Global Information Assurance Certification Paper

AUDITING CHECK POINT SECUREPLATFORM NG WITH APPLICATION INTELLIGENCE (R55)

WEB USER INTERFACE

AN AUDITOR'S PERSPECTIVE

JEFFREY SHAW, P.ENG, CISSP

GSNA PRACTICAL ASSIGNMENT

VERSION 2.1, OPTION 1

TABLE OF CONTENTS

1 ABSTRACT

This document outlines one approach to auditing a proprietary web configuration interface for a security device such as a firewall. A web user interface for firewall configuration should be an inherently secure interface. Security is especially important in the development of a web interface for configuring a security device because the running configuration and operational parameters of the system can be modified and saved from the web user interface. Were this interface to be compromised, it could lead to the compromise of the underlying operating system. If this happens at the gateway of the corporate network, the risk of further penetration into the network is greatly increased.

The primary goal in developing this audit process and checklist is to effectively analyze the security, vulnerabilities and risk associated with the firewall web configuration tool so that owner's of like systems have a better understanding of the risk in using it for remote configuration. This audit guide and checklist should provide a firm foundation for developing further assessment procedures for this application and for other similar products. Check Point's Secure Platform NG with Application Intelligence (R55) web configuration interface was selected for this project to apply the audit process to. It should be noted that Check Point states clearly in the User Manual for SecurePlatform that "the web user interface is not accessible in FIPS 140-2 compliant mode" –1 (Check Point Software Technologies SecurePlatform User Manual, p.19).

Normally, in a secure configuration, the web configuration interface for SecurePlatform would have strict access controls placed on it. This audit exercise simulates an environment where the security of the firewall access controls has been circumvented or has a logical flaw. The audit identifies the system to be examined, attempts to identify the risk associated with web user interface if it was exposed to attack, outlines an audit process with checklist and details the results of the execution of the audit in a clear and concise report.

2 RESEARCH IN AUDIT, MEASUREMENT PRACTICE AND CONTROL

2.1 SYSTEM DESCRIPTION AND AUDIT SCOPE

The system evaluated for this practical assignment is the web user interface for Check Point's SecurePlatform NG with Application Intelligence (R55). Secure Platform was installed in a default configuration in a VMWare session on the auditor's Toshiba A20 laptop.

In any organization, the role of the firewall is critical in providing access-control between networks with varying levels of trust relative to the organization hosting the firewall. Firewalls may provide perimeter security to protect the organization from threats associated with the Internet. Firewalls may also be configured to control network traffic between internal segments or partner connections. The security of these devices themselves is often overlooked in favour of examining the security of related systems and networks. In particular, in the appliance market, an assumption is often made by organizations that the appliance itself is always hardened and secure. This type of assumption will probably also be applied in many circumstances to a product such as SecurePlatform which is marketed as an instant "appliance maker" based upon the virtue that it has been developed and hardened by the security product vendor.

Check Point SecurePlatform is a purpose built, pre-hardened platform for Check Point firewall software. Check Point SecurePlatform is developed and maintained by Check Point Software Technologies Ltd. According to Check Point 9 of the top 10 Fortune 500 companies and 80% of the Fortune 500 use Check Point security products. Check Point provides other market share and interesting facts online at http://www.checkpoint.com/corporate/facts.html. Check Point software is clearly widely deployed and relied upon to provide security both on the Internet and within the corporation. Therefore, Check Point Secure Platform NG can be expected to have a relatively large install base within the worldwide firewall market.

The purpose of the SecurePlatform product is to provide a secure host operating system for the Check Point security product. The purpose of the web user interface is to provide a graphical interface to change key configuration parameters of the underlying operating system and installed security products. Configuration settings that may be accessed using the SecurePlatform web user interface include:

- Device status
- User administration
- Secure internal communications (with other Check Point products including the security policy server or SmartCenter).
- Network configuration including interfaces, PPPoE, VLANs, and PPTP. Routing and DNS are also configurable through the web user interface
- Products (add/remove, apply licensing)

- Device control- Start and stop any installed Check Point products.

As can be seen from the above list, many important attributes of a secure firewall platform can be modified through this web user interface. Having the capability to shut down the firewall or disrupt policy control are serious issues should the system be compromised.

A primary reason for choosing to audit this web interface is because a real need exists to confirm that the interface is relatively secure. Any device that utilizes a web interface for configuration may be open to attacks perpetrated upon that interface, if permitted by the firewall's access controls. In choosing to assess the security of the web user interface one is making the assumption that the access controls implemented on the firewall may be faulty in that they permit exposure of the web server and application to an attacker, either external or internal. In other words, the firewall security policy alone cannot be relied upon to protect the web user interface from attack.

This audit paper describes an audit process and checklist for the Check Point SecurePlatform web user interface only. The security of the underlying SecurePlatform operating system is not examined in this practical assignment. Any other servers running on the SecurePlatform installation (e.g. SSH, etc) were also excluded from the audit. The security of the Check Point firewall software itself was also not examined and not enabled for the duration of the testing.

2.2 RISK ANALYSIS

Threat agents for this risk analysis include malicious attackers, both internal and external, curious internal employees, and script kiddies to name a few. By first making the assumption that the web user interface is exposed to an external or internal threat agent, the risk analysis for the system is relatively straightforward.

For the sake of brevity in this report, low and medium risk threats and vulnerabilities for the most part aren't identified here. Most of these lower risk threats and vulnerabilities can be mitigated through proper implementation of policies and standards. Examples of these issues include verification of file integrity before upgrades, proper change management processes, and separation of duties.

When considering a web configuration interface such as that provided for SecurePlatform, some threats that could be combined with vulnerabilities (if they exist) and lead to compromise are detailed in the table below.

Threat	Vulnerability	Risk	Consequences
Un-patched system is exposed to threat agents and events (port scanning, VA tools, etc).	Assuming the application is actually the latest revision available then this threat is applicable to systemic vulnerabilities typically embedded in the web server software being used to host the target application	High, especially in off-the-shelf products that embed a particular version of server software as the host server (Apache is commonly used for this) and where the vendor doesn't follow the normal patch distribution for that web server.	Denial of service, data tampering, misuse of resources, privilege escalation, breach of network and ultimately unauthorized disclosure of sensitive information.
Weakly encrypted traffic or unencrypted traffic intercepted by threat agent	Not all systems strictly enforce a strong cipher. Weak ciphers are more vulnerable to attack.	Medium, the problem is particularly prevalent in the case where a vendor creates a generic product for worldwide export. Export restrictions may require the vendor to provide weak ciphers for certain countries.	Weak ciphers may be open to successful brute force attack in a reasonable timeframe, allowing the attacker to recover passwords for example.
Weak identification and authentication process exposed to threat agents and events (brute force, cached password, etc).	Vulnerabilities related to authentication processes are typically introduced when proper limitations are not imposed on the number attempts to authenticate	High, this needs to be evaluated for all systems. The security of the authentication process for device administration in particular in often overlooked. How many web administration interfaces require a client-side certificate for authentication?	Successfully exploiting vulnerabilities in the authentication process is akin to obtaining the keys to that system. Credentials may even be common across multiple systems resulting in an aggravated breach scenario where much of the network is at risk.
Inadequate auditing functionality	This vulnerability is introduced when logging of administrator access and actions is insufficient to track those actions.	Medium, audit logs need to be available when required in order that the organization understands the administration processes that are going on with respect to the firewall system.	Without proper audit logs, change management cannot be enforced, rollback to a know state may not be possible, and actions taken by administrators not understood or even known.

Application discloses excessive information either in normal operation or through error condition.	Error message with too much detail, detailed comments in source, hidden values, etc.	High, many vulnerabilities have their basis in providing the attacker with too much information.	Error messages may provide clues to directory structure, server version, etc. Hidden comments may provide insight into application architecture or the use of known scripts or CGIs. This type of information allows an attacker to better set up for further attacks.
Unexpected input	There is an opportunity to manipulate the data being sent to the application to elevate privilege or access the underlying OS, etc. Examples of this type of vulnerability include SQL and OS command injection, directory traversal and cross-site scripting.	High, these types of vulnerabilities are the most common across web applications. The vulnerabilities may be either inherent in the application code or in the web server itself.	These types of exposures may lead to direct access to the operating system or a sensitive database on a supporting internal system and often allow the attacker to execute arbitrary code, tamper with or retrieve sensitive data.
Logical manipulation	Vulnerabilities included in this category are information processing flaws related to improper session handling, page sequence flaws, etc. These are typically described generally as software errors and omissions introduced by the developer.	High. Any vulnerability that is associated with a flaw in the logical processing of input by the system is considered a serious problem. These risks are usually the result of a flaw in the application code not the web server.	Consequences include session related issues (hi-jacking, etc), impersonation, value manipulation (e.g. account, amount), etc. For example, a read-only account escalated to an administrative account could be and end result.

Table 1- High Level Threat and Risk Analysis

Now that threats and risks and potential consequences particular to a web configuration interface have been identified, security objectives and controls can be formulated. Important security objectives (this isn't an exhaustive list) and associated controls for a web configuration interface that manipulates critical settings and functionality of a bastion host are as follows:

Objective	Controls
The web server software must be at the latest patched version that addresses all know security vulnerabilities and functional issues for that version. If the server is embedded as part of the vendor product and cannot be patched separately, this needs to be identified as a high-risk exception.	▪ Vulnerability assessment and patch management tools must be run against the server on all ports servicing web requests. ▪ The server version, if it can be determined should be manually cross-referenced in CVE databases and confirmation made that tools being used are in fact testing for vulnerabilities and exposures particular to that version. ▪ If the server type cannot be determined then the server must be assessed for common web server vulnerabilities and exposures.
The web server should incorporate a strong authentication method involving both tokens and passwords or biometrics. If it does not and relies only upon user name and password then it must enforce the usage of strong cryptic passwords and also must restrict the number of logon attempts so as not to be open to brute force attack.	▪ Policy to enforce use of strong authentication if the device supports it. ▪ If username and password only, then confirm that a strong, cryptic password policy is being enforced. ▪ Run brute force tools against interface to confirm that it is not subject to this type of attack. ▪ ▪ Ensure that adequate administrative action logging is being enforced by reviewing logs provided by administrator of the system.
The web sever must not allow weak ciphers	▪ Assess which SSL compliant ciphers are accepted. ▪ Capture data to ensure that it is encrypted
The application and hosting web server must not disclose excessive information.	▪ Provide stimulus to illicit responses related to error conditions. ▪ Retrieve commonly available information including page source and banner information.
The web interface must be able to handle unexpected input without failing to a condition where the system is vulnerable.	▪ Test for common vulnerabilities in web applications of this type including directory traversal, OS injection, Unicode vulnerability, etc.
The interface application must be robust and logical in the performance of the functions that it is required to fulfill.	▪ Attempt to manipulate the logic of the interface using ethical hacking techniques.

Table 2- High-level Overview of Security Objectives and Potential Controls

The above tables will be used to create a detailed audit checklist that reflects the analysis performed in this section. The checklist provided herein has been developed to determine whether the security objectives highlighted are met in the product using the basic controls identified. In essence what is to be determined is whether the target interface is well developed in that it is not subject to the most common types of vulnerabilities and exposures for web based applications.

2.3 STATE OF EXISTING PRACTICE

2.3.1 CHECK POINT SECUREPLATFORM NG WITH APPLICATION INTELLIGENCE (R55)

Research specifically focused on determining whether an existing audit process or checklists were already available for the SecurePlatform web user interface yielded virtually no results. The primary research tools employed for this project were the Google search engine (www.google.com) and search engines within various security industry sites that are normally considered as audit information resources. These included SANS(www.sans.org), CERT (www.cert.org), NIST Computer Security Resource Center (http://csrc.nist.gov/), the Center for Internet Security (http://www.cisecurity.org) and numerous others.

One article in Linux Magazine Issue 28 (www.linuxmagazine.com) by Jörg Fritsch that provided a good general review of SecurePlatform, its purpose and capabilities was discovered. In particular, this article described some the hardening procedures that Check Point applied to the SecurePlatform product as determined by examination of the operating system by the reviewer. This information does provide a sense of assurance for SecurePlatform administrators that basic operating system security has been well architected in SecurePlatform. However, specific information, test results, and opinion concerning the security of the web configuration interface are not detailed in this article.

There are no existing GIAC GSNA practical assignments on this particular topic. Research into common criteria certification on http://niap.nist.gov/cc-scheme/VPL-Vendor.html revealed that Check Point NG has achieved Common Criteria certification of EAL4 and equivalent European E3 certification. A review of the material available indicates that SecurePlatform was not included in the certification process and therefore no existing audit process material could be gleaned from this source. SecurePlatform is currently under evaluation for FIPS 140-2 Level 2 certification according to Check Point (ref: http://www.checkpoint.com/products/downloads/government_certification.pdf). As an interesting aside, according to the SecurePlatform User Manual, page 19, the web interface is not accessible in FIP 140-2 compliant mode and there is a command to enable or disable this mode which among other things, enables and disables the web interface by removing the web daemon application.

2.3.2 STATE OF PRACTICE- WEB APPLICATIONS

Since there appears to be no existing specific practice in auditing Check Point's SecurePlatform web user interface then the scope of research was widened to examine security of web-based configuration front-ends for firewalls and network devices. Using the same research techniques as detailed in the previous section, an attempt was made to discover audit papers and checklists related to web applications. This yielded many more results than could be detailed in this paper. Some excellent sources for information on specific audit and assessment techniques for web applications included:

- http://www.cgisecurity.com/lib/ - Many papers on various topics including best practices for securing web applications, techniques for assessing security, and general knowledge papers.

- http://www.sans.org/rr/ - Again, many papers here on all information security topics from audit techniques to web server security.

- http://csrc.nist.gov/pcig/cig.html - NIST checklists and implementation guides. Of particular value was the STIG (Security Technical Implementation Guide) and associated checklists for web servers and applications.

- http://csrc.nist.gov/publications/nistpubs/ - NIST 800 Series Publications. Good general guidelines on many security topics including web server security (800-44).

- http://www.cert.org/security-improvement/#practices -This site outlines several excellent practices for securing web servers and applications. Information here detailing protection measures can definitely be applied to developing audit checklists to ensure that those protection measures have been implemented.

In addition to these resources, there are many excellent papers available that are provided by popular web server providers such as Microsoft and Apache. While much of the vendor's material is specific to their own product, there is also good general practice information available as well. Information from many of these sources was used to formulate checklist items and a specific reference is included in each checklist item. Enough information was available through research as detailed above to provide a firm foundation and general guidance for developing this paper.

3 CREATE AN AUDIT CHECKLIST

This section details a checklist developed based on the objectives and risk analysis presented in sections.

3.1 ITEM 1- UP-TO-DATE VERSION AND PATCHES

Reference	General knowledge, experience.
Control Objective	System is the latest available version by vendor distribution.
Risk	This item addresses the risk of an un-patched system compromise. The review of the latest release notes for SecurePlatform provides enough information to confirm that running an out-of-date version is not advisable. In particular, references to "do not use the character % in a password" are interesting. Risk Assessment- High
Compliance	The version of SecurePlatform is at the latest build numbers issued by Check Point and the latest Hotfix Accumulator has also been installed for that version. Build numbers are identified in the latest release notes for the product. Important build numbers are the SVN Foundation and the SecurePlatform build.
Testing	Confirm with the administrator the build number of the system. Have the administrator provide the command line menu output of: "cpshared_ver" "ver" The latest release notes can be obtained for comparison purposes at the following URL: http://www.checkpoint.com/techsupport/downloads.jsp
Objective/ Subjective	Objective

3.2 ITEM 2- KNOWN CVE IN PUBLIC DATABASES

Reference	http://www.cert.org/octave/methodintro.html#phase2
Control Objective	Ensure that the running version of SecurePlatform's web user interface is not subject to known vulnerabilities and exposures.
Risk	This checklist item addresses the risk that a product having known vulnerabilities and exposures will be exposed to threat agents and events. Risk Assessment- High
Compliance	Searches for common vulnerabilities and exposures should not return positive results.
Testing	Search CVE databases (ICAT (http://icat.nist.gov/icat.cfm) or Cassandra (https://cassandra.cerias.purdue.edu/main/) for known vulnerabilities and exposures associated with SecurePlatform NG Web User Interface.
Objective/ Subjective	Objective

3.3 ITEM 3- COMMON WEB SERVER VULNERABILITIES

Reference	Rhoades, 62. Also CERT OCTAVE Methodology- http://www.cert.org/octave/methodintro.html#phase2
Control Objective	Ensure that the web server is not subject to known vulnerabilities and exposures typical to web servers.
Risk	This checklist item addresses the risk that a product having known vulnerabilities and exposures will be exposed to threat agents and events. Risk Assessment- High

Compliance	Tests for common vulnerabilities and exposures should not return positive results.
Testing	This test assumes that the web server is proprietary and that the server type cannot be determined easily, requiring common tests to be executed. Run Nessus to determine whether known issues are present in the web server. Web server tests should be tunnelled through SSL using a program such as Stunnel to ensure that they are effective.
Objective/ Subjective	Objective

3.4 ITEM 4- DEFAULT CGI AND OTHER DEFAULT MATERIAL

Reference	Rhoades, 63. Also DISA Field Security Operations. WEB Server Checklist Procedures- http://csrc.nist.gov/pcig/CHECKLISTS/web_checklist_121203.zip
Control Objective	Ensure that the web server does not have default material available to anonymous users.
Risk	This checklist item addresses the risk that known and potentially risky default material is exposed to threat agents and events. Risk Assessment- High
Compliance	Tests for common default CGI and other material should not return positive results.
Testing	Use N-Stealth to evaluate the web server. N-Stealth must be tunnelled through Stunnel in order to properly evaluate the SSL secured web application.
Objective/ Subjective	Objective

3.5 ITEM 5- RESTRICTED ANONYMOUS ACCESS

Reference	Microsoft- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/CL_SecWebs.asp
Control Objective	Ensure that the web server does not allow extensive directory and file access to anonymous users.
Risk	This checklist item addresses the risk that files and directory structure information is exposed to threat agents and events without authentication. Risk Assessment- High
Compliance	For a secure web server, minimal information should be available to anonymous connections. Only the basic functionality required to authenticate the user should be available.
Testing	Manually investigate the website's directory structure. View source on the main page to determine whether other referenced pages and scripts can be accessed without credentials.
Objective/ Subjective	Objective

3.6 ITEM 6- POLICY EXISTS REGARDING ACCESS, REVOCATION OF ACCESS, AND USAGE

Reference	Stein- q.7. Also ISO 17799 Section 9.6 Application Access Control.
Control Objective	Ensure that the web configuration has appropriate policies associated with it addressing access, revocation of access and usage of the interface.
Risk	This checklist item addresses the risk that a user of the system will be granted access permissions that are not required for their job function. It also addresses other risks associated with leaving user accounts enabled that should be revoked. The risk that administrators can claim ignorance concerning appropriate usage of the system is also addressed through policy. For example, the policy may state that the application cannot be used to alter the configuration of the firewall without proper change management controls. It may also identify policy regarding timeframes for usage, etc.

	Risk Assessment- High
Compliance	A policy should exist that addresses who may access the system, when, for what purpose, etc. The policy should also identify procedures for granting and revoking access to the configuration interface. It should identify by job function who should have read access and who may have change access.
Testing	Confirm that written policy exists addressing access. Inventory access accounts and compare the list of accounts to the list of identified administrators to ensure that old accounts are not left on the system.
Objective/ Subjective	Objective

3.7 ITEM 7- ONLY ENCRYPTED COMMUNICATIONS PERMITTED

Reference	CERT security improvement module practice 80- http://www.cert.org/security-improvement/practices/p080.html DISA Field Security Operations. WEB Server Checklist Procedures- http://csrc.nist.gov/pcig/CHECKLISTS/web_checklist_121203.zip
Control Objective	Ensure the web server only allows SSL connections to protect the privacy of communications between the client and the server.
Risk	This checklist item allows the auditor to confirm whether the web server allows unencrypted connections in addition to encrypted connections. The data being sent between the client and the web server is potentially sensitive network related information, passwords, etc. Risk Assessment- High
Compliance	The web server should not accept a connection to web daemon on an unencrypted connection.
Testing	Run a port scan on the SecurePlatform and determine which ports are servicing web requests. Connect to these ports with a normal web browser, ensure that the connection is either redirected to a secure connection or is refused.

Objective/ Subjective	Objective

3.8 ITEM 8- ONLY STRONG CIPHERS PERMITTED

Reference	Granger, p.8. Also UC Berkeley news release, 1/29/97- http://www.berkeley.edu/news/media/releases/97legacy/code.html
Control Objective	Ensure the web server only allows strong ciphers for SSL connections to protect the privacy of communications between the client and the server.
Risk	This checklist item allows the auditor to confirm whether the web server allows weak ciphers in addition to strongly encrypted connections. It is much more feasible given the processing power available today to crack 40-bit and 56-bit encryption whereas 128-bit is still computationally challenging enough as to be not worth the processing effort in most cases. The data being sent between the client and the web server is potentially sensitive network related information, passwords, etc and warrants that strong encryption techniques be applied. Risk Assessment- Medium
Compliance	The web server should not accept a connection to web daemon using a weak cipher.
Testing	Nessus (www.nessus.org) has a plug-in to retrieve which ciphers the web server will accept for SSL negotiation. Acceptance of less than 128-bit encryption will be flagged as an exception.
Objective/ Subjective	Objective

3.9 ITEM 9- WEB CONFIGURATION INTERFACE IS NOT SUSCEPTIBLE TO BRUTE FORCING OF AUTHENTICATION MECHANISM

Reference	Rhoades, 64, 191
Control Objective	Ensure the web server is not susceptible to brute force attack on the authentication mechanism.

Risk	This checklist item addresses the risk that the system will be breached simply due to a brute force attack on the web configuration interface authentication mechanism. The likelihood that this type of attack will be perpetrated against an exposed web service is high. Risk Assessment- High
Compliance	The web application should not allow brute forcing of authentication mechanism.
Testing	Use Brutus through Stunnel to attempt to brute-force the authentication process.
Objective / Subjective	Objective

3.10 ITEM 10- WEB CONFIGURATION INTERFACE IS NOT SUSEPTIBLE TO ACCOUNT LOCKOUT DENIAL OF SERVICE

Reference	Rhoades, 191
Control Objective	Ensure the web server is not susceptible to a denial of service caused by the account lockouts during password attacks.
Risk	This checklist item addresses the risk that the system will be rendered unusable due to account lockout issues. Risk Assessment- Medium
Compliance	The web application should not permanently lock accounts or a similar mechanism to prevent lockout from the system must be available.
Testing	Use Brutus through Stunnel to attempt to brute-force the authentication process using a test account created by the administrator of the system. Review with the administrator of the system the effect of the attack on the system account. Review with the administrator of the system what account lockout features exist which may not be enabled. Ensure that an alternate account with sufficient privileges exists to unlock the default account.
Objective / Subjective	Objective

3.11 ITEM 11- THE SYSTEM LOGS ADMINISTRATIVE ACTIVITY PERFORMED THROUGH THE WEB USER INTERFACE.

Reference	DISA Field Security Operations. WEB Server Checklist Procedures- http://csrc.nist.gov/pcig/CHECKLISTS/web_checklist_121203.zip
Control Objective	Ensure that adequate logging of administrative access and activities is captured when using the web user interface.
Risk	This checklist item addresses the high risk that not enough information is captured in an audit log to adequately determine access and administrative activity performed through the web configuration interface. Risk Assessment- High
Compliance	Log files recording web user interface access and administrative activity must be captured on the local device and stored in a secure directory with read-only access for administrators only.
Testing	Have the administrator issue the command "log list" and provide the output. Review the list of available logs and have the administrator provide copies of logs which should be relevant to the logging of administrative activity at the web configuration interface. Review the information available in these files to determine if the information collected is sufficient for basic access and action auditing purposes.
Objective / Subjective	Objective

3.12 ITEM 12- THE WEB SERVER LOGS ACCESS AND ERRORS

Reference	Apache HTTP Server Log Files- http://httpd.apache.org/docs/logs.html also DISA Field Security Operations. WEB Server Checklist Procedures- http://csrc.nist.gov/pcig/CHECKLISTS/web_checklist_121203.zip also CERT Security improvement module practice 77- http://www.cert.org/security-improvement/practices/p077.html
Control Objective	Ensure that adequate logging of web server related access and error events are captured when using the web user interface.

Risk	This checklist item addresses the risk that not enough information is captured in an audit log to adequately determine what access and what error generating commands and traffic are being seen by the web server. Risk Assessment- High
Compliance	Log files recording web access and error conditions must be captured on the local device. Access logs must capture a minimum of: Host, user, date, time, request(method, path, query), and status Error logs must capture detailed error information that can be correlated with the access logs.
Testing	In order to ensure that adequate logs exist, have the administrator provide copies of the logs. Review the log files to ensure that information is being captured as outlined above.
Objective / Subjective	Objective

3.13 ITEM 13- WEB SERVER ACCESS AND ERROR LOGS HAVE APPROPRIATE PERMISSIONS

Reference	DISA Field Security Operations. WEB Server Checklist Procedures- http://csrc.nist.gov/pcig/CHECKLISTS/web_checklist_121203.zip also Microsoft Corporation- Securing Your Web Server. http://msdn.microsoft.com/library/en-us/dnnetsec/html/THCMCh16.asp?
Control Objective	Ensure the web server log files have appropriate permissions to prevent tampering and modification.
Risk	This checklist item addresses the risk that log files will be modified to hide attacks against the server. Risk Assessment- Medium
Compliance	Ensure that the log files have read-only permission for all users except system, auditors, and root.
Testing	Have the administrator execute ls –l on the log files and review the

	permissions associated with the files. Greater than read access for any user other than system, auditors, and root will be noted as an audit exception.
Objective / Subjective	Objective

3.14 ITEM 14- DISK SPACE RESTRICTED FOR UPLOADED FILES

Reference	Tracy, p. 29.
Control Objective	Ensure that disk space allocated for uploads is restricted in size and is separate from the system partition.
Risk	Allowing uploads to the web server opens up the risk that what is uploaded is not as expected with respect to size in particular. A denial of service could occur if uploads are not restricted properly and the space allotted is not isolated from the system's operating files and swap space. Most systems require a certain amount of free space (swap, etc) to operate. Check Point SecurePlatform allows a token file to be uploaded that facilitates the recovery of lost passwords.
Compliance	The space allotted for the token upload will be restricted and separate from the system and swap partitions.
Testing	Attempt to post file content that is much larger than what is expected. Examine the web server's reaction to this stimulus. 1. Does the server parse the file for format before or after allowing the upload? 2. Does the server disallow the upload? 3. Does the server allow the upload and continue to operate normally? 4. Does the server allow the upload and then fail?
Objective / Subjective	Objective

3.15 ITEM 15- VERBOSE ERROR MESSAGES DO NOT REVEAL EXCESSIVE INFORMATION

Reference	Previous incidents. Example- http://icat.nist.gov/icat.cfm?cvename=CAN-1999-0990

Control Objective	Ensure that the interface's standard user-visible error messages do not disclose excessive and/or sensitive information.
Risk	This checklist item addresses the risk that a server or application error message will provide information and clues to the attacker regarding for example directory structure, supporting applications, version information, etc. Risk Assessment- Medium
Compliance	The web server and application will not provide verbose error messages to users of the system who are not authenticated.
Testing	Provide stimulus that will elicit error conditions for this web configuration interface as follows: ▪ Provide an incorrect user name for login ▪ Provide an incorrect password for login ▪ Upload an incorrect format file for password recovery and record response ▪ Append a long string to the URL
Objective / Subjective	Objective

3.16 ITEM 16- LOGIN POST PROCESS IS SECURE

Reference	Rhodes, 204 also DISA Field Security Operations- Web Server Security Technical Implementation Guide Section 4 http://csrc.nist.gov/pcig/STIGs/webserverstig-v4r1-082903.doc
Control Objective	Ensure that the initial login post cannot be easily manipulated by substituting values for variables in hidden fields or through modifying other form elements.
Risk	This checklist item is designed to address the risk that the login post process when subjected to user input manipulation can be used to force the server into an error condition or imply that the user is already authenticated, for example. Risk Assessment- High

Compliance	The interface should not be easily manipulated and forced into a condition as described above.
Testing	Examine the source code of the initial login page. Identify variables and fields which may be manipulated in the post command following a "submit" of the user's credentials. Manipulate variables in transit using Achilles and note the responses to the stimulus.
Objective / Subjective	Objective

3.17 ITEM 17- LOST PASSWORD TOKEN UPLOAD PROCESS IS SECURE

Reference	Rhodes, 204 also DISA Field Security Operations- Web Server Security Technical Implementation Guide Section 4 http://csrc.nist.gov/pcig/STIGs/webserverstig-v4r1-082903.doc
Control Objective	Ensure that the lost password token upload process cannot be easily manipulated by substituting values for variables in hidden fields or through modifying other form elements.
Risk	This checklist item is designed to address the risk that the lost password token upload process when subjected to user input manipulation can be used to force the server into an error condition or imply that the user is already authenticated, for example by changing a variable from status=error to status=ok. Risk Assessment- High
Compliance	The interface should not be easily manipulated and forced into a condition as described above.
Testing	Examine the source code of the initial login page. Identify variables and fields which may be manipulated in the post command following a "submit" of a invalid token file. Manipulate variables in transit using Achilles and note the responses to the stimulus.

Objective / Subjective	Objective

3.18 ITEM 18- SESSION IDS ARE NOT PREDICTABLE

Reference	Rhodes, p. 141
Control Objective	Evaluate the method used to track users and determine if the methodology employed is secure. Ensure that session IDs are not predictable.
Risk	This checklist item addresses the risk that predictable session IDs will result in user spoofing (cloning). Risk Assessment: Medium
Compliance	The system will use random session IDs for user state tracking.
Testing	Examine the use of session IDs within the interface. Examine session IDs for randomness.
Objective / Subjective	Objective

3.19 ITEM 19- SESSION IDS ARE NOT TRANSMITTED IN PLAIN TEXT

Reference	Rhodes, p. 140 also **CAN-2003-0728**
Control Objective	Ensure that session IDs and authentication tokens are not transmitted in plain text.
Risk	This checklist item addresses the risk that a session ID or authenticated user token will be intercepted between the client and the server. Risk assessment: High
Compliance	The system will use encryption between the client and the server to prevent interception of session IDs.
Testing	Confirm that SSL is used to protect the communication between the client and the server.
Objective / Subjective	Objective

3.20 ITEM 20- HTML SOURCE OF INITIAL LOGIN PAGE AND SUPPORTING PAGES DO NOT REVEAL EXCESSIVE INFORMATION THROUGH COMMENTARY

Reference	Rhodes, p.111
Control Objective	Ensure that the initial login page and related supporting pages that can be accessed without authentication do not disclose excessive information.
Risk	This checklist item addresses the risk that html source discloses excessive and potentially risky information. Risk Assessment- Medium
Compliance	The web application HTML and HTTP source will not disclose more than the minimum required information to perform the basic login functionality.
Testing	Walk through the web server pages that allow connections without authentication using the Achilles web proxy program. Record all HTML source and HTTP content and review for hidden content, excessive comments, improper restrictions on supporting pages, etc.
Objective / Subjective	Objective

4 AUDIT EVIDENCE

4.1 CONDUCT THE AUDIT

All 20 checklist items were tested in conducting the audit. The following 10 checklist items with associated audit evidence are reflective of the most serious security concerns with the interface regardless of whether the result of the test was positive or negative.

4.1.1 ITEM 5- RESTRICTED ANONYMOUS ACCESS

Reference	Microsoft- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/CL_SecWebs.asp
Control Objective	Ensure that the web server does not allow extensive directory and file access to anonymous users.
Risk	This checklist item addresses the risk that files and directory structure information is exposed to threat agents and events without prior authentication taking place. Risk Assessment- High
Compliance	For a secure web server, minimal information should be available to anonymous connections. Only the basic functionality required to authenticate the user should be available.
Testing	Manually investigate the website's directory structure. View source on the main page to determine whether other referenced pages and scripts can be accessed without credentials.
Objective/ Subjective	Objective
Result	**Audit Checklist Item 5 Result:** **Exception**

Audit Evidence

The investigation of the SecurePlatform web configuration interface was carried out using Achilles to inspect all http traffic between the client and the server.

The following logical sequence was followed beginning with the server default page to determine available files and directories. Only interesting line items detailing file names are presented in order to shorten the output for this report and this does not represent the complete HTTP communication sequence. The important point to note is that all tests were performed without authenticating to the server:

Client> GET / HTTP/1.0

Server> HTTP/1.0 200 OK

File references in /
<script src="/wm_index.js"></script>
<script src="/wm_license.js"></script>
<script src="/wm_installation_type.js"></script>
<script src="/wm_status_device.js" ></script>
<script src="/wm_fingerprints.js"></script>
id="wm_main" name="wm_main" src="/wm_index.html?
id="wm_hidden" name="wm_hidden" src="/wm_request.html"
id="wm_swap" name="wm_swap" src="/tmp.html"

Client> GET /wm_index.js

File reference in wm_index.js
/wm_default_admin.html

Client> GET /wm_license.js
GET /wm_installation_type.js
GET /wm_status_device.js
GET /wm_fingerprints.js

File reference in wm_fingerprints.js
/wm_fingerprints.html

Client> GET /wm_index.html?0.6480420820032685 (**session ID appended**)

File references in wm_index.html

<script src="/wmapi.js" >
<script src="/is_compatible.js">

Client> GET /tmp.html (empty HTML File)
GET /wm_request.html (hidden)

File reference in /wm_request.html- /xml.js

Client> GET /styles.css (style sheet)
GET /wmapi.js
GET /xml.js

GET /is_compatible.js

Investigate Lost Password Function Secondary Page (/lostpwd.html)

Client> GET /lostpwd.html

Server> HTTP/1.0 200 OK

File references in /lostpwd.html –

<script src="/is_compatible.js"></script>
<script src="/wmapi.js"></script>
/cgi-bin/cpwm_cgi **(only reference to a CGI is this sequence)**
/img/

Investigate Other Accessible Secondary Pages

/wm_default_admin.html
Change Default Login Name and Password

Client> GET /wm_default_admin.html

Server> HTTP/1.0 200 OK

File references in /wm_default_admin.html

<script src="/wmapi.js">
<script src="/wm_default_admin.js">

Client> GET /wm_default_admin.js

Server> HTTP/1.0 200 OK

File reference in /wm_default_admin.js

/wm_main.html

/wm_main.html

Client> GET /wm_main.html

Server> HTTP/1.0 200 OK

File references in /wm_main.html

wm_title.html
<script src="/wm_menu_entry.js">
<script src="/is_compatible.js">
<script src="/xml.js">
<script src="/wm_main.js">
<script src="/wm_wizard.js">
cpwm_conf.xml

Client > GET /wm_title.html

GET /wm_menu_entry.js
GET /wm_main.js
GET /wm_wizard.js

File reference in /wm_wizard.js

/wm_wizard_main.html

Client> GET /cpwm_conf.xml

File references in /cpwm_conf.xml

wm_title.html
wm_status_device.html
wm_tab_base.html
wm_cpadmin.html
wm_gui.html
wm_sic.html
wm_tab_base.html
wm_network_interfaces.html
wm_routing.html
wm_arp.html
wm_dns.html
wm_hosts.html
wm_tab_base.html
wm_products.html
wm_licstatus.html
wm_tab_base.html
wm_commands.html
wm_time.html
wm_diagnostics.html
wm_backup.html
wm_upgrade.html
wm_logout.html
wm_help.html

Client> GET /wm_wizard_main.html

File references in /wm_wizard_main.html

<script src="/xml.js">
<script src="/wmapi.js">
<script src="/netapi.js">
<script src="/wm_wizard.js">
<script src="/wm_wizard_apply.js">
<script src="/wm_dns.js">
<script src="/wm_gui_admin.js">
<script src="/wm_gui_clients.js">
<script src="/wm_web_clients.js">
<script src="/wm_product.js">
<script src="/wm_wizard_interface.js">
<script src="/wm_connection.js">
<script src="/wm_router.js">

```
<script src="/wm_fqdn.js">
<script src="/wm_lsm.js">
<script src="/wm_installation_type.js">
<script src="/wm_wizard_objects.js">
```

One can continue to extract additional information from the files referenced in /wm_wizard_main.html and /cpwm_conf.xml above however, the auditor believes that the point has been well demonstrated that Check Point SecurePlatform NG (R55) web configuration interface allows anonymous access to much more information than is necessary to facilitate the secure login process. The javascript code with its references to enticing variables such as authenticate_ administrator_status (wm_index.js) in particular provides an excellent start for any attacker to build up an attack approach by examining the client side scripts for potential variables to manipulate and flaws to exploit.

Important Note Concerning this Audit Process!

It is important to note that during the configuration process, a mechanism exists to permit only those workstations which the administrator of the system wishes to be able to connect to the web interface from. This is a temporary protection measure until the first Check Point security policy is installed (see screenshot below). After the first policy is installed, this list is ignored in favor of access being controlled via firewall policy. This audit makes the assumption that the firewall software is either not configured correctly or is not operating thus exposing the interface to threat agents and events. Consider the administrator who wishes to be able to access the web interface from anywhere on the corporate net. The firewall rule he adds is **corporate net > firewall > https > allow** thus exposing the interface to internal threats. What about the administrator that feels that the interface is secure enough to stand on its own (it uses SSL after all) and allows access to it from anywhere for those quick and easy changes? The intent is to audit the interface without consideration for firewall security policy. Based on these criteria, access restrictions prior to initial policy installation are not a consideration. Protecting the interface itself is not a solution for limiting anonymous access permissions to a secure interface.

Conclusion: Audit Exception

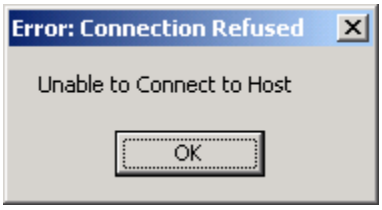
4.1.2 ITEM 7- ONLY ENCRYPTED COMMUNICATIONS PERMITTED

Reference	CERT security improvement module practice 80- http://www.cert.org/security-improvement/practices/p080.html DISA Field Security Operations. WEB Server Checklist Procedures- http://csrc.nist.gov/pcig/CHECKLISTS/web_checklist_121203.zip
Control Objective	Ensure the web server only allows SSL connections to protect the privacy of communications between the client and the server.
Risk	This checklist item allows the auditor to confirm whether the web server allows unencrypted connections in addition to encrypted connections. The data being sent between the client and the web server is potentially sensitive network related information, passwords, etc. Risk Assessment- High

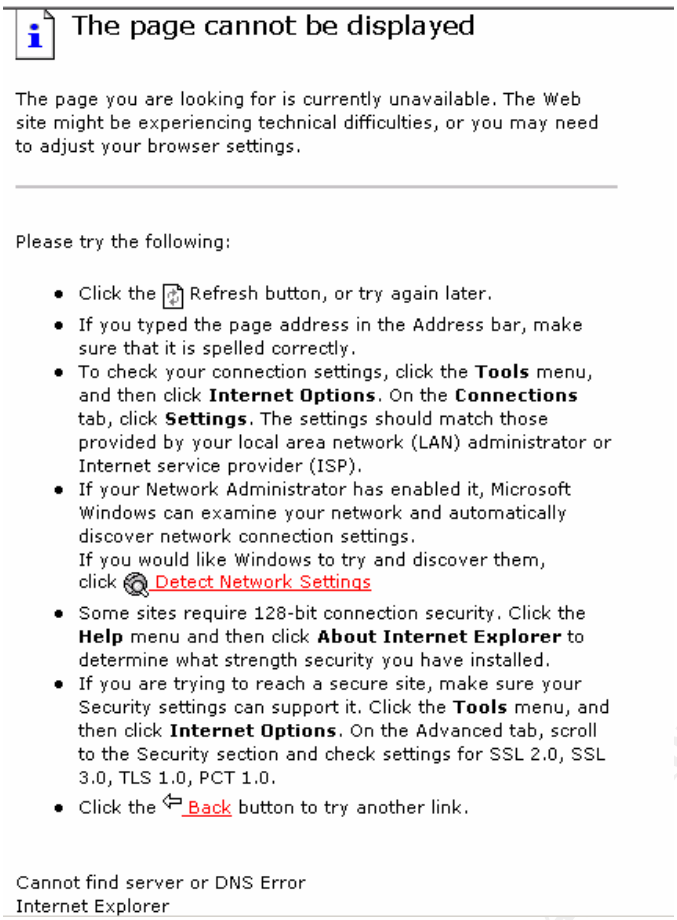
Compliance	The web server should not accept a connection to web daemon on an unencrypted connection.
Testing	Run a port scan on the SecurePlatform and determine which ports are servicing web requests. Connect to these ports with a normal web browser, ensure that the connection is either redirected to a secure connection or is refused.
Objective/ Subjective	Objective
Result	**Audit Checklist Item 7 Result:** **Compliant**

Audit Evidence

Connect to http://host/ yields a connection refused result:



Connect to http://host:443 result:


The page cannot be displayed

The page you are looking for is currently unavailable. The Web site might be experiencing technical difficulties, or you may need to adjust your browser settings.

Please try the following:

- Click the Refresh button, or try again later.
- If you typed the page address in the Address bar, make sure that it is spelled correctly.
- To check your connection settings, click the **Tools** menu, and then click **Internet Options**. On the **Connections** tab, click **Settings**. The settings should match those provided by your local area network (LAN) administrator or Internet service provider (ISP).
- If your Network Administrator has enabled it, Microsoft Windows can examine your network and automatically discover network connection settings. If you would like Windows to try and discover them, click Detect Network Settings
- Some sites require 128-bit connection security. Click the **Help** menu and then click **About Internet Explorer** to determine what strength security you have installed.
- If you are trying to reach a secure site, make sure your Security settings can support it. Click the **Tools** menu, and then click **Internet Options**. On the Advanced tab, scroll to the Security section and check settings for SSL 2.0, SSL 3.0, TLS 1.0, PCT 1.0.
- Click the Back button to try another link.

Cannot find server or DNS Error
Internet Explorer

Conclusion: Compliant

4.1.3 ITEM 8- ONLY STRONG CIPHERS PERMITTED

Reference	Granger, p.8. Also UC Berkeley news release, 1/29/97- http://www.berkeley.edu/news/media/releases/97legacy/code.html
Control Objective	Ensure the web server only allows strong ciphers for SSL connections to protect the privacy of communications between the client and the server.
Risk	This checklist item allows the auditor to confirm whether the web server allows weak ciphers in addition to strongly encrypted connections. It is much more feasible given the processing power available today to crack 40-bit and 56-bit encryption whereas 128-bit is still computationally challenging enough as to be not worth the processing effort in most cases. The data being sent between the client and the web server is potentially sensitive network related information, passwords, etc and warrants that strong encryption techniques be applied.

	Risk Assessment- Medium
Compliance	The web server should not accept a connection to web daemon using a weak cipher.
Testing	Nessus (www.nessus.org) has a plug-in to retrieve which ciphers the web server will accept for SSL negotiation. Acceptance of less than 128-bit encryption will be flagged as an exception.
Objective/ Subjective	Objective
Result	**Audit Checklist Item 5 Result:** **Exception**

Audit Evidence

Nessus Plugin ID 10330 results

"A web server is running on this port through SSL"
"A SSLv2 server answered on this port"

Nessus Plugin ID 10863 results

"Here is the SSLv2 server certificate:

```
Certificate:
   Data:
      Version: 1 (0x0)
      Serial Number: 1804289383 (0x6b8b4567)
      Signature Algorithm: sha1WithRSAEncryption
      Issuer: CN=192.168.199.1
      Validity
         Not Before: Jan 24 13:00:15 2004 GMT
         Not After : Jan 21 13:00:15 2014 GMT
      Subject: CN=192.168.199.1
      Subject Public Key Info:
         Public Key Algorithm: rsaEncryption
         RSA Public Key: (1024 bit)
            Modulus (1024 bit):
               00:a6:a3:3e:93:61:49:a3:1d:f4:b2:bc:d7:11:1d:
               e1:83:45:07:d3:00:f0:2e:02:77:b0:00:d6:75:a1:
               18:9e:1e:48:fb:9d:d3:a8:52:b9:c8:71:60:be:78:
               5f:8e:3f:4e:d6:4a:97:7f:ef:dd:17:a1:df:b7:61:
```

```
            2e:9a:97:02:a0:39:ad:b6:d0:7a:68:8a:74:87:b9:
            8b:25:05:15:d8:e4:87:c8:1b:c0:16:27:91:d2:20:
            73:05:0f:a5:98:25:79:12:ac:27:c6:6a:a0:83:a6:
            85:7b:11:bb:9b:3d:ee:f3:84:2e:48:9c:0d:ae:c5:
            75:75:ca:99:e9:94:ef:87:53
        Exponent: 3 (0x3)
Signature Algorithm: sha1WithRSAEncryption
    a0:d0:3c:5e:d6:d0:5f:e4:1a:6b:4f:86:14:d1:9b:a7:98:e8:
    65:39:08:0b:b8:ed:0f:f8:34:fd:41:31:c1:f0:2c:9a:81:9e:
    a7:62:cb:0c:80:69:8d:6c:40:b3:15:4a:b3:21:26:fc:63:4b:
    2f:49:e3:bc:35:04:55:97:1c:8b:ba:90:68:42:69:bc:b5:6c:
    33:d1:6a:f2:d8:8c:9e:ce:84:67:bf:51:07:db:8e:d6:3f:b1:
    57:75:24:cd:b2:6e:71:b2:6e:e4:53:0c:d8:c6:38:a7:55:19:
    59:51:b4:5c:88:db:ca:cf:f6:a1:62:f5:18:29:0f:55:3b:88:
    56:b2"
```

"Here is the list of available SSLv2 ciphers:
RC4-MD5
EXP-RC4-MD5
DES-CBC-MD5
DES-CBC3-MD5
RC4-64-MD5"

"This SSLv2 server also accepts SSLv3 connections."
"This SSLv2 server also accepts TLSv1 connections."

"The SSLv2 server offers 4 strong ciphers, but also 0 medium strength and 1 weak "export class" ciphers. The weak/medium ciphers may be chosen by an export-grade or badly configured client software. They only offer a limited protection against a brute force attack

Solution: disable those ciphers and upgrade your client software if necessary"

Conclusion: Audit Exception

4.1.4 ITEM 9- WEB CONFIGURATION INTERFACE IS NOT SUSEPTIBLE TO BRUTE FORCING OF AUTHENTICATION MECHANISM

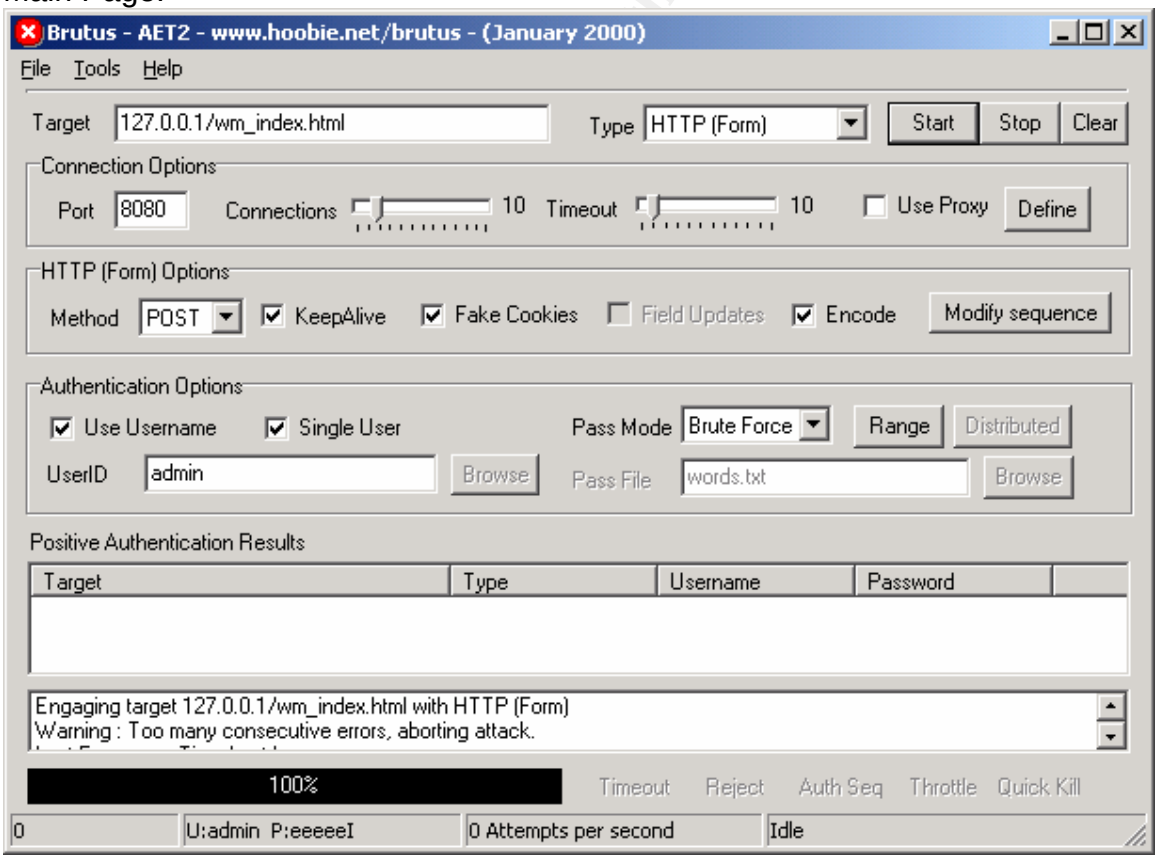
Reference	Rhoades, 64, 191
Control Objective	Ensure the web server is not susceptible to brute force attack on the authentication mechanism.
Risk	This checklist item addresses the risk that the system will be breached simply due to a brute force attack on the web configuration interface authentication mechanism. The likelihood that this type of attack will be perpetrated against an exposed web service is high.

	Risk Assessment- High
Compliance	The web application should not allow brute forcing of authentication mechanism.
Testing	Use Brutus through Stunnel to attempt to brute-force the authentication process.
Objective / Subjective	Objective
Result	**Audit Checklist Item 9 Result:** **Compliant**

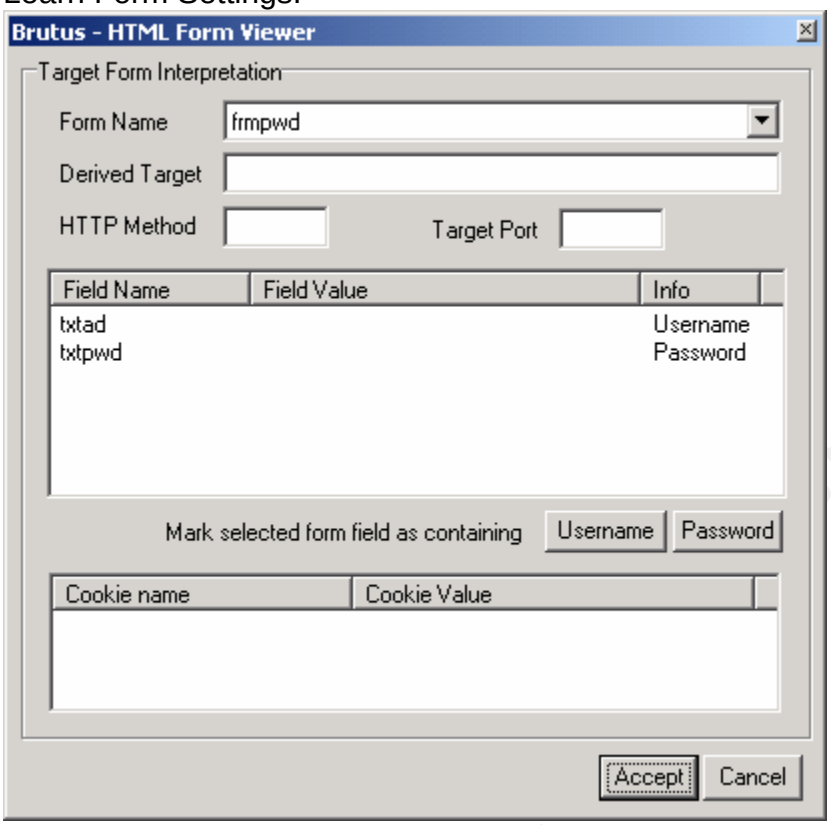
Audit Evidence:

Brutus Configuration for SecurePlatform NG (R55) web configuration interface form based authentication:

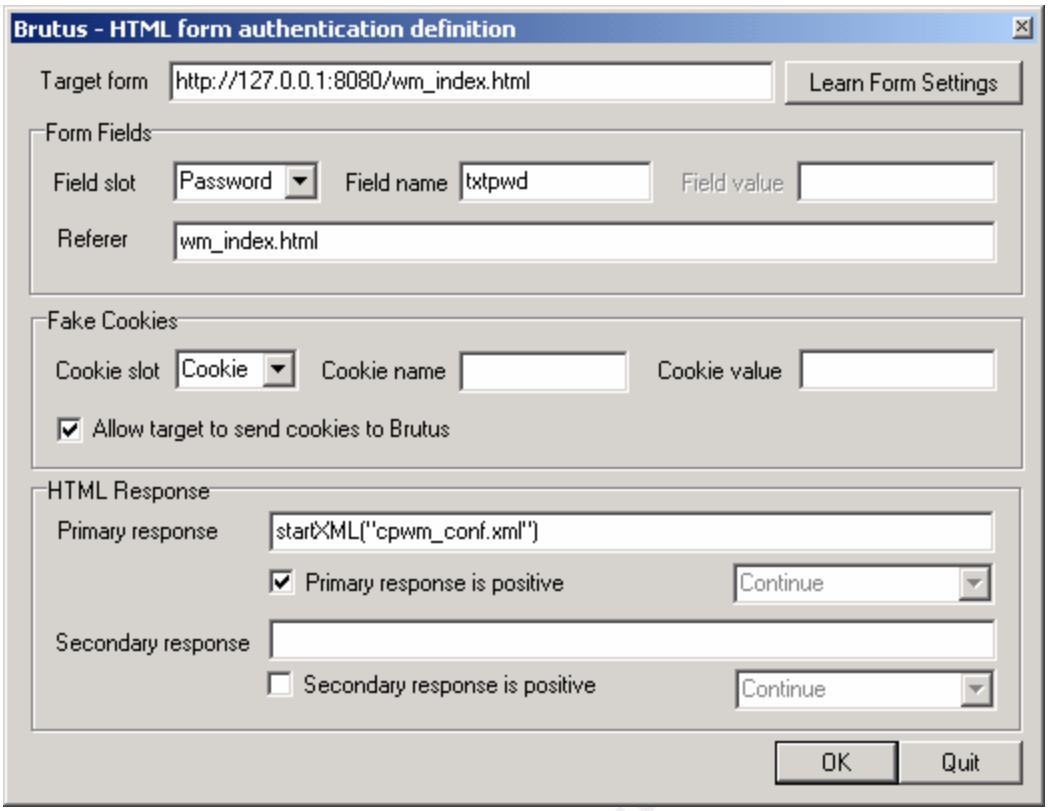
Main Page:



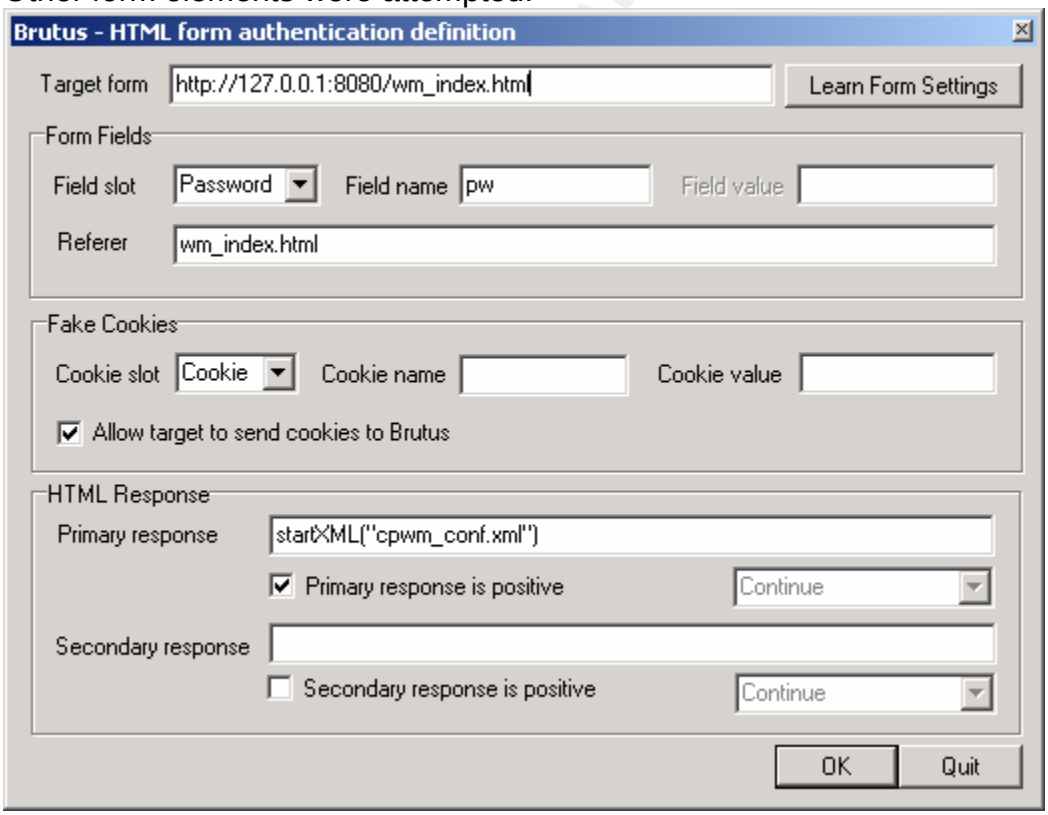
Learn Form Settings:



HTTP Form Options > Modify Sequence



Other form elements were attempted:



After making many adjustments to the username/password and form sequence options within the configuration of Brutus, a decision was made by the auditor to abandon further efforts to use Brutus to brute force the password authentication process for the Check Point SecurePlatform web configuration interface. This interface does not appear to be trivial to brute force using standard methods.

Conclusion: Compliant

4.1.5 ITEM 11- THE SYSTEM LOGS ADMINISTRATIVE ACTIVITY PERFORMED THROUGH THE WEB USER INTERFACE.

Reference	DISA Field Security Operations. WEB Server Checklist Procedures- http://csrc.nist.gov/pcig/CHECKLISTS/web_checklist_121203.zip
Control Objective	Ensure that adequate logging of administrative access and activities is captured when using the web user interface.
Risk	This checklist item addresses the high risk that not enough information is captured in an audit log to adequately determine access and administrative activity performed through the web configuration interface. Risk Assessment- High
Compliance	Log files recording web user interface access and administrative activity must be captured on the local device and stored in a secure directory with read-only access for administrators only.
Testing	Have the administrator provide copies of /var/log/secure, /var/log/messages and /opt/CPshared-R55/log/cp_httpd_server.elg. Review the information available in the files to determine if the information collected is sufficient for auditing purposes based on corporate security policy.
Objective / Subjective	Objective in confirming that such logs exist. Subjective as to forming an opinion as to whether the logging levels meet policy objectives.
Result	**Audit Checklist Item 11 Result:** **Exception**

Audit Evidence

According to the SecurePlatform administrator's "log list" the following logs are available:

Index	File	Index	File
0)	messages	16)	arlogind.elg
1)	wtmp	17)	asessiond.elg
2)	lastlog	18)	asmtpd.elg
3)	secure	19)	aufpd.elg
4)	cpstart.log	20)	genericd.elg
5)	fwd.elg	21)	lhttpd.elg
6)	dtlsd.elg	22)	pingd.elg
7)	dtpsd.elg	23)	mdq.elg
8)	sdsd.elg	24)	snauth.elg
9)	vpnd.elg	25)	cp_http_server.elg
10)	aftpd.elg	26)	cpwd.elg
11)	atelnetd.elg	27)	cpd.elg
12)	ahttpd.elg	28)	rtmd.elg
13)	unifiedd.elg	29)	fgd.elg
14)	aclientd.elg	30)	boot.log
15)	ahclientd.elg		

The following logs are specific to Check Point security products that may be installed and are not considered in this audit.

4)	cpstart.log
5)	fwd.elg
6)	dtlsd.elg
7)	dtpsd.elg
8)	sdsd.elg
9)	vpnd.elg
10)	aftpd.elg
11)	atelnetd.elg
12)	ahttpd.elg
13)	unifiedd.elg
14)	aclientd.elg
15)	ahclientd.elg
16)	arlogind.elg
17)	asessiond.elg
18)	asmtpd.elg
19)	aufpd.elg
20)	genericd.elg
21)	lhttpd.elg
22)	pingd.elg
23)	mdq.elg
24)	snauth.elg
26)	cpwd.elg
27)	cpd.elg
28)	rtmd.elg
29)	fgd.elg

This leaves the following logs that can be reviewed for information pertaining to administrative activity at the web interface:

Index	File
0)	messages
1)	wtmp
2)	lastlog
3)	secure
25)	cp_http_server.elg
30)	boot.log

Of the above list, boot.log can easily be eliminated, wtmp is related to system up / down status, and lastlog is a virtually empty file (likely for log swaps).

As it turns out, "Messages" records most information concerning administrator access on the system via the web configuration interface. This file also holds all the traditional kernel level messages as well so there is a requirement to use a facility such as Grep to sort through the file. As a note, Grep is not included in the default SecurePlatform distribution.

Examples of relevant log entries from "Messages" are identified below:

- Jan 30 08:05:46 cptestsp55 cpwebui (pam_unix)[628]: authentication failure; logname= uid=0 euid=0 tty=tty0 ruser= rhost= user=admin

- Jan 31 12:55:17 cptestsp55 cpwmd[627]: System administrator password was changed [tester], operation performed by admin

However, not all administrator access is captured in "messages". The "secure" log file receives actual account (add/delete/mod) actions performed on the web interface.

- Jan 31 12:37:36 cptestsp55 adduser[989]: new user: name=tester, uid=0, gid=0, home=/home/tester, shell=/bin/cpshell

The last log to examine is the cp_http_server.elg file. This file contains error messages related to the server itself:

rand_add_seedfile: Failed to create mutex.: Operation not permitted
rand_add_external_source: Failed to create mutex.: Operation not permitted
fwrand_write_seed: Failed to create mutex.: Operation not permitted

It is not clear to the auditor that a mechanism exists to log administrative activity performed via the web configuration interface. This was tested by added entries to the host file, adding a secondary address to an interface, etc. and then searching the available log files for entries related to these actions. None were found except those relating to account administration and administrator access activities.

Log archiving capabilities do exist for available logs and logs can be adjusted for size and number of back logs via the command line interface. There are other modes of operation available for

Conclusion: Exception

4.1.6 ITEM 12- THE WEB SERVER LOGS ACCESS AND ERRORS

Reference	Apache HTTP Server Log Files- http://httpd.apache.org/docs/logs.html also DISA Field Security Operations. WEB Server Checklist Procedures- http://csrc.nist.gov/pcig/CHECKLISTS/web_checklist_121203.zip also CERT Security improvement module practice 77- http://www.cert.org/security-improvement/practices/p077.html
Control Objective	Ensure that adequate logging of web server related access and error events are captured when using the web user interface.
Risk	This checklist item addresses the risk that not enough information is captured in an audit log to adequately determine what access and what error generating commands and traffic are being seen by the web server. Risk Assessment- High
Compliance	Log files recording web access and error conditions must be captured on the local device. Access logs must capture a minimum of: Host, user, date, time, request(method, path, query), and status Error logs must capture detailed error information that can be correlated with the access logs.
Testing	In order to ensure that adequate logs exist, have the administrator provide copies of the logs. Review the log files to ensure that information is being captured as outlined above.
Objective / Subjective	Objective

Result	**Audit Checklist Item 12 Result:** **Exception**

Based on previous audit checklist items, the relevant file for the web server related log is cp_http_server.elg. In order to generate events and provide stimulus for this log facility a top 20 scan using N-Stealth was executed against the web server using Stunnel to provide the SSL support. The log file was acquired and examined following this stimulus and the following new entries were noted after reviewing the provided log file:

fwasync_call_mux_in: 11: internal error: inbuf.state=0
rand_add_external_source: Failed to create mutex.: Operation not permitted
cp_uri_parse: forbidden escape representation in URI (possibly layred %<xx> times and times again)
cp_uri_parse: forbidden escape representation in URI (possibly layred %<xx> times and times again)
cp_uri_parse: forbidden escape representation in URI (possibly layred %<xx> times and times again)
cp_uri_parse: forbidden escape representation in URI (possibly layred %<xx> times and times again)
….
Continues to end of file

These error messages, while alerting the administrator of the system that something is wrong are generally not very helpful. There is no date stamp, no host address, no detail on method, etc and as such it does not meet the audit criteria. As an aside, this test also confirmed our assumption in checklist item 3 that a parsing method to detect special characters is in force.

One other final point to note is that the contents of cp_http_server.elg are not maintained when the web interface is disabled and enabled or when the system is rebooted. Although the "log list" command shows that this file is supposed to have up to 4 back logs, none could be found on the system following a cycling of the web server daemon.

Conclusion: Exception

4.1.7 ITEM 14- DISK SPACE RESTRICTED FOR UPLOADED FILES

Reference	Tracy, p. 29.
Control	Ensure that disk space allocated for uploads is restricted in size and is

Objective	separate from the system partition.
Risk	Allowing uploads to the web server opens up the risk that what is uploaded is not as expected with respect to size in particular. A denial of service could occur if uploads are not restricted properly and the space allotted is not isolated from the system's operating files and swap space. Most systems require a certain amount of free space (swap, etc) to operate. Check Point SecurePlatform allows a token file to be uploaded that facilitates the recovery of lost passwords. Risk Assessment: Medium
Compliance	The space allotted for the token upload will be restricted and separate from the system and swap partitions.
Testing	Attempt to post file content that is much larger than what is expected. Examine the web server's reaction to this stimulus. 1. Does the server parse the file for format before or after allowing the upload? 2. Does the server disallow the upload? 3. Does the server allow the upload and continue to operate normally? 4. Does the server allow the upload and then fail?
Objective / Subjective	Objective
Result	**Audit Checklist Item 14 Result:** **Exception**

Audit Evidence

Without logging into the system, there is a facility whereby a token file for authentication can be uploaded to the server. This feature provides a mechanism to handle issues around lost passwords. This feature was explored to determine whether it is compliant with the checklist objective.

Does the server parse the file for format before or after allowing the upload?

Attempt to upload a test file containing "bogus token"

Server response:

HTTP/1.0 200 OK
Date: Wed, 04 Feb 2004 00:36:26 GMT
Server: Check Point SVN foundation
Connection: close
Content-type: text/html

<HTML><script src="/xml.js"></script><body bgcolor=#252525><script>
importXMStr('<webgui_query_result request_id="LOST_PASSWORD" status="ok" reason=""><authenticate_administrator_status status=**"error"** reason="Authentication **failure**"/></webgui_query_result>');</script></BODY></HTML>

The upload function appears to have a parsing function.

Attempt to transfer a large file (4 MB) to determine if the parsing is taking place before or after the upload. Observing the output of TCP dump confirmed that the file is transferred to the server.

```
20:52:41.435704 auditor.3546 > cptestsp55.https: . 3651262:3652722(1460) ack 580
 win 63661 (DF)
20:52:41.435707 auditor.3546 > cptestsp55.https: . 3652722:3654182(1460) ack 580
 win 63661 (DF)
20:52:41.435709 auditor.3546 > cptestsp55.https: . 3654182:3655642(1460) ack 580
 win 63661 (DF)
20:52:41.435710 auditor.3546 > cptestsp55.https: . 3655642:3657102(1460) ack 580
 win 63661 (DF)
20:52:41.435712 auditor.3546 > cptestsp55.https: . 3657102:3658562(1460) ack 580
 win 63661 (DF)
20:52:41.435713 auditor.3546 > cptestsp55.https: . 3658562:3660022(1460) ack 580
 win 63661 (DF)
20:52:41.435714 auditor.3546 > cptestsp55.https: P 3660022:3660358(336) ack 580
win 63661 (DF)
```

Try to transfer a very large file, observing through TCP dump the progress of the transfer.

Result: The system appears to only allow the transfer of a plain text file. So the server does not disallow the upload but does restrict it on the client side to a plain text file. The parsing function to actual check whether the file is a valid token must take place on the server side.

A quick look at the source code for the page reveals the relevant code that probably checks for plain text:

```
function sendFile()
{
	if(!isCompatible())return;
	document.fp.action = m_activ;
	document.fp.request_id.value ="LOST_PASSWORD";
	document.fp.token.value ="";
	document.fp.request_data.value ="<upload_file></upload_file>";
	document.fp.target='wm_swap';
	if(document.fp.upload_file.value)
	{
		document.fp.submit();
	}
}
function openFileDialog()
```

Both during and after the large file upload, additional sessions were opened to the server without issue. Based on these results, it appears that the application is at least making basic checks to ensure that uploads are plain text and are compatible with the token file format since the server is sending back error code information. It is not apparent however

that file size of uploads is being restricted. This could potentially lead to a denial of service condition if a very large plaintext file was uploaded. To explore this further, a larger (25Mb) file was uploaded and this test was repeated several times. During the tests, so sporadic issues with connectivity to the server were observed but the server did not crash. The tests also generated some log entries in the cp_http_server.elg error log as follows indicating that the CGI had problems processing the request:

```
handle_cgi_io: failed to write to child...
fwasync_call_mux_in: 6: internal error: inbuf.state=0
handle_cgi_io: failed to write to child...
fwasync_call_mux_in: 8: internal error: inbuf.state=0
handle_cgi_io: failed to write to child...
rand_add_external_source: Failed to create mutex.: Operation not permitted
handle_cgi_io: failed to write to child...
handle_cgi_io: failed to write to child...
handle_cgi_io: failed to write to child...
fwasync_call_mux_in: 6: internal error: inbuf.state=0
[Expert@cptestsp551# _
```

Even though these entries are not time stamped, their appearance in the log did coincide with the upload activities.

Summary, it does not appear that the interface restricts the size of the upload file on the client side but it does seem that the CGI has a limited buffer to fill regarding this upload and that when it does reach capacity; the server logs an error condition as identified above. None of the above tests crashed the server but it may be theoretically possible to fill up the server's disk space using the upload feature creating a DoS condition. It is interesting to note that a valid token is a very small single line file which could be easily checked for boundary conditions.

Conclusion: Exception

4.1.8 ITEM 16- LOGIN POST PROCESS IS SECURE

Reference	Rhodes, 204 also DISA Field Security Operations- Web Server Security Technical Implementation Guide Section 4 http://csrc.nist.gov/pcig/STIGs/webserverstig-v4r1-082903.doc
Control Objective	Ensure that the initial login post cannot be easily manipulated by substituting values for variables in hidden fields or through modifying other form elements.
Risk	This checklist item is designed to address the risk that the login post process when subjected to user input manipulation can be used to force the server into an error condition or imply that the user is already

	authenticated, for example by changing a variable from status=error to status=ok. Risk Assessment- High
Compliance	The interface should not be easily manipulated and forced into a condition as described above.
Testing	Examine the source code of the initial login page. Identify variables and fields which may be manipulated in the post command following a "submit" of the user's credentials. Manipulate variables in transit using Achilles and note the responses to the stimulus.
Objective / Subjective	Objective
Result	**Audit Checklist Item 16 Result:** **Compliant**

Audit Evidence

Relevant source code of initial login page (retrieved without credentials)

```
function doSubmit() {

var admin_id ="AUTHENTICATE_ADMINISTRATOR";

      if(!isCompatible())return;
      var nm=document.frmpwd["txtad"].value;
      var pw=document.frmpwd["txtpwd"].value;
      if(nm==""||pw=="")
      {
            if(nm=="")alert('Enter Login name');
            else if(pw=="")alert('Enter Password');
            return ;
      }
      top.new_credentional_name =nm;
      top.new_credentional_id =pw;
      var astr = '<authenticate_administrator name="'+nm+'"
password="'+pw+'"></authenticate_administrator>'
      window.status = "Logging in. Please Wait...";
      top.wm_hidden.getRequest(astr,admin_id,top);
}

Also,
<form name="frmpwd" onsubmit="return false">
      <table class=Tab border=0 cellpadding=5 cellspacing=0 style="width:300px;">
```

```
        <tr class=TabHead><th colspan=2 align=left>Enter your password:</th></tr>
        <tr class=TabRow0>    <td>Username:</td>    <td  align="rigth"><INPUT value="" type="text"
name="txtad"  maxlength="16" onkeyup="javascript:nextField(event);"></td></tr>
        <tr class=TabRow0><td>Password:</td><td align="rigth"><INPUT value="" type="password"
name="txtpwd" maxlength="16" onkeydown="javascript:isEnter(event);"></td></tr>
        <tr class=TabRow0><td align="center" colspan=2><script  LANGUAGE="javascript">

createButton("dd","Enter","doSubmit();");

</script></td>
        </tr></table>
</form>
```

As can be seen in the above HTML source code, there are several variables and INPUT values which are of interest. Both the user name and password are restricted to a length of 16 in the form fields for example. Here is the post command following "doSubmit()":

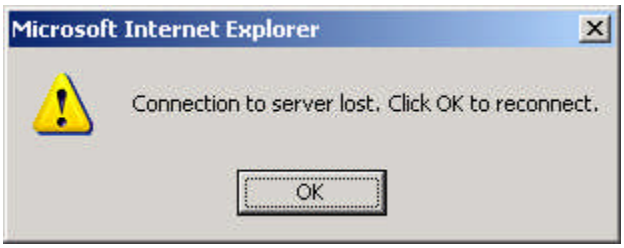
```
POST /cgi-bin/cpwm_cgi HTTP/1.0
Accept: */*
Accept-Language: en-us
Referer: https://172.19.174.4/wm_request.html
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.0.3705)
Host: 172.19.174.4
Content-Length: 150
Cache-Control: no-cache

request_id=AUTHENTICATE_ADMINISTRATOR&token=&request_data=<authenticate_administrator
name="admin" password="whatever">
```

So, cpwm_cgi is the main gatekeeper of the system and all post commands observed during the course of this audit were targeted to this CGI.

Test 1: Replace password in the Post command with a very long string. Remove the content length statement.

```
POST /cgi-bin/cpwm_cgi HTTP/1.0
…
request_id=AUTHENTICATE_ADMINISTRATOR&token=&request_data=<authenticate_ad
ministrator name="admin" password="<long string > 4000
chars">
```

Test 1 Response:

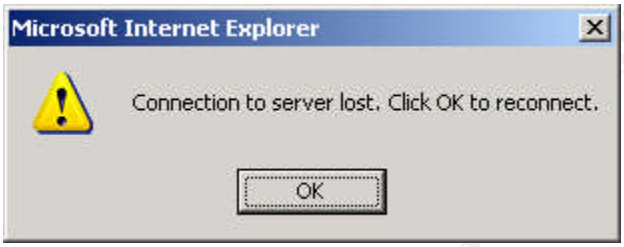


Server reloads main page.

Test 2: Replace user name in the Post command with a very long string. Remove the content length statement.

POST /cgi-bin/cpwm_cgi HTTP/1.0
…
request_id=AUTHENTICATE_ADMINISTRATOR&token=&request_data=<authenticate_administrator **name="<long string > 4000 chars"**
password="aaaaaa"**>**</authenticate_administrator>

Test 2 Response:

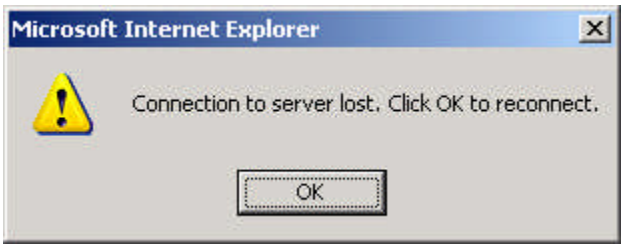


Server reloads main page.

Test 3: Insert long "token" in the Post command. Remove the content length statement.

POST /cgi-bin/cpwm_cgi HTTP/1.0
…
request_id=AUTHENTICATE_ADMINISTRATOR&token=**<long string > 4000 chars**
&request_data=<authenticate_administrator name="admin"
password="whatever"></authenticate_administrator>

Test 3 Response:



Server reloads main page.

The server's CGI appears to parse the input and impose some boundaries correctly. This was not an exhaustive test.

Conclusion: Compliant

4.1.9 ITEM 17- LOST PASSWORD TOKEN UPLOAD PROCESS IS SECURE

Reference	Rhodes, 204 also DISA Field Security Operations- Web Server Security Technical Implementation Guide Section 4 http://csrc.nist.gov/pcig/STIGs/webserverstig-v4r1-082903.doc
Control Objective	Ensure that the lost password token upload process cannot be easily manipulated by substituting values for variables in hidden fields or through modifying other form elements.
Risk	This checklist item is designed to address the risk that the lost password token upload process when subjected to user input manipulation can be used to force the server into an error condition or imply that the user is already authenticated, for example by changing a variable from status=error to status=ok. Risk Assessment- High
Compliance	The interface should not be easily manipulated and forced into a condition as described above.
Testing	Examine the source code of the initial login page. Identify variables and fields which may be manipulated in the post command following a "submit" of a invalid token file. Manipulate variables in transit using Achilles and note the responses to the stimulus.

Objective / Subjective	Objective
Result	**Audit Checklist Item 17 Result:** **Exception**

It should be noted that the upload functionality was examined in checklist item 14 and was found to have some issues with restricting the format and size of the token upload file. This checklist item continues to examine this upload process for additional issues.

Audit Evidence

Relevant source code of lost password page (retrieved without credentials)

```
var m_activ="/cgi-bin/cpwm_cgi";
…
function sendFile()
{
	if(!isCompatible())return;
	document.fp.action = m_activ;
	document.fp.request_id.value ="LOST_PASSWORD";
	document.fp.token.value ="";
	document.fp.request_data.value ="<upload_file></upload_file>";
	document.fp.target='wm_swap';
	if(document.fp.upload_file.value)
	{
		document.fp.submit();
	}
…
<form name="fp" METHOD="POST" ENCTYPE="multipart/form-data" onsubmit="return false;"
target="wm_swap">
…
<input type=hidden name="request_id"><input type=hidden name="request_data"><input type=hidden
name="token">
```

Testing

On submit the token file is uploaded to the server (as long as it is plain text). The server response to the invalid token is as follows:

```
HTTP/1.0 200 OK
Date: Wed, 04 Feb 2004 14:16:34 GMT
Server: Check Point SVN foundation
Connection: close
Content-type: text/html

<HTML><script src="/xml.js"></script><body bgcolor=#252525><script>
```

importXMStr('<webgui_query_result request_id="LOST_PASSWORD" status="ok" reason=""><**authenticate_administrator_status status="error"** reason="**Authentication failure**"/></webgui_query_result>');</script></BODY></HTML>

Now, there is an opportunity here to trick the client into believing the transaction was successful by substituting some results:

<authenticate_administrator_status **status="ok" reason="Authentication success**"/>

Result: The client begins to load the main menu pages.

GET /wm_main.html HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, application/x-shockwave-flash, */*
Referer: https://192.168.199.4/
Accept-Language: en-us
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.0.3705)
Host: 192.168.199.4

Server response> HTTP/1.0 200 OK

Communications continue and the client proceeds to load most of the entire menu system consisting of many pages and scripts (many of these were identified earlier through manually crawling through the site but many of these are also new source files not previously seen because they weren't previously referenced).

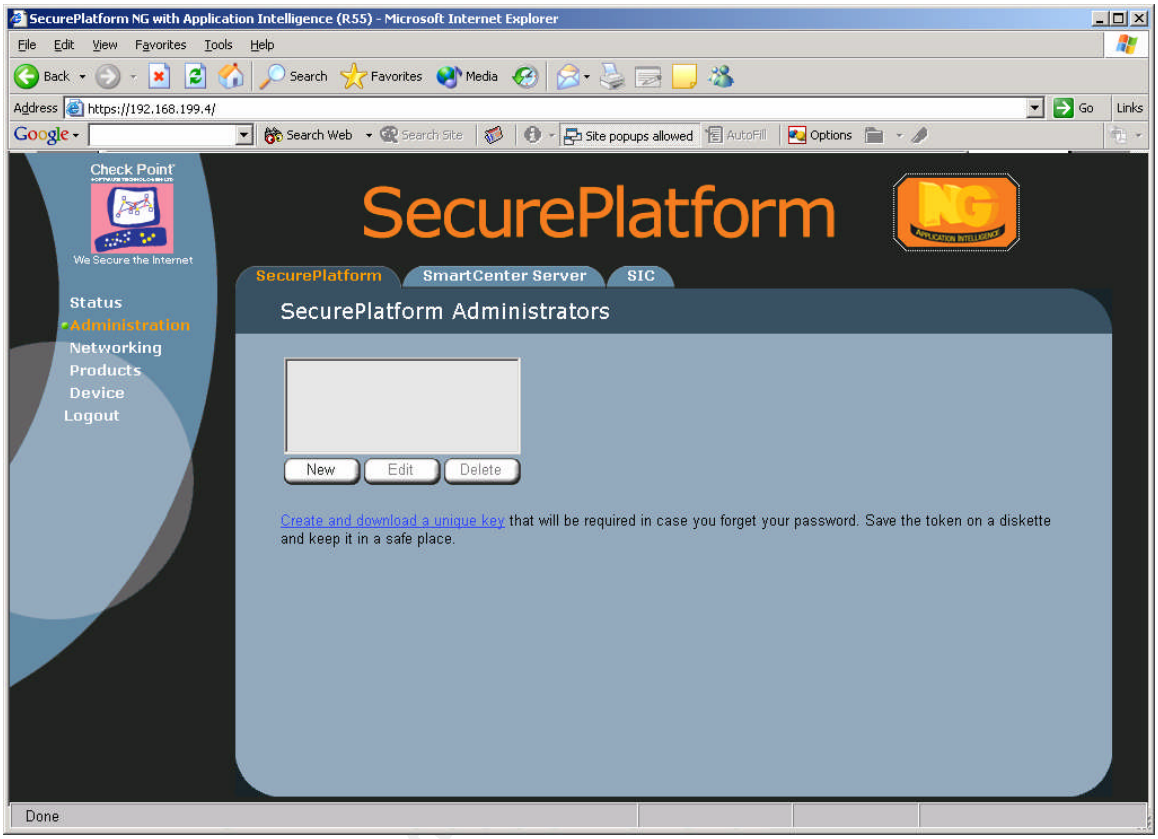
However, we still are not authenticated and the gatekeeper eventually requires another status update. Here's what the status update POST to the CGI looks like:

POST /cgi-bin/cpwm_cgi HTTP/1.0
Accept: */*
Accept-Language: en-us
Referer: https://192.168.199.4/wm_request.html
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.0.3705)
Host: 192.168.199.4
Content-Length: 71
Cache-Control: no-cache

request_id=KEEP_ALIVE&**token=null**&request_data=<keep_alive></keep_alive>

Token=null means we are likely not getting any further into the system. This token process was examined further and the token is a long string, essentially a new session ID that is generated following successful authentication. During navigation of the site the client is frequently require to update the gatekeeper CGI with the token. Inserting the pre-authentication token does not help the process continue and retrieving a valid token without authentication seems unlikely.

End result, after trying this a few times and acknowledging a couple of different error messages (session invalid and connection lost) in the process, access to an empty menu system which one can even navigate through was achieved but there is no actual system data in the fields and any actions attempted (add user for example) result in a system timeout on the operation. This does not really represent a compromise of the system but it did allow the client to retrieve the entire source of the site for offline analysis.



Empty menu system

Conclusion: Exception

4.1.10 ITEM 18- SESSION IDS ARE NOT PREDICTABLE

Reference	Rhodes, p. 141
Control Objective	Evaluate the method used to track users and determine if the methodology employed is secure. Ensure that session IDs are not predictable.
Risk	This checklist item addresses the risk that predictable session IDs will result in user spoofing (cloning).
Compliance	The system will use random session IDs for user state tracking.
Testing	Examine the use of session IDs within the interface. Examine

	session IDs for randomness.
Objective / Subjective	Objective
Result	**Audit Checklist Item 18 Result:** **Compliant**

Audit Evidence

Pre-authentication session IDs are passed between the client and the server either in the URL or in the referrer field:

GET /wm_index.html?0.7111782939791258 HTTP/1.0

Referer: https://192.168.199.4/wm_index.html?0.7111782939791258

These session IDs appear to cover a wide range of numbers and seem to be random in that they vary in size. A few simple mathematical relationships were tried (addition, subtraction, multiplication but no relationship between session IDs is apparent. Here are some samples harvested from the interface prior to login:

0.005753401609984055
0.049562788020579784
0.4798848604345778
0.8852629975371284
0.5317156306220518
0.7725272106957378
0.9508305521359475
0.12352705699551236
0.3362342680987218
0.06680781620607479
0.1822415602222115
0.1822415602222115

Pre-login session IDs are not very valuable anyway since they are not carried through into the authenticated session.

Post-authentication user tracking is maintained using a token field which is sent periodically to the gatekeeper CGI:

POST /cgi-bin/cpwm_cgi HTTP/1.0
…
request_id=KEEP_ALIVE&**token=<token>**&request_data=<keep_alive></keep_alive>

This token is even larger than the pre-login session IDs and it does not seem feasible to acquire or predict what this token will be unless by chance there is some sort of relationship created by using the pre-session ID as a seed. This would require extensive mathematical analysis and is not examined further in this paper.

Conclusion: Compliant

4.2 MEASURE RESIDUAL RISK

The Firewall in any organization plays a crucial role in maintaining security between network segments with varying levels of trust between them. Firewalls today also provide many more services than just access control including VPN and authentication services. From a business operation point of view, the firewall is the gateway to critical information available on the Internet or through partner connections. It may be securing an e-commerce web DMZ that is the core of an organizations business process.

In the context of what a firewall represents from an operational and business processing perspective, configuration and control of the firewall are critical issues. Some of the issues discovered through this paper can be addressed through configuration changes on the system but many of them can probably only be addressed through vendor improvements in subsequent releases or through updates. Issues relating to permissions for web folders will require the vendor to separate html source files into different directories (anonymous, authenticated) in order to resolve anonymous access issues.

None of the issues identified in this paper led to a serious compromise of firewall configuration control. Nevertheless, many of the issues that were identified create an increased risk of a further compromise potentially being possible.

The overall objective of the audit was to develop and audit process for evaluating a secure web configuration interface. This checklist was then applied against a relevant product in the Check Point Secure Platform NG (R55) web configuration interface. For most checklist items, the control objectives were straightforward and were confirmed through testing as being achieved or not achieved. That said, some checklist objectives may have been a bit too broad, especially those concerning user input manipulation as they left many questions remaining unanswered.

In summary, the checklist developed seems a good start for auditing a proprietary web based configuration interface including others such as Sonicwall, Nokia Voyager, and many more. The intent of the audit is to help management and security officers in organizations make a decision on whether or not to allow configuration of the firewall using these web interfaces. It is important to realize that the firewall security policy cannot always be guaranteed to protect these interfaces and there will be administrators who expose this interface to external connections.

The audit process outlined herein still needs refinement and some of the broader checks need to be narrowed down somewhat to provide more significant and specific findings. The amount of remaining residual risk in the system that has not been measured through this checklist is represented by the wide scope of the user input manipulation testing items but it is the belief of the auditor that the main gatekeeper CGI of the web configuration interface seems to enforce session control very strictly which does mitigate the residual risk to a large extent.
.

4.3 IS THE SYSTEM AUDITABLE?

The checklist and audit process outlined in this document provides a reasonable first step in determining whether or not to use a web configuration interface for a device such as a firewall. Customization of some of the steps in this audit to reflect differences in vendor approaches to securing the interface will definitely need to be developed.

Checklist items such as "Administrative Access and Activity Performed Through the Web Interface are Logged" are clear cut and the findings conclusive. As stated in the previous section, some of the broader input manipulation items will need to be broken down into checklists steps specific to the vendor interface being audited.

Should one rely strictly on the vendor and administrators of the system to worry about the security of the interface? No

Should an audit process such as this be applied to secure web configuration interfaces? Yes because their use may be optional as is the case with SecurePlatform NG.

Is a secure web configuration interface auditable using this audit process? Yes

5 AUDIT REPORT

5.1 EXECUTIVE SUMMARY

The purpose of this report is to clearly and concisely outline audit findings with respect to Check Point SecurePlatform NG's web configuration interface. Security is especially important in the development of a web interface for configuring a security device because the running configuration and operational parameters of the system can be modified and saved from the web user interface. Were this interface to be compromised, it could lead to the compromise of the underlying operating system. If this happens at the gateway of the corporate network, the risk of further penetration into the network is greatly increased.

The goals and objectives of this audit were formulated to identify risk associated with firewall administrators using the web configuration interface for modifying the firewall's running configuration. The checklist items tested were designed to measure whether control objectives were achieved covering a fairly broad number of potential risks and issues.

In summary, there are several exceptions reported in this document that bear consideration when weighing the benefit versus risk in employing the web interface for configuration control. There is a menu driven command line interface that can be accessed via SSH in a secure manner and that command line interface performs all of the functions of the web user interface. Use of the command line interface should not cost the organization in additional training costs given that it is menu driven. Given these facts, it is the auditor's recommendation that the command line menu interface be used instead of the web configuration interface. In fact, the device should be run in FIPS 140-2 compliant mode as this mode not only disables the web user interface but also automatically enables account lockout and timeouts to prevent brute force password attacks. This administration mode should be enforced until such time as a patch or workaround can be obtained from the vendor to address anonymous access issues, web server logging, and a potential problem identified with file uploads is clarified by the vendor.

5.2 AUDIT FINDINGS, RISK, AND RECOMMENDATIONS

The following table is an overall summary of findings that identify each checklist item and indicates the compliance of the web configuration interface in meeting the checklist control objective.

Checklist item	Description	Result
1	Up-to-date Version and Patches	**Compliant**
2	Known CVE in Public Databases	**Compliant**
3	Common Web Server Vulnerabilities	**Compliant**
4	Default CGI and Other Default Material	**Compliant**
5	Restricted Anonymous Access	**Exception**
6	Policy Exists regarding Access, Revocation of Access, and Usage	**Compliant**
7	Only Encrypted Communications Permitted	**Compliant**
8	Only Strong Ciphers Permitted	**Exception**
9	Web Configuration Interface is not Susceptible to Brute Forcing of Authentication Mechanism	**Compliant**
10	Web configuration Interface is not Susceptible to Account Lockout Denial of Service	**Compliant**
11	The System Logs Administrative Activity Performed Through the Web User Interface.	**Exception**
12	The Web Server Logs Access and Errors	**Exception**
13	Web Server access and Error Logs Have Appropriate Permissions	**Compliant**
14	Disk Space Restricted for Uploaded Files	**Exception**
15	Verbose Error Messages Do Not Reveal Excessive Information	**Compliant**
16	Login POST Process is Secure	**Compliant**

17	Lost Password Token Upload Process is Secure	**Exception**
18	Session IDs are Not Predictable	**Compliant**
19	Session IDs are Not transmitted in Plain Text	**Compliant**
20	HTML Source of Initial Login Page and supporting pages DO Not Reveal Excessive Information through Commentary	**Compliant**

The table above outlines several exceptions that were noted during the course of executing the audit checklist. There were many positive results in this audit process which can be construed from the audit checklist results above. However, this report is primarily concerned with providing recommendations regarding the exceptions identified, therefore, the rest of this section deals with the exceptions only. If review of additional compliant checklist items not detailed in this report is required, supplementary audit evidence can be provided. The following subsection is a brief summary of each exception with references to supporting material.

5.2.1 EXCEPTION ANALYSIS

5.2.1.1 Exception 1- Restricted Anonymous Access

Risk Assessment- High

Finding:

The basic problem identified is that there are no restrictions placed on navigation through the web server source pages and script files using anonymous access credentials. By reviewing the HTML source of the primary login page it was possible to determine through reference numerous other supporting pages and script files (see Section 4.1.1 p.27). These files could be retrieved anonymously and they in turn had numerous references which create linkage to other content intended for authenticated users. This probably was not intended by developers of the interface.

In summary, it was possible to retrieve most of the site content without authenticating. This does not allow and attacker to perform actions using the web interface but does allow the material to be downloaded and reviewed offline to search for potential logical flaws and vulnerabilities.

Risk:

The likelihood of an attacker retrieving as much of the source material for the site as possible prior to implementing an attack is high. Allowing the contents of the site to be retrieved anonymously helps the attacker to gather information. The risk that this represents is hard to quantify as there are mitigating factors which are unknown such as how well the code is written, etc that are not easily measured. It is a fact however that allowing the bulk of material to be retrieved without credentials increases the risk to the system by providing a starting point for investigation.

Recommendation:

The vendor will need to be involved in order to resolve this issue in the long term. It also cannot be resolved easily by changing permissions on the source files themselves as they are all in the root folder directory on the server. It may be possible to a certain extent to restrict access to most files using individual file permissions. The long term solution is to separate anonymous access pages and authenticated access pages into separate directories with permission by group on the authentication access pages. This architecture needs to be supported in the html source file references. For example a reference to /wm_main.html will need to be changed to /authenticated/wm_main.html. Another approach would be to move the users working directory to the authenticated subfolder. This would avoid most of the recoding issues revolving around file references.

5.2.1.2 Exception 2- Only Strong Ciphers Permitted

Risk Assessment- Medium

Finding:

The server allows one weak export class cipher for SSL communications. See section 4.1.3, p. 35 for details. This cipher is provided because of Check Point's worldwide distribution status.

Risk:

The likelihood of someone trying to crack the SSL encryption keys for any given session is fairly low. However, based on the checklist objective and supporting references, a minimum of 128-bit encryption is recommended to ensure security between the client and the server. The risk of this exception being exploited is low-medium depending on the value of the target.

Recommendation:

It may be possible to acquire a recommendation from the vendor on a specific procedure to remove the weak cipher support or to acquire a patch.

5.2.1.3 Exception 3- The System Logs Administrative Activity Performed Through the Web User Interface.

Risk Assessment- High

Finding:

Administrative access to the SecurePlatform web configuration interface is adequately logged. These entries are recorded in the var/log/messages log file in an appropriate format with timestamps and user information. Administrator activity performed while in the configuration interface is not adequately logged. While changes to account status (add/delete) could be found in the var/log/secure log file, other administrator actions such as adding a host file, adding a secondary IP to an interface, etc. could not be discovered in any log file (see Section 4.1.5 p. 40).

Risk:

Not having adequate logs regarding administrator activity represents a high risk that actions will be performed on the system that cannot be traced back to any particular individual. This is against the security principle of accountability. Also, for change management purposes, a log of administrative activity can provide a back out process to reverse changes. The system cannot be audited properly for security policy verification without adequate logs.

Recommendation:

It may be possible to adjust the logging level of the Check Point HTTP daemon. Contact the vendor for a workaround to increase the logging level.

5.2.1.4 Exception 4- The Web Server Logs Access and Errors

Risk Assessment- High

Finding:

While some administrative activity is logged with respect to account activity, this checklist item was intended to confirm that standard web server access and error logs exist to track basic information such as host, user, date, time, request(method, path, query), and status. Error logs must be available that capture detailed error information that can be correlated with the access logs.

The primary log file for the Check Point web server is the cp_http_server.elg file. Among other issues, this file's contents are not maintained through a reboot of the system or cycling of the web service. The log entries that do should up in the file after providing extensive stimulus to the web server are generally not very helpful. An example is as follows:

cp_uri_parse: forbidden escape representation in URI (possibly layred %<xx> times and times again)

This log entry does indicate there is a problem but it obviously doesn't meet basic criteria for information (host, date, time, etc) to assist in tracking the issue.

Risk:

Not having adequate logs regarding web server access and error conditions represents a high risk that actions will be performed on the system that cannot be traced back to any particular source. Any security incident or even operational problem involving the web configuration interface cannot be investigated fully using these logs as a basis.

Recommendation:

Again, it may be possible to adjust the logging levels of the Check Point HTTP daemon. Contact the vendor for a workaround to increase the logging level. There is a debugging instance of the operating system available in the Grub boot menu but having to operate the server in a non-default mode is not an acceptable solution.

5.2.1.5 Exception 5- Disk Space Restricted for Uploaded Files

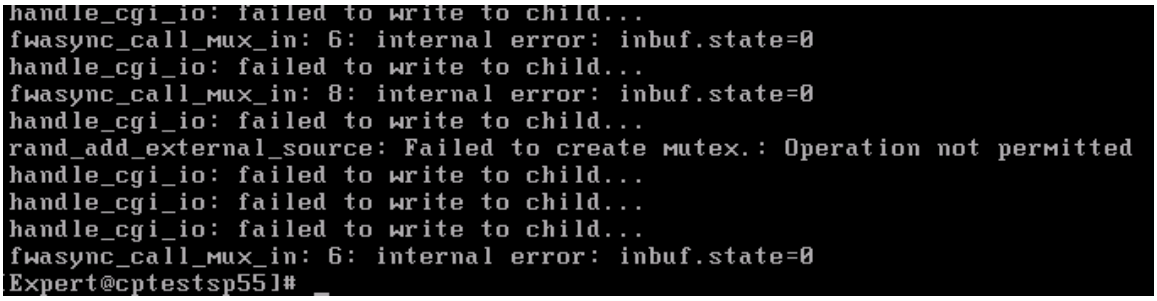
Risk Assessment- Medium

Finding:

The Check Point web configuration interface provides a mechanism for recovering lost passwords via a token file upload. Upon examination, it was discovered that the parsing of the file for correct format seems to be limited to confirming the file is a plain text file. Other types of files were tried unsuccessfully.

The reason for noting this upload process as an exception is that the server allows the upload of a very large text file even though a single line text file is all that is required to store the token.

It was possible to upload a very large text file (25MB) to temporary swap space. Based on several errors noted, this action did appear to have an affect on the application CGI that handles Post requests and uploads. A snapshot of these messages is as follows:



Due to the log record format issue, it is impossible to perfectly correlate time with action but the entries appeared at some point late in the upload process. These entries would seem to indicate that the CGI had difficulties in processing the request although the server never crashed and a second session was able to be initiated to the system.

Risk:

If discovered, this upload process will definitely be examined for the potential to upload executables. Allowing large files to be uploaded may theoretically allow for the disk space of the system to be filled up. This could cause operational issues including a possible halting of the logging process for Check Point firewall related logs among other issues.

Recommendation:

Again, it may be possible to adjust the logging levels of the Check Point HTTP daemon. Contact the vendor for a workaround to increase the logging level. There is a debugging instance of the operating system available in the Grub boot menu but having to operate the server in a non-default mode is not an acceptable solution.

5.2.1.6 Exception 6- Lost Password Token Upload Process is Secure

Risk Assessment- High

Finding:

The token upload process described in the previous section is subject to a user input manipulation flaw. It is possible to change the response that is sent back from the server to fool the client-side scripts into believing the process was successful:

<authenticate_administrator_status status="**error**" reason="Authentication **failure**"/>

Can be changed to:

<authenticate_administrator_status status="**ok**" reason="Authentication **success**"/>

This does not mean that authenticated access was achieved, but by injecting these changes, it was possible to fool the client-side scripts into loading the configuration menu system. Combined with the lack of anonymous access restrictions, an empty menu system can be loaded. The system does eventually request an authentication token update which requires a response bearing a long token string. Without that token, actions cannot be performed in the menu system nor can information be even read. What is eventually loaded is just an empty menu system.

Risk:

There are many automated tools that can perform javascript and forms analysis vulnerabilities. Given the exposure of the site to a threat, the real risk lies in the fact that the entire site contents can be retrieved without authentication and taken offline for analysis with these tools. This could potentially lead to further attacks when plans of action or likely vulnerabilities have been identified by the attacker. Again, much of the site contents could be retrieved anyway (see exception 1) because of the loose access restrictions but this little manipulation allows the entire site to be retrieved.

Recommendation:

There are several ways to mitigate this risk, the primary one being to correct the loose access controls on anonymous connections. Without this access, only a small portion of the site would be accessible to support the login process. Another way to mitigate this risk is obfuscating the status codes instead of using obvious status messages. This would help only if the status code for successful login was never reused for scripts related to anonymous access.

5.3 COSTS

The costs required to implement the recommendations outlined in the audit are very dependant on the approach used. The cost could be as little as nothing or a minor training cost in switching administration methodology to the command line menu driven system.

The cost associated with having the vendor correct these issues is impossible to predict. Certainly as a base cost, a valid support contract with the vendor will be required. Typically, this support cost is around 25% of the product purchase cost. For an implementation involving several firewalls and a product cost of $100,000 the yearly support cost would be $25,000 for example. This is not really an additional cost that would be incurred by the client organization as the contract would already likely be in place to support the overall product. This would entitle the organization to work with vendor support to acquire workarounds and patches for the system.

Ultimately the root cause of the problem is probably too much reliance on the firewall software product to protect the interface in the first place. To change this approach in future revisions would represent a significant change in this mindset and undoubtedly cause additional development costs to be incurred by the vendor.

5.4 COMPENSATING CONTROLS

The basic compensating control to implement here is to operate the system in FIPS 140-2 compliant mode. This disables the web interface and enables some password lockout facilities among other things. Other potential compensating controls are:

- A strict change management process (mitigates risk of unknown administrator activity).
- A policy concerning designated management workstations for web clients.
- A secure proxy mechanism to record web server access and actions.

Some compensating controls are already in place such as the protection provided by the firewall product itself. This cannot be strictly relied upon however as discussed throughout this paper as the interface could still be exposed were a mistake (actual or in judgement) made. As well, the firewall logs will record all connections to the firewall and unknown connections should be examined during the log review process.

6 REFERENCES

Alberts, Christopher and Dorofee, Audrey. "An Introduction to the OCTAVE Method". Software Engineering Institute, Carnegie Mellon University. January, 2001 URL: http://www.cert.org/octave/pubs.html (31 Jan 2004)

Carnegie Mellon. "CERT Security Improvement Modules- Securing Public Web Servers". June 19, 2002. URL: http://www.cert.org/security-improvement/modules/m11.html (31 Jan 2004)

Check Point Software Technologies. User Manual- SecurePlatform NG With Application Intelligence (R55). November, 2003.

Check Point Software Technologies. "Facts @ a Glance". January, 2004. URL: http://www.checkpoint.com/corporate/facts.html (31 Jan 2004)

Check Point Software Technologies. "Government Certified Solutions". 2002. URL: http://www.checkpoint.com/products/downloads/government_certification.pdf (31 Jan 2004)

DISA Field Security Operations. "Web Server Security Technical Implementation Guide Version 4, Release 1". 29 August 2003. URL: http://csrc.nist.gov/pcig/STIGs/webserverstig-v4r1-082903.doc (31 Jan 2004)

DISA Field Security Operations. "Web Server Checklist Procedures Version 4.0 Release 1". 29 August 2003. URL: http://csrc.nist.gov/pcig/CHECKLISTS/web_checklist_121203.zip (31 Jan 2004)

Fritsch, Jörg. www.linux-magazine.com- "Check Point SecurePlatform with Firewall-1-Quick Hardening".. March, 2003 URL: http://www.linux-magazine.com/issue/28/CheckPointSecurePlatform.pdf (31 Jan 2004)

Granger, Sarah. InFocus- "Unlocking the Secrets of Crypto: Cryptography, Encryption, and Cryptology Explained". August 2002. URL: http://www.securityfocus.com/infocus/1617 (31 Jan 2004)

Hoelzer, David. SANS Training Course. Track 7, Module 7.1-Auditing Principles and Concepts. 2003

International Association for Standardization (ISO). ISO/IEC 17799:2000 (E)- International Standard ISO/IEC 17799 Information Technology — Code of Practice for Information Security Management. December 1, 2000

Microsoft Corporation. "Checklist: Securing Your Web Server". June 2003
URL: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/CL_SecWebs.asp (31 Jan 2004)

Rhoades, David. SANS Training Course. Track 7, Module 7.3-Auditing Web Servers and Applications Version 1.6. 2003

Stein, Lincoln and Stewart, John. "The World Wide Web Security FAQ Version 3.1.2". February, 2002. URL: http://www.w3.org/Security/Faq/www-security-faq.html (31 Jan 2004)

Tracy, Miles, Jansen, Wayne and MacLarnon, Mark- NIST Special Publication 800-44- "Guidelines on Securing Public Web Servers". September 2002.
URL: http://csrc.nist.gov/publications/nistpubs/800-44/sp800-44.pdf (31 Jan 2004)

Last Updated: July 28th, 2017

Upcoming Training

SANS Network Security 2017	**Las Vegas, NV**	**Sep 10, 2017 - Sep 17, 2017**	**Live Event**
SANS AUD507 (GSNA) @ Canberra 2017	**Canberra, Australia**	**Oct 09, 2017 - Oct 14, 2017**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**