jakarta project has been established. The Jakarta Project creates and maintains Open Source solutions on the Java platform for distribution to the public at no charge for the programs themselves. One of the products supported is called Tomcat. Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process. You can find them on the Web at:

<http://java.sun.com/products/servlets>

<http://java.sun.com/products/jsp>

Tomcat is extremely popular and has been deployed at many sites that need more than simple http processing, but do not need the full Java2 Enterprise Edition (J2EE) functionality. The combination of Tomcat and the Web Transaction Server for ClearPath MCP provides an efficient, secure Web application deployment platform.

MCP provides new defense for Web servers.

The Web Transaction Server for ClearPath MCP enjoys the rich security architecture of MCP, along with the high availability of ClearPath systems. It is deployed at a number of ClearPath sites that require exceptionally high security and availability.

Note that the coordinated hardware/software enforcement of array bounds is an integral part of Algol, COBOL and Java, but the definition of the C language disallows it. So hardware-enforced bounds checking for C programs is only to the level of the entire data pool for a C program. However, code is still maintained and tagged separately and data cannot be substituted for code.

CLEARPATH WEB TRANSACTION SERVER FUTURES.

For both ClearPath product lines, Unisys has committed to providing full implementations of the J2EE environment. These will be based on the highly popular Open Source implementations of Tomcat and the JBoss Application Server.

A disruptive technology driven by the Open Source movement is taking hold in the server software infrastructure area. Tomcat and JBoss are two components of this technology. JBoss is an Open Source application server and Tomcat serves as a Web container for JBoss.

MCP provides new defense for application servers.

The following graphics show how the HTTP Server, Tomcat and JBoss Application Server fit into the J2EE environment on open, ClearPath and other servers. Figure 1 shows the basic components of the J2EE environment. Functionally, it is divided into the Web container and the Enterprise Java Bean container. The Apache HTTP Server fits in as a process that responds to requests for html (both .htm and .html URLs) from the client machine. Alone, the Apache HTTP Server cannot provide dynamic Web pages. However, in the J2EE environment, both servlets and Java Server Pages (.jsp) provide dynamic content for the client machine through Tomcat. Figure 2 illustrates this.

The JBoss Application Server can contain the entire J2EE environment (see Figure 1), but Tomcat is often preferred for the Web container. JBoss Application Server releases up through version 3.0.x included JBossWeb to handle the Web container, but its newest release, 3.2.x, includes Tomcat specifically to handle the Web container functionality. Figure 3 illustrates this environment.

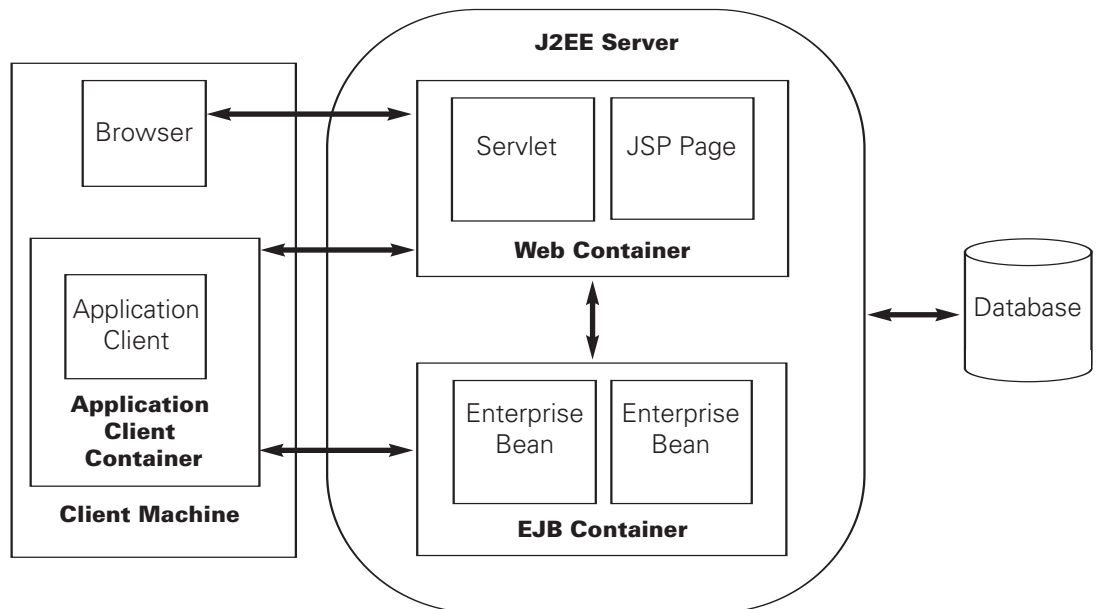


Figure 1: The J2EE Server

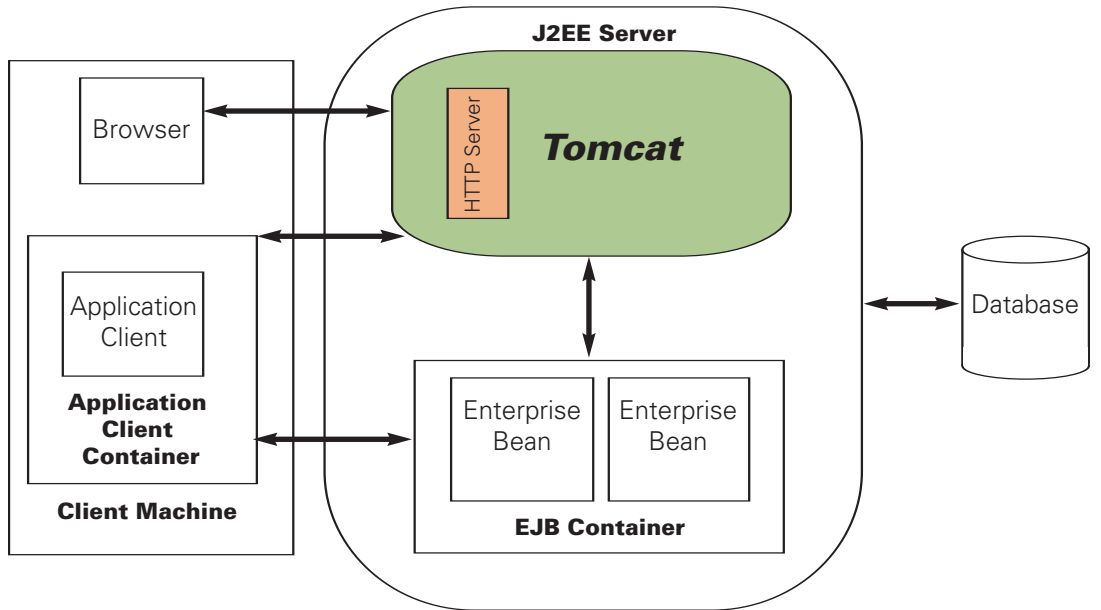


Figure 2: The Tomcat Environment

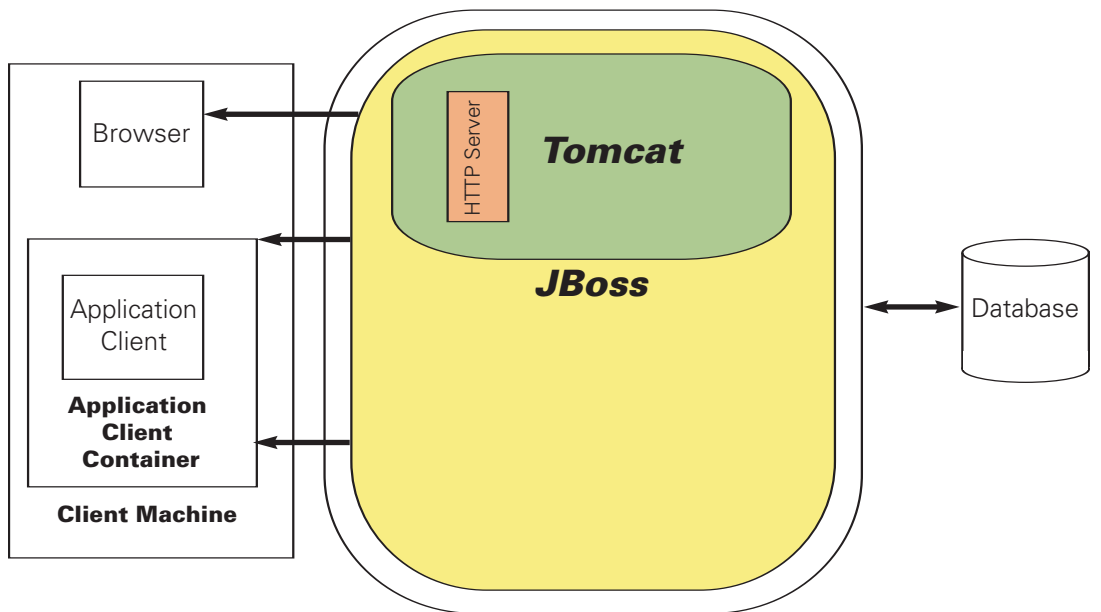


Figure 3: The JBoss Environment

JBoss Application Server is now available on both ClearPath MCP and ClearPath OS 2200 servers. The future details of the Unisys Web Transaction Server are still under consideration, however, Unisys is dedicated to providing a mainframe-quality environment for the JBoss Application Server. Tomcat, along with an appropriate HTTP Server such as the Web Transaction Server for ClearPath MCP, will be probably the container for this environment.

Along with a robust Web server, Unisys will infuse the J2EE environment with its traditional mainframe qualities, including improvements in security, availability, recovery and data integrity to the levels mainframe customers have come to expect.

CONCLUSIONS.

Today's Web servers are the ultimate line of defense in a very hostile environment. There are malicious people working diligently on new ways to take advantage of software vulnerabilities to either create meaningful havoc or just to achieve some personal gratification. The cost of these attacks on the computing community is escalating. One can only expect that more virulent attacks will appear in the future.

Most of the attacks on Web servers rely on overflowing buffers and then executing rogue code. This is an effective strategy on most servers because the underlying hardware architecture does not provide the necessary line of defense to repel the attack that slips through holes in software layers. As a practical matter, software alone cannot guarantee protection of all potential points of vulnerability. It must have hardware help. For ClearPath systems, any attempt to intrude on non-authorized memory (buffer overflow) or any attempt to execute an area of memory that is not designated as code will be immediately detected and blocked.

For the CIO, security management is risk management. When the risk was low, the amount of attention could be low. But, in today's more hostile world, where ever more malicious viruses emerge almost daily, the entire enterprise can be at risk. ClearPath systems provide levels of intrusion protection beyond that provided in any other commercial operating systems today. Though all risk can never be eliminated, ClearPath MCP provides additional depth in resisting the current methods used for costly cyber attacks.

Authors.

Dr. Michael Salsburg.

Michael is currently a director within the Chief Technology Office for the Unisys Systems and Technology group. He explores and recommends engineering's strategic technical direction for their future products, with a focus on system performance and overall architecture. Michael received his Doctorate in Mathematics (Probability and Statistics) from Drexel University in 1992. He has been awarded two international patents in the area of algorithms and software. In addition, Michael has published dozens of papers and has lectured worldwide on the topic of computer performance evaluation. His current interests are focused on very large-scale storage solutions and emerging enterprise management technologies.

Larry Krablin.

Larry's current responsibilities include coordinating the evolution of the ClearPath MCP instruction set architecture, to which he has been a contributor for many years as an architect and consultant for MCP software development. Larry holds engineering degrees from Cornell University and the University of Pennsylvania and has more than 35 years of experience with MCP software and architecture, especially compilers. He holds patents in the area of binary translation and emulation.

APPENDIX A.

The CERT Summary Report details.

Description	Overflow	Execute Arbitrary Code	Other
CERT Summary CS-2003-01			
Buffer overflow vulnerability in core Windows DLL	Y	Y	
Remote buffer overflow in Sendmail	Y		
Increased activity targeting Windows shares			Take advantage of null or weak administrative passwords
Samba contains buffer overflow in SMB/CIFS packet fragment reassembly code	Y	Y	
MS-SQL Server worm (SQL Slammer)	Y	Y	
Multiple vulnerabilities in implementations of the Session Initiation Protocol (SIP)			SIP is a new protocol that, due to its lack of maturity, may cause unexpected behavior in a system
Multiple vulnerabilities in SSH implementations	Y	Y	
Buffer overflow in Microsoft Windows Shell	Y	Y	
Double-Free Bug in CVS Server		Y	
Buffer overflow in Windows Locator Service	Y	Y	
CERT Summary CS-2003-02			
Integer overflow in Sun RPC XDR library routines		Y	
Multiple vulnerabilities in Lotus Notes and Domino	Y	Y	
Buffer overflow in Sendmail	Y	Y	
Multiple vulnerabilities in Snort intrusion detection preprocessors	Y	Y	
CERT Summary CS-2003-03			
W32/Sobig.F Worm			Malicious attachment installs an executable and data file
Exploitation of vulnerabilities in Microsoft RPC Interface	Y	Y	
W32/Blaster Worm	Y	Y	
W32/Welchia	Y	Y	
Cisco IOS Interface blocked by Ipv4 Packet			A device receiving specifically crafted IPv4 packets will force the inbound interface to stop processing traffic
Vulnerabilities in Microsoft Windows libraries and Internet Explorer	Y	Y	

Description	Overflow	Execute Arbitrary Code	Other
CERT Summary CS-2003-03			
W32/Mimail variants			Executable attached to an email
Buffer overflow in Windows Workstation Service	Y	Y	
Multiple vulnerabilities in Microsoft Windows and Exchange	Y	Y	
Multiple vulnerabilities in SSL/TLS implementations	Y	Y	
Exploitation of Internet Explorer vulnerability		Y	Convinces user to read an html document that contains HTA (HTML Application) objects that that in turn enables execution of arbitrary code at the user's authorization level
W32/Swen.A Worm			User must execute an attachment

For more information about ClearPath Plus MCP environments, please contact your Unisys representative.

Or call Unisys today at:

1-800-874-8647, ext. 776 (U.S and Canada)

00-1-585-742-6780, ext. 776 (Other countries)

You can also find the most current information about ClearPath Plus MCP environments on our website at:

<http://www.unisys.com/cp/libra>

You can also contact us by email at:

<mailto:cic@unisys.com>

**For the most up-to-date information about
ClearPath Plus MCP environments, visit
our website at:
<http://www.unisys.com/cp/libra>**

**Or call Unisys today at:
1-800-874-8647, ext. 776 (U.S and Canada)
00-1-585-742-6780, ext. 776 (Other countries)**

Specifications are subject to change without notice.

© 2004 Unisys Corporation. All rights reserved.

Unisys and ClearPath are registered trademarks of Unisys Corporation. All other brands and products referenced herein are acknowledged to be trademarks or registered trademarks of their respective holders.

11/04



4126 5323-000