WHITE PAPER

McAfee
Together is power.

Unisys ClearPath MCP

Independent Security Assessment

Table of Contents

Unisys ClearPath MCP

Independent Security Assessment

Unisys coordinated with Foundstone Services—part of the McAfee® Professional Services offering—on a multiphase independent product security assessment of the Unisys ClearPath MCP operating system. The scope of this project focused on the ClearPath MCP core security. The engagement consisted of operating system evaluation, default security configuration assessment, and a threat modeling exercise. Foundstone consultants conducted reviews and attempted exploitation during each phase of the assessment. The original testing began in June 2015 with a retest conducted in October 2015. Foundstone® consultants provided daily feedback to the Unisys team to discuss findings and work with staff on remediation.

This document provides a detailed security review of the MCP operating system. The Foundstone group recognizes that the MCP product line was designed for a high level of security, and the core operating system was designed following best practices for authorization, resource management, and reliability.

MCP Overview

The Unisys MCP operating system is a general-purpose operating system designed to provide a robust and secure platform for a wide variety of customer-specific requirements. Consequently, security requirements for an MCP system deployment are highly specific to each customer application and security requirements.

The MCP core operating system provides, manages, and monitors resources including memory, processor, and file access. These resources are allocated to user and system processes based on permission and priority to facilitate a multifunctional operating environment. Monitored resource usage is stored in a central log for system analysis by Administrators, and MCP supports multiple user roles and permissions. User accounts may be assigned levels of access based on requirements to files read/write/execute on the host.

In addition to the security built into the MCP platform, Unisys provides and maintains extensive documentation on security features, and PCI best practice guide for secure deployment of the solution and the various components therein.

Testing Methodology and Findings

Foundstone consultants utilized multiple testing methodologies that focused on a specific component of the architecture. All testing was conducted against MCP version 16.0, with an emphasis on manual testing using Foundstone proprietary testing methodologies and industry experience testing non-standard platforms.

Operating System Assessment

Multiple interviews were conducted with developers and engineers, as well as a review of the MCP default configuration, update methods, and a review of MCP core operating architecture and system services.

This top-down approach used in assessing MCP included a review of default settings, user interaction, and the running environment. Strong emphasis was placed on testing the allocation of operating system resources such as disk space, memory, and processor cycles. These resources were vetted for vulnerability to direct or indirect attacks which would allow an attacker to gain unauthorized access to the system, escalate privileges, or deny service. This involved an extensive review of system architecture as well as creating custom software for testing the host system. Application and configurations not integrated into to the core operating system were outside the scope of this assessment. These included, but were not limited to, the following: network services, compilers, file editors, databases, and third-party applications.

Threat Modeling

The threat-modeling process began with a high-level architecture discussion with the MCP solution architects in order to understand the overall platform's functionality, security requirements, and deployment environment. Foundstone consultants then worked with the development team to understand the implementation of the requirements within the live environment. From these initial meetings and a thorough review of the documentation provided by Unisys, Foundstone consultants derived a list of credible threats to the various components which make up the solution. This process exposed the key components that are likely to be the target of an attack, including storage locations for sensitive information, configuration files, data flows, and other significant features that may pose a threat to the security of the platform. The threat model also resulted in a list of potential threat vectors and existing countermeasures.

Areas of Analysis

Foundstone consultants focused on multiple areas for the purposes of analyzing the security of the overall solution. The table below breaks these focus areas down into high-level topics and provides guidance and recommendations based on Foundstone consultants' observations during this engagement.

Analysis Topic	Best Practice	Evaluation	Recommendation
Memory Management	Resources must be constantly monitored for ownership, authorization, and availability. Resources assigned to one process should not be available to another process unless explicitly required.	MCP implements a safe memory program execution environment, which prevents intentional or unintentional memory corruptions. All programs are written in high-level languages in which access to data segments are automatically bounds-checked through various restrictions as a feature. Programmers work with memory descriptors, or references, and never use direct memory pointers. Memory segments are tagged by the operating system to describe their intended purpose—for example, stack, data, or code segment. This prevents the execution of data or stack segments as code. All data received by the system from the network is automatically stored in memory and tagged as a data segment, meaning it cannot be executed. These controls mitigate common memory corruption attacks such as buffer overflow. Based on interviews with MCP developers, Foundstone consultants learned that before memory is allocated to a new process, the memory is scrubbed by setting all bits to zero. This prevents memory from a previous process from being accessed by a new process. However, this process was only performed upon allocation of memory, not upon de-allocation—which left the previous contents of the memory unaltered.	Memory management is an intrinsic responsibility of the operating system, and preservation of its integrity (in this context, scrubbing its contents) must be handled automatically, regardless of the state of running applications and their assigned memory. The memory management either meets or exceeds industry best practice.
Logging and Auditing	Extensive monitoring of the host operating system and applications allows administrators insight into operations which may not be visible to the user. The operating system should be capable of presenting administrators with a log of system activities. Administrators should review logs at regular intervals for security violations and anonymous activity.	MCP provides an auditing and logging component an inherent part of the operating system. There are two primary logs managed by MCP. The SYSTEM/SUMLOG file contains log records for a variety of system-related events as well as security events such as login success/failure events, access control violations, network access violations, attempted resource misuse, and other security related events. The SYSTEM/SECURITYLOG logs security-sensitive information from SSL, IPSec, and SSH that are relevant for diagnostics. A log analyzer tool provides system security administrators the ability to view and filter the SUMLOG to readily identify security-related events for detection and forensic purposes.	No recommendations are necessary for this topic. The system either meets or exceeds industry best practice.

Analysis Topic	Best Practice	Evaluation	Recommendation
Resource Availability	The core functionality of any operating system is the management of resources. In order for an operating system to function reliably and securely, it must be capable of allocating the appropriate resources for each process. The operating system should attempt to prevent any open process from adversely affecting other processes from resource starvation.	Processes running on MCP use a shared pool of resources. Each MCP installation provides a finite amount of resources which are shared among all running processes. Additional resources can be manually assigned by an Administrator. Foundstone consultants created a custom application to attempt allocation of all available processor cycles, memory, and disk space on an MCP system. An unprivileged user account was used to execute the resource allocation process. Upon execution, other processes were blocked from allocating processing time, disk I/O, and memory. Consultants discovered that as other processes de-allocated resources, the custom application would expand and allocate those resources. Processes (including the core MCP) were able to maintain resources allocated prior to the execution of the malicious process. This allowed most core MCP processes to remain running, despite all visible resources being consumed. The MCP core operating system maintained uptime during resource starvation. However, processes requiring additional or initial resources were unable to continue to function properly. MCP supports manual restrictions on resource allocation, and prioritization of processes. MCP allows processes with a higher priority to receive more frequent processing cycles. Disk allocation can be restricted for user accounts in specific contexts. This method would still allow a process to consume all available disk space accessible to the user.	Foundstone consultants recommend implementing functionality that limits the resource allocation for a single process. This must restrict the default processing, memory, and disk write capabilities of each process.

Analysis Topic	Best Practice	Evaluation	Recommendation
Default Security Configuration	Apply stringent resource restrictions based on user access requirements. Vendors should encourage secure installations of products. Optional security settings should be configured in the default secure state.	MCP supports a wide variety of security configuration options. These options allow administrators to configure a system to comply with standard security best practices, and context specific security policies in an organization. MCP provides several built-in security profiles (S0-S2). Each profile creates a more stringent environment than the previous profile. The modification of security configuration has the potential to cause interruption for running MCP environments. It is the practice of Unisys not to force changes on the operating system that could cause disruption or configuration conflict. A best practice security implementation may not be enabled by default, in the interest of system usability. A robust security support library allows Unisys customers to customize authentication controls to meet their specific requirements. The MCP installation was configured with three default user accounts. The user accounts were publicly known usernames and passwords.	Force the modification of all default user account passwords during the installation and initial configuration period. Users must be required to configure new and complex passwords for each user account. The user must be prevented from reusing the default password for that account. Foundstone recommends the development of a Security Technical Implementation Guide (STIG) for the MCP operating system. MCP servers subject to PCI regulations can be configured following the Unisys "MCP Security Payment Card Industry (PCI) Data Security Standard Guidelines." MCP servers subject to additional standards can reference the "MCP Security Overview" for configuration options.
Threat Modeling	Threat modeling is a structured approach for identifying, evaluating, and mitigating risks to system security. The view for threat modeling varies greatly depending on the environment. However, general best practices can be applied to most technologies. These include resource allocation, user authentication, authorization based on roles, secure coding practices, and update management.	Foundstone interacted with key personnel at Unisys, including members of the development team and security department. The team conducted a two-day workshop to understand the current architecture and design, and then considered all the threats, countermeasures, and potential vulnerabilities that might exist in the system.	No recommendations are necessary for this topic. The system either meets or exceeds industry best practice.

Analysis Topic	Best Practice	Evaluation	Recommendation
Account Security	User accounts should be assigned permission based on the principle of “least privilege.” This principal dictates that permission be provided for accessing resources only when required. Resources should be restricted to a single owner by default and require manual modification for additional user access privileges. Administrator accounts should be provided only to users with the authority to perform system-wide modifications. Mechanisms should be enforced that prevent user accounts from being used by an unauthorized party. The account username and password combination should be complex enough to thwart attempts at recovery.	All access points into the MCP system are designed to require authentication; there are no entry points where a user can gain unauthenticated access. The MCP operating system recognizes three account types: User, Administrator, and SecAdmin. User accounts may access Public files and those in their account directory. Administrators may access Private and user files. Administrators also have permissions to modify most system settings. SecAdmin has full access to system resources. In addition to the permissions of the Administrator, the SecAdmin can modify security-specific resources and settings. These account types allow administrators to assign accounts with appropriate permissions. These accounts were granted appropriate permissions by default. Additional permissions could be permitted as required. Foundstone consultants observed that the MCP system did not record usernames as case sensitive. All lower case alphabetical characters were converted to upper case prior to being interpreted and stored by MCP. This method of username string storage greatly decreases to key space for user accounts, and might assist an attacker when attempting to brute-force potential usernames to gain access to the system. Default password policies on MCP did not meet recommended standards, and most password policies were either set to the minimum or disabled entirely. The following policies were observed: ▪ Enforce Password History = 0 ▪ Minimum Password Age = 0 ▪ Minimum Password Length = 0 ▪ Password Must Meet Complexity Requirements = Disabled ▪ Account Lockout Duration = 0 minutes ▪ Account Lockout Threshold = 0 ▪ Password Case Sensitive = Disabled	Unisys provides an optional security library to its clients, which can be configured to enforce best practices. The existing FAQ article 4257 provides details on configuration option for the security library. The security library should be installed by default to provide sufficient security in its default configuration. Foundstone consultants recommend requiring that the default configuration for new MCP deployments enforce case sensitivity for system usernames, and that a process/update is implemented to enforce the same requirement on existing MCP deployments. ▪ Enforce Password History = 24 ▪ Minimum Password Age >= 1 ▪ Minimum Password Length >=12 for regular users, >=14 for administrators ▪ Password Must Meet Complexity Requirements = Enabled ▪ Account Lockout Duration >= 30 minutes ▪ Account Lockout Threshold <=3 ▪ Password Case Sensitive = Enabled In the default implementation, AdMiniStRatOr would convert to ADMINISTRATOR when assigned as the username. User names can be case sensitive if placed in double quotes. For example, “AdMiniStRatOr” would maintain the upper and lower case characters. Unisys should store all usernames in quotation to allow case-sensitive usernames. Foundstone recommends that additional measures be taken to ensure that user credentials are not compromised due to weak password storage mechanisms. Eliminate the use of LanMan (LM) hashes that support backward compatibility with LanMan authentication protocol, as well as the legacy/proprietary custom digest function. Use stronger hashing techniques that are consistent with current security best practices and resistant to brute-force attacks that are possible with modern computing resources. Adding a cryptographically random 8-byte salt value to the SHA-256 hash would be beneficial to help ensure resistance to brute force attacks. Examples of algorithms that provide functionality such as recommended above include PBKDF2, bcrypt, and scrypt.

Analysis Topic	Best Practice	Evaluation	Recommendation
		MCP was designed to provide a variety of mechanisms for authenticating end users, including simple "usercode" (username) and password, Kerberos, NTLM/NTLMv2, smart cards, and more. A malicious privileged user or attacker who gains access to the MCP USERDATAFILE may be able compromise user passwords using brute-force techniques against weak password hashes. During discussions with the Unisys team, it was noted that user passwords are stored in the MCP USERDATAFILE using weak hashes: LanMan (LM) hash, SHA-256 using a salt that is the user's usercode (login id), and a 48-bit legacy/proprietar y digest function developed by Unisys.	
File Access	Files must be evaluated for ownership and permissions prior to access. Files assigned ownership to one process or user should not be accessible to another process or user unless explicitly configured. Access to resources should be permitted based on read, write, and/or execute.	Access restrictions are enforced on all files within the MCP environment. Access permissions on files can be configured to Private and Public as well as read and write. These settings are stored in the header for each file. A Public file can be accessed by all users, while a Private file is restricted to Administrators and system processes. MCP is designed with access control capabilities that extend the POSIX ACL model for discretionary access control with a more finely-grained access control mechanism known as GUARDFILEs. This provides a flexible means for users and system administrators to control which users and programs can access their data files, programs, and databases. Access is based on a variety of criteria, allowing for the creation of an access control model to meet a wide variety of customer requirements. Processes are not permitted to view files without permission. Additional permissions can be assigned custom for specific users, groups, and processes.	Evaluate open access to resources prior to confirming changes to permission in order to avoid unauthorized access after permission changes take effect.

Analysis Topic	Best Practice	Evaluation	Recommendation
		File permissions are verified at resource access time. The modified permissions do not affect access in conditions when resources are currently loaded by a process. Foundstone consultants leveraged a standard user account to execute a process that opened a Public file for reading and writing. While the user's process maintained access to the file, an Administrator account changed the privilege to take ownership of the file and set it to Private. This permission setting allowed only the Administrator to access the files. After the file had been changed to Private, the running user process was able to write to the Private file. After process termination, additional attempts to access the Private file were denied to the user.	
Development Environment Security	Access should be restricted to system development software to authenticated user accounts. Development tools that can bypass system access restrictions should be explicitly allowed only when required. Administrators should be made aware of the potential risks.	MCP provides a robust development environment to allow consumers to develop custom applications. Software supported by MCP is predominantly written in NEWP (a derivative of ALGOL designed for operating system and low-level program development). Foundstone consultants observed that the NEWP compiler was provided by default on MCP installations. The ability for Administrators to execute "unsafe intrinsics"—built-in operating system functions for low-level access (for example, raw memory access)—was also enabled by default. Foundstone consultants observed that the debug analysis tool (included in the default installation) allowed Administrators unlimited visibility into the running system via the live-analysis mode—including the MCP internals and raw memory. Consultants observed it demonstrated by an MCP systems developer as it was used in a typical development environment.	Foundstone recommends restricting access to the NEWP compiler and/or its ability to use unsafe intrinsics (for example, functions that allow raw memory access and low-level system manipulation) for default installations. Consider leveraging existing security implementations to restrict access, as they are already mature production code and included in the design. For example, the file header for the NEWP compiler and any code it generates could contain a flag only accessible and visible to the MCP core system to determine if access is allowed. Disable or remove tools that are not required by the MCP server for general operation. Limit visibility to context only required by end-users for debugging purposes rather than allowing unrestricted visibility into operating system internals.

Analysis Topic	Best Practice	Evaluation	Recommendation
Update Management	All software update packages should be provided in a trusted manner. Packages should be transported encrypted over the network from the update server to host(s). All software updates should be verifiable for authenticity and integrity by the host.	MCP files produced by Unisys are the primary software used by the MCP operating system host. Foundstone consultants were provided with two update package container files. The updates were contained in a CON file. The container files may include sourcecode, configuration files, sourcefiles, sample files, libraries, etc. The update files were examined for a better understanding of the software as part of how the system operates. Files are not encrypted and could be read on disk. Files not sent securely could be captured in transit. Foundstone consultants were able to extract and view contents of update packages, as no encryption was implemented. Modifications to the container file wrapper were successfully accepted by MCP using the standard installation method. Modifications to the headers and the files were identified by the MCP using checksums contained in the wrapper.	While it is not currently a common practice for MCP administrators to install third-party software, Foundstone consultants recommend that the MCP update service perform a check on all containers before installation of any new updates. All update files must be signed by a recognized trusted source and verifiable by the MCP. The updates must then be verified by the MCP to prevent installation in the event of file tampering. Encrypt the package files to mitigate the risk of their contents being read while residing on disk or in transit. Implement a method whereby decryption of packages residing on disk can only be performed by authorized developers and the software that the package updates.

Conclusion

Unisys ClearPath MCP provides the capability for consumers to configure a highly secure operating system environment. Security design principles such as "deny by default" and "least privilege" have been incorporated into the foundation of MCP from its origins. Administrators can leverage the extensive MCP documentation on configuration options to customize an environment which meets a variety of security best-practice standards.

MCP is also designed to provide a secure environment for program execution, which protects against attempts to inject and execute malicious code. MCP access control capabilities extend the POSIX ACL model for discretionary access control to allow users and administrators a more granular means to control which users and processes can access data files, programs, and databases. ClearPath MCP provides a robust auditing and logging mechanism integrated into the operating system. Extensive logging includes a large number of events that are logged as being either security relevant, or security violations, enabling the rapid discovery and forensics of potential attacks. MCP provides a truly integrated technology stack in which all system components, including resource allocation, system monitoring, and account management have all been designed, implemented, and tested to work in unison securely.

As with any software platform, administrators and users can greatly affect the overall security. The MCP environment offers a wide variety of security configuration capabilities. MCP configuration settings in the "MCP Security Overview" (3834 7639–006) published by Unisys allow systems to operate in a secure manner within the context of the overall platform. An additional security guide "MCP Security Payment Card Industry (PCI) Data Security Standard Guidelines" (3850 7315–002) is a reference for a secure configuration for the server based on PCI compliance. Furthermore, Unisys has demonstrated a proactive stance on security by conducting an independent assessment of the MCP technology.

About Foundstone Services

Foundstone Services offers expert services and education to help organizations continuously and measurably protect their most important assets from the most critical threats. Through a strategic approach to security, Foundstone consultants identify and implement the right balance of technology, people, and process to manage digital risk and leverage security investments more effectively. The company's professional services team consists of recognized security experts and authors with broad security experience with multinational corporations, the public sector, and the US military.

Reference Material

- "MCP Security Overview" (3834 7639–006)
- "MCP Security Payment Card Industry (PCI) Data Security Standard Guidelines" (3850 7315–002)
- "Security Administration Guide" (8600 0973–421)
- "Security Operations Guide" (8600 0528–207)
- "System Configuration Guide" (8600 0445–309)

About McAfee

McAfee is one of the world's leading independent cybersecurity companies. Inspired by the power of working together, McAfee creates business and consumer solutions that make the world a safer place. By building solutions that work with other companies' products, McAfee helps businesses orchestrate cyber environments that are truly integrated, where protection, detection and correction of threats happen simultaneously and collaboratively. By protecting consumers across all their devices, McAfee secures their digital lifestyle at home and away. By working with other security players, McAfee is leading the effort to unite against cybercriminals for the benefit of all.

www.mcafee.com.

McAfee™
Together is power.

2821 Mission College Boulevard
Santa Clara, CA 95054
888 847 8766
www.mcafee.com

62176wp_unisys-mcp_0116
JANUARY 2016