



Solution Guide

Endpoint Defense

Release 4.1.1

Juniper Networks, Inc.

1194 North Mathilda Avenue
Sunnyvale, CA 94089

USA

408-745-2000

www.juniper.net

411b09162004Cd

Juniper Networks, the Juniper Networks logo, NetScreen, NetScreen Technologies, the NetScreen logo, NetScreen-Global Pro, ScreenOS, and GigaScreen are registered trademarks of Juniper Networks, Inc. in the United States and other countries.

Juniper Networks, the Juniper Networks logo, NetScreen, NetScreen Technologies, Neoteris, Neoteris-Secure Access, Neoteris-Secure Meeting, NetScreen-SA 1000, NetScreen-SA 3000, NetScreen-SA 5000, IVEGigaScreen, and the NetScreen logo are registered trademarks of Juniper Networks, Inc. NetScreen-5GT, NetScreen-5XP, NetScreen-5XT, NetScreen-25, NetScreen-50, NetScreen-100, NetScreen-204, NetScreen-208, NetScreen-500, NetScreen-5200, NetScreen-5400, NetScreen-Global PRO, NetScreen-Global PRO Express, NetScreen-Remote Security Client, NetScreen-Remote VPN Client, NetScreen-IDP 10, NetScreen-IDP 100, NetScreen-IDP 500, GigaScreen ASIC, GigaScreen-II ASIC, and NetScreen ScreenOS are trademarks of Juniper Networks, Inc. All other trademarks and registered trademarks are the property of their respective companies.

Copyright © 2001 D. J. Bernstein. Copyright © 1985-2003 by the Massachusetts Institute of Technology. All rights reserved. Copyright © 2000 by Zero-Knowledge Systems, Inc. Copyright © 2001, Dr Brian Gladman <brg@gladman.uk.net>, Worcester, UK. All rights reserved. Copyright © 1989, 1991 Free Software Foundation, Inc. Copyright © 1989, 1991, 1992 by Carnegie Mellon University. Derivative Work - 1996, 1998-2000. Copyright © 1996, 1998-2000 The Regents of the University of California. All Rights Reserved. Copyright © 1999-2001 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted. Copyright © 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland. All rights reserved. Copyright © 1986 Gary S. Brown. Copyright © 1998 CORE SDI S.A., Buenos Aires, Argentina. Copyright © 1995, 1996 by David Mazieres <dm@lcs.mit.edu>. Copyright © 1998-2002. The OpenSSL Project. All rights reserved. Copyright © 1989-2001, Larry Wall. All rights reserved. Copyright © 1989, 1991 Free Software Foundation, Inc. Copyright © 1996-2002 Andy Wardley. All Rights Reserved. Copyright © 1998-2002. Canon Research Centre Europe Ltd. Copyright © 1995-1998. Jean-loup Gailly and Mark Adler.

Endpoint Defense Solution Guide, Release 4.1.1

Copyright © 2004, Juniper Networks, Inc.
All rights reserved. Printed in USA.

Writer: Claudette deGiere

Revision History

16 September 2004

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Contents

Endpoint defense overview	3
Host Checker overview	4
Defining Host Checker policies	5
Implementing Host Checker policies	6
Installing Host Checker	7
Executing Host Checker policies	7
Cache Cleaner overview	8
Cache Cleaner execution	9
Configuring endpoint defense options	11
Security > Host Checker tab	11
Security > Cache Cleaner tab	15
Host Checker restrictions	17
Cache Cleaner restrictions	18
Installers tab	19
Host Checker interfaces	21
Host Check Client Interface	22
Signing your custom DLL	22
Deploying and maintaining your custom DLL	23
NetScreen Host Checker (NHC) API	23
Host Check Server Integration Interface	26
Deploying third party applications through Host Checker	26
Creating your interface DLL	27

Endpoint defense overview

Juniper Networks has developed the Juniper Endpoint Defense Initiative (J.E.D.I.) to provide a comprehensive solution to assess the trust worthiness of SSL VPN endpoints. J.E.D.I. uses a layered approach to address the full range of risks that endpoints can pose to your enterprise network. Using J.E.D.I. components, you can secure the systems of users inside and outside your network before allowing them to connect to your IVE appliance. J.E.D.I. components include:

- **Native Host Checks and policy-based enforcement**

Native Host Check (also called Host Checker) is a native IVE component that you can use to perform endpoint checks on hosts that connect to the IVE. You can use Host Checker to ensure that specified processes, files, registry entries, ports, or integrated third-party endpoint security products conform to your specifications before allowing a user access to an IVE realm, role, or resource policy. For example, you may use Host Checker's third-party product checking capability to specify that a user can only access a particular IVE role if he has enabled a personal firewall on his system. You can also use host checks to a loosely-coupled integration with systems that are not yet J.E.D.I. compliant. For more information, see "Host Checker overview" on page 4.

- **Host Check Client Interface**

The Host Check Client Interface is an API that allows you to communicate with DLLs using Host Checker or a J.E.D.I. compliant DLL. Through the interface, you can prompt Host Checker to run a DLL that you have already installed on the user's system or distributed as part of a corporate OS image, including programs that check compliance with corporate images, antivirus software, and personal firewall clients. Host Checker runs the specified DLL when a user signs into the IVE, and then bases its subsequent actions on the success or failure result returned by your DLL. For example, you may deny a user access to the IVE if the client check software fails. For more information, see "Host Check Client Interface" on page 22.

- **Host Check Server Integration Interface**

The Host Check Server Integration Interface is an API that allows you to tightly integrate J.E.D.I. compliant system with the IVE. Like the Host Check Client Interface, you can use the Host Check Server Integration Interface to prompt Host Checker to run third-party software on the client, including host integrity scans, malware detectors, and virtual environments. With this interface, you may also specify with extreme granularity what Host Checker should do based on the result of these diverse policy checks conducted by the third-party applications. You can invoke these policies to dynamically map users to realms, roles, and resources based on the results of individual policies contained in your software package. For more information, see "Host Check Server Integration Interface" on page 26.

- **Cache Cleaner**

Cache Cleaner is a native IVE component that you can use to clear residual data, such as temporary files or application caches, off of a user's machine after an IVE session. Cache Cleaner helps secure the user's system by preventing subsequent users from finding temporary copies of the files that the previous user was viewing. For more information, see "Cache Cleaner overview" on page 8.

Using these endpoint defense components, you may develop a layered protection approach, managing and provisioning a variety of endpoint checks all from within the IVE. For example, you may choose to check for virus detection and personal firewall software before allowing a user access to any of the IVE realms, launch the software on the user's system if necessary, map the user to roles based on individual policies defined in your own DLL, and then further restrict access to individual resources based on the existence of spyware detection software. Then, you may use Cache Cleaner to remove residual files and clear the user's application cache once the user has left his IVE session.

Host Checker overview

Host Checker is a client-side agent that performs endpoint checks on hosts that connect to the IVE. You can invoke Host Checker before displaying an IVE sign-in page to a user and when evaluating a role mapping rule or resource policy. The IVE may check hosts for endpoint properties using:

- **The Host Checker implementation of a supported endpoint security application**

The ActiveX control calls the Host Checker integration of the specified third-party endpoint security product and examines the return value to see if the product is running in accordance with its configured policies. The IVE currently supports tight Host Checker integration with:

- Sygate Enforcement API
- Sygate Security Agent
- Zone Labs: ZoneAlarm Pro and Zone Labs Integrity
- McAfee Desktop Firewall 8.0
- InfoExpress CyberGatekeeper Agent

- **Host Checker integration using a custom DLL**

The Host Check Client Interface enables you to write a DLL that performs customized client-side checks. You must install this DLL on each client machine.

- **Attribute checking**

The ActiveX control looks for the specified application fingerprints, including processes, files, and registry entries.

For more information, see:

- "Defining Host Checker policies" on page 5
- "Implementing Host Checker policies" on page 6
- "Installing Host Checker" on page 7
- "Executing Host Checker policies" on page 7

Defining Host Checker policies

In order to use Host Checker as a policy enforcement tool for managing endpoints, you must create global Host Checker policies at the system level and then implement the policies at the realm, role, and resource policy levels.

When creating Host Checker policies through the Web console, you must specify host checking methods and/or rule settings. A **host checking method** is the Host Checker implementation of a third-party endpoint security product. A method determines if an application is running in accordance the policies that you have configured. Currently, Host Checker provides methods for:

- Sygate Enforcement API
- Sygate Security Agent
- Zone Labs: ZoneAlarm Pro and Zone Labs Integrity
- McAfee Desktop Firewall 8.0
- InfoExpress CyberGatekeeper Agent

A **host checking rule** is a requirement that a client must meet in order for Host Checker to return a success value to the IVE. You can specify five types of rules:

- **3rd Party NHC check**

Use this rule to specify the location of a custom DLL that you write with the Host Check Client Interface. Host Checker calls the DLL to perform customized client-side checks. If the DLL returns a success value to Host Checker, then the IVE considers the rule met. For information about creating a custom DLL using the Host Check Client Interface, see “Host Check Client Interface” on page 22.

- **Port check**

Port checks control the network connections that a client can generate during a session. Use this rule to require a client machine to have certain ports open or closed in order for the user to access the IVE.

- **Process check**

Process checks control the software that a client may run during a session. Use this rule to require a client machine to be either running or not running a certain process in order for the user to access the IVE.

- **File check**

Use this rule to require a client machine to possess or not possess a certain file in order for the user to access the IVE. You may also use file checks to evaluate the age of required files and allow or deny access accordingly.

- **Registry settings check**

Registry settings checks control the corporate PC images, system configurations, and software settings that a client must have in order to access the IVE. Use this rule to require a client machine to have certain registry settings in order for the user to access the IVE. You may also use registry checks to evaluate the age of required files and allow or deny access accordingly.

You can select any number of methods and rules for Host Checker to use to verify that a client possesses the required endpoint properties. These rules are combined to create the policy that Host Checker verifies on the client. For configuration instructions, see “Security > Host Checker tab” on page 11.

You may also define Host Checker policies using the Host Check Server Integration Interface. Policies defined in this manner are recognized by the IVE when you upload the third-party integration package to the IVE. For more information, see “Host Check Server Integration Interface” on page 26.

Implementing Host Checker policies

Once you have created global policies, you can restrict IVE and resource access by requiring Host Checker in a:

- **Realm authentication policy**

When administrators or users try to sign in to the IVE, the IVE evaluates the specified realm’s authentication policy to determine if the pre-authentication requirements include Host Checker. You can configure a realm authentication policy to download Host Checker, download Host Checker and enforce Host Checker policies specified for the realm, or not require Host Checker. The user must sign in from a machine that adheres to the Host Checker requirements specified for the realm. If it does not, then the IVE does not forward the user’s credentials to the authentication server and the user is denied access to the IVE.

- **User role**

When the IVE determines the list of eligible roles to which an administrator or user may be mapped, it evaluates each role’s restrictions to determine if the role requires that the user’s machine adheres to certain Host Checker policies. If it does and the user’s machine does not follow the specified Host Checker policies, then the IVE does not map the user to that role.

- **Resource policy**

When a user requests a resource, the IVE evaluates the resource policy’s detailed rules to determine if the resource requires that the user’s machine adheres to certain Host Checker policies. The IVE denies access to the resource if the user’s machine does not follow the specified Host Checker policies.

You may specify that the IVE evaluate your Host Checker policies only when the user first tries to access the realm, role, or resource that references the Host Checker policy, or you may specify that the IVE periodically re-evaluate the policies throughout the user’s session. If you choose to periodically evaluate Host Checker policies, the IVE dynamically maps users to roles and allows users access to new resources based on the most recent evaluation. For more information, see “Executing Host Checker policies” on page 7.

Installing Host Checker

If you implement any policy at the realm, role, or resource policy level that requires Host Checker, you must provide a mechanism by which the IVE or the user can install Host Checker on the client machine. Otherwise, when the IVE evaluates the Host Checker policy, the user's machine will fail because the Host Checker client is not available to return a success status.

You use two methods to install Host Checker on a user's system:

- **The IVE automatically installs Host Checker**

Enable automatic installation through the **User/Administrator > Authentication Realm > Authentication Policy > Host Checker** page of the Web console. When you do, the IVE evaluates the realm-level option when the user accesses the IVE sign-in page and then determines if the current version of Host Checker is installed on the user's machine. If the Host Checker control is not installed, the IVE installs it.

- **The user or administrator manually installs Host Checker**

Download the Host Checker installer from the **Maintenance > System > Installers** page of the Web console and use it to manually install the Host Checker ActiveX control on to the user's system.

Note: Users need to enable ActiveX on their machines to enable Host Checker. By default, users must also have administrator or power user privileges. If the user does not have these privileges, use the Juniper Installer Service available from the **Maintenance > System > Installers** page of the Web console to bypass this requirement.

Executing Host Checker policies

When the user tries to access the IVE, Host Checker evaluates its policies in the following order:

1. Initial evaluation

When a user first tries to access the IVE sign-in page, Host Checker performs an initial evaluation.¹ Using the methods and rules you specify in your policies, Host Checker verifies that the client meets your endpoint requirements and returns its results to the IVE. Host Checker performs an initial evaluation regardless of whether you have implemented Host Checker policies at the realm, role, or resource policy level.

Note that the IVE waits 120 seconds for Host Checker to return a pass or fail result from the initial evaluation before timing out. If the IVE does not receive a result from Host Checker, it displays an error and directs the user back to the sign-in page. Otherwise, the IVE goes on to evaluate the realm level policies.

2. Realm-level policies

The IVE uses the results returned by Host Checker's initial evaluation to determine which realms the user may access. Then, the IVE displays or hides realms from the user, only allowing him to sign into those realms that are enabled for the sign-in page and whose Host Checker

1. If the user accesses the sign-in page and then closes the browser or does not sign in to the IVE, Host Checker continues to run on the user's machine. Host Checker will properly shut down the next time the user signs in to the IVE from the same machine.

requirements are met. If the user cannot meet the Host Checker conditions required by any of the available realms, the IVE does not display the sign-in page. Instead, it displays an error stating that the computer does not comply with the endpoint policy.

Note that Host Checker only performs realm-level checks when the user first signs into the IVE. If the state of the user's system changes during his session, the IVE does not remove him from the current realm or allow him access to a new realm based on his new system state.

3. Role-level policies

Once the user has signed into a realm, the IVE evaluates role-level policies and maps the user to the role or roles whose Host Checker requirements are met. Then, the IVE displays the IVE homepage to the user and enables those options that are allowed by the mapped role(s).

If Host Checker returns a different status during a periodic evaluation, the IVE dynamically remaps the user to roles based on the new results. If the end user loses rights to all available roles during one of the periodic evaluations, the IVE disconnects the user's session.

4. Resource-level policies

Once the IVE allows the user to access the homepage, the user may try to access a resource that is controlled by a resource policy. When he does, the IVE determines whether or not to perform the action specified in the resource policy based on the last status returned by Host Checker.

If Host Checker returns a different status during a periodic evaluation, the new status only impacts new resources that the user tries to access. For example, if successfully initiates a Network Connect session, and then fails his next resource-level host check, he may continue to access the open Network Connect session. The IVE only denies him access if he tries to open a new Network Connect session. The IVE checks the last status returned by Host Checker whenever the user tries to access a new Web resource or open a new Secure Application Manager, Network Connect, or Secure Terminal Access session.

With either a success or fail, Host Checker remains on the client in the C:\Program Files\Neoteris\Host Checker directory. Users may manually uninstall the agent by running `uninstal.l.exe` in this directory. This directory also contains a log file, which is rewritten each time Host Checker runs.

For information realm, role, and resource configuration instructions, see "Host Checker restrictions" on page 17.

Cache Cleaner overview

Cache Cleaner is a Windows client-side agent that removes residual data, such as temporary files or application caches, left on a user's machine after an IVE session. For example, when a user signs in to the IVE from an Internet kiosk and opens a Microsoft Word document using a browser plug-in, Cache Cleaner removes the temporary copy of the Word file stored in the browser cache (Windows folder) when the session terminates. By removing the copy, Cache Cleaner prevents other kiosk users from finding and opening the Word document after the IVE user concludes the session. You can also specify that Cache Cleaner clears content from specific hosts and domains, as well as specific files and folders.

You can restrict IVE and resource access by requiring Cache Cleaner in a:

- **Realm authentication policy**

When administrators or users try to sign in to the IVE, the IVE evaluates the specified realm's authentication policy to determine if the pre-authentication requirements include Cache Cleaner. You can configure a realm authentication policy to download Cache Cleaner, download and start running Cache Cleaner, or not require Cache Cleaner. The user must sign in from a machine that adheres to the Cache Cleaner requirements specified for the realm. If it does not, then the IVE does not forward the user's credentials to the authentication server and the user is denied access to the IVE.

- **User role**

When the IVE determines the list of eligible roles to which an administrator or user may be mapped, it evaluates each role's restrictions to determine if the role requires Cache Cleaner to be running on the user's workstation. If it does and the user's machine is not already running Cache Cleaner, then the IVE does not map the user to that role.

- **Resource policy**

When a user requests a resource, the IVE evaluates the resource policy's detailed rules to determine whether or not Cache Cleaner needs to be installed or running on the user's workstation. The IVE denies access to the resource if the user's machine does not meet the Cache Cleaner requirement.

For more information, see:

- "Cache Cleaner execution" on page 9
- "Security > Cache Cleaner tab" on page 15
- "Cache Cleaner restrictions" on page 18

Cache Cleaner execution

If you want to require Cache Cleaner in a role or resource policy, you need to minimally install the agent when the user signs in, which you configure in the realm's authentication policy. If configured, then the IVE downloads the ActiveX control to the user's system.

You may specify that the IVE evaluate your Cache Cleaner policies only when the user first tries to access the realm, role, or resource that references the Cache Cleaner policy. Or, you may use settings in the **System > Configuration > Security > Cache Cleaner** tab¹ to specify that the IVE periodically re-evaluate the policies throughout the user's session. If you choose to periodically evaluate Cache Cleaner policies, the IVE dynamically maps users to roles and allows users access to new resources based on the most recent evaluation. The IVE uses the same logic to perform periodic Cache Cleaner evaluations as it does to perform periodic Host Checker evaluations. For more information, see "Executing Host Checker policies" on page 7.

Note: Users need to enable ActiveX on their machines to enable Cache Cleaner to run.

1. Users with personal firewalls see a log entry every time Cache Cleaner clears the cache.

Cache Cleaner performs a final cleanup when:

- **The user explicitly signs out of the user session**

When a user clicks **Sign Out** on the IVE home page, Cache Cleaner performs a final cleanup and then uninstalls itself from the user's system.

- **The user session times out**

When a user session times out, Cache Cleaner performs a cleanup, and then if user signs in again, Cache Cleaner performs another cleanup. Cache Cleaner is aware of session timeouts, because it periodically checks the validity of a session at the interval you specify on the **System > Configuration > Security > Cache Cleaner** tab (see "Security > Cache Cleaner tab" on page 15).

Note: When checking the validity of a user session, Cache Cleaner connects to the IVE. This action may trigger warnings on personal firewalls. Users must permit this traffic to ensure that Cache Cleaner functions correctly.

- **A client system restarts after an abnormal termination**

If Cache Cleaner terminates abnormally due to a system, session, or network connection problem, Cache Cleaner performs a final cleanup and uninstalls itself from the user's system after the system restarts. Note that Cache Cleaner cannot log data after it terminates.

Cache Cleaner does not log entries to the standard IVE log, but if you enable client-side logging through the **System > Configuration > Security > Client-side Logs** page, Cache Cleaner does log data to a temporary client-side text file:

```
c:\Program Files\Neoteris\Cache Cleaner\dsCacheCleaner.log
```

This encrypted log is deleted when Cache Cleaner uninstalls itself.

Important: If you configure Cache Cleaner to remove files from a directory, Cache Cleaner clears all files, including those that the user has explicitly saved to the directory and files that were in the directory prior to the IVE session.

Configuring endpoint defense options

The following sections include instructions for configuring endpoint defense options on your IVE appliance:

Security > Host Checker tab.....	11
Security > Cache Cleaner tab	15
Host Checker restrictions	17
Cache Cleaner restrictions	18
Installers tab	19

Security > Host Checker tab

Use the **System > Configuration > Security > Host Checker** tab to:

Specify global Host Checker options.....	11
Create a global client-side policy.....	12
Create a global server-side policy	14
Download the Host Checker installer	15

Specify global Host Checker options

You can specify global options for Host Checker that apply to any user for whom Host Checker is required in an authentication policy, a role mapping rule, or a resource policy.

To specify global Host Checker options:

1. In the Web console, choose **System > Configuration > Security > Host Checker**.
2. Under **Options**:
 - In the **Perform check every X minutes** field, specify the interval at which you want Host Checker to run on a client machine. If the client machine fails to meet the requirements of the Host Checker policies required by a role or resource policy, then the IVE denies the associated user requests.

For example, you may require that a user runs a specific third-party anti-virus application in order to map to Role A, which enables network connections from an external location. If the user's client machine is running the required anti-virus application when the user signs in to the IVE, then the user maps to Role A and is granted all access features enabled for Role A. If the anti-virus application stops running during the user session, however, the next time Host Checker runs, the user fails to meet the security requirements for Role A and

therefore loses all access privileges for Role A.

Important: If you enter a value of zero, Host Checker only runs on the client machine when the user first signs into the IVE.

- Check **Auto-upgrade Host Checker** if you want the IVE to automatically download the Host Checker application to a client machine when the version of Host Checker on the IVE is newer than the version installed on the client. If you select this option, note the following:
 - The user must have Administrator privileges in order for the IVE to automatically install the Host Checker application on the client.
 - If a user uninstalls Host Checker and then signs in to an IVE for which the **Auto-upgrade Host Checker** option is not enabled, the user no longer has access to Host Checker.

3. Click **Save Changes**.

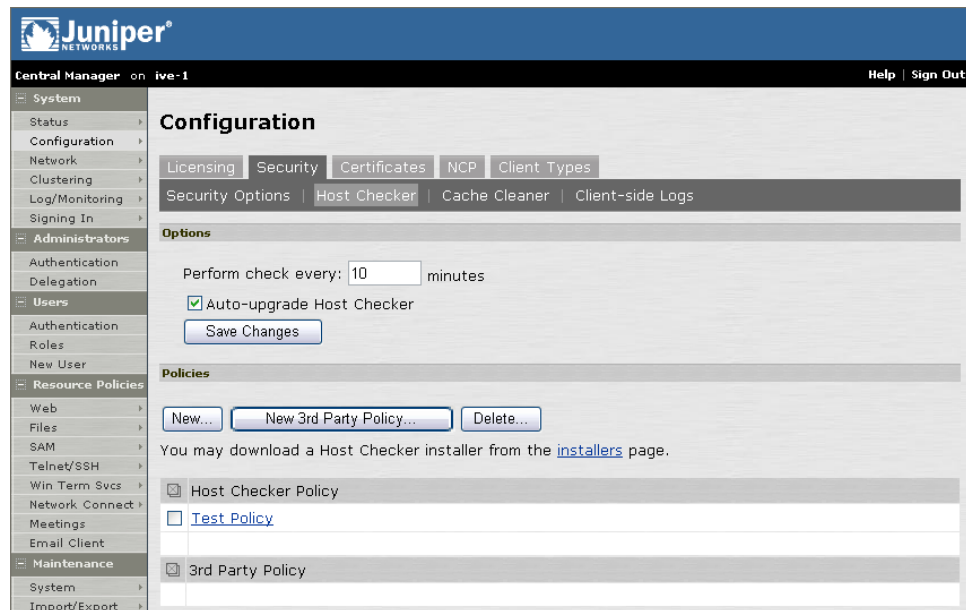


Figure 1: System > Configuration > Security > Host Checker

Create a global client-side policy

You can create global Host Checker policies that ensure that specified client-side processes, files, registry entries, ports, or integrated third-party endpoint security products conform to your specifications. Once you create these policies, you can then invoke them at the realm, role, and resource levels.

To create a global Host Checker policy:

1. In the Web console, choose **System > Configuration > Security > Host Checker**.
2. Under **Policies**, click **New**.

3. On the **Configuration** page, enter a name in the **Policy Name** field and then click **Continue**.
4. Under **Host Checking Method**, select any number of the following options (optional):
 - **Sygate Enforcement API** — requires that a Sygate Personal Firewall product is installed on the client machine.
 - **Sygate Security Agent** — requires that Sygate Security Agent is installed on the client machine.
 - **Zone Labs: Zone Alarm Pro and Zone Labs Integrity** — requires that either Zone Alarm Pro or Zone Labs Integrity is installed on the client machine.
 - **McAfee Desktop Firewall 8.0** — requires that McAfee Desktop Firewall 8.0 is installed on the client machine.
 - **InfoExpress CyberGatekeeper Agent** — requires that InfoExpress CyberGatekeeper Agent is installed on the client machine.
5. Under **Rule Settings**, select a type of rule (see page 5 for descriptions) from the drop-down list and then click **Add** (optional). The configuration dialog for the rule appears. In the configuration dialog for:
 - **3rd Party NHC Check:**
 - 1 Enter the name for the DLL.
 - 2 Enter the location of the DLL on client machines (path and file name).
 - 3 Click **Save Changes**.
 - **Attribute Check: Ports:**
 - 1 Enter a comma delimited list (without spaces) of ports or port ranges, such as: 1234, 11000-11999, 1235.
 - 2 Select **Required** to require that these ports are open on the client machine or **Deny** to require that they are closed.
 - 3 Click **Save Changes**.
 - **Attribute Check: Process:**
 - 1 Enter the name of a process (executable file), such as: good-app. exe.
 - 2 Select **Required** to require that this process is running in the Task Manager or **Deny** to require that this process is not running.
 - 3 Specify the MD5 Checksum value of each executable file to which you want the policy to apply (optional). For example, an executable may have different MD5 checksum values on a desktop, laptop, or different Windows OS versions. Specify each valid value.
 - 4 Click **Save Changes**.
 - **Attribute Check: File:**
 - 1 Enter the name of a file (any file type), such as: \Temp\Bad-file.doc.

Important: You cannot include variables as part of the file path.

- 2 Select **Required** to require that this file is present on the client machine or **Deny** to require that this file is not present.
- 3 Specify the maximum age (in days) for a file (optional). If the file is older than the specified number of days, then the client does not meet the attribute check requirement.

Tip:

Use this option to check the age of virus signatures. Make sure to specify the path to a file (in the **File Name** field) whose timestamp indicates when virus signatures were last updated, such as a virus signature database or log file that is updated each time the database is updated. For example, if you use TrendMicro, you may specify: C:\Program Files\Trend Micro\OfficeScan Client\TmUpdate.ini.

- 4 Specify the MD5 Checksum value of each executable file to which you want the policy to apply (optional).
- 5 Click **Save Changes**.
- **Attribute Check: Registry Setting:**
 - 1 Select a root key from the drop-down list.
 - 2 Enter the path to the application folder for the registry subkey.
 - 3 Enter the name of the key's value that you want to require (optional). This name appears in the **Name** column of the Registry Editor.
 - 4 Select the key value's type (string, binary, or dword) from the drop-down list (optional). This type appears in the **Type** column of the Registry Editor.
 - 5 Specify the required registry key value (optional). This information appears in the **Data** column of the Registry Editor.

If the key value represents an application version, select the **Minimum version** checkbox to allow the specified version or newer versions of the application. The IVE uses lexical sorting to determine if the client contains the specified version or higher. For example:

- 3.3.3 is newer than 3.3
- 4.0 is newer than 3.3
- 4.0a is newer than 4.0b
- 4.1 is newer than 3.3.1

Tip:

Use this option to specify version information for an antivirus application to make sure that client antivirus software is current.

- 6 Click **Save Changes**.

Note: If you specify only the Key and Subkey, Host Checker simply verifies the existence of the Subkey folder in the registry.

- 7 Repeat this process to add another rule to the Host Checker policy. When you are finished adding rules, click **Save Changes**. The IVE adds the policy to the Host Checker **Configuration** page.

Create a global server-side policy

You can create global Host Checker policies that take software that you have uploaded to the IVE and run it on client machines. Once you create these policies, you can then invoke them at the realm, role, and resource levels. For more information, see "Host Check Server Integration Interface" on page 26.

To create a global Host Checker policy:

1. In the Web console, choose **System > Configuration > Security > Host Checker**.

2. Under **Policies**, click **New 3rd Party Policy**.
3. Enter a name to identify your zip file on the IVE.
4. Browse to the local directory where your zip file is located.
5. Click **Save Changes**. The IVE adds the policies defined in your zip file to the Host Checker **Configuration** page.

Download the Host Checker installer

To download the Host Checker application as a Windows executable file, go to **Maintenance > System > Installers**. For more information about downloading Host Checker, see "Download an application or service" on page 19.

Security > Cache Cleaner tab

Use the **System > Configuration > Security > Cache Cleaner** tab to specify how often Cache Cleaner runs and updates the IVE of its status, as well as specific browser cache and directory data to clear. See "Cache Cleaner overview" on page 8 for more information about this feature.

Specify global Cache Cleaner settings

To specify global Cache Cleaner settings:

1. In the Web console, choose **System > Configuration > Security > Cache Cleaner**.
2. At the top of the page:
 1. In the **Cleaner Frequency** field, specify how often Cache Cleaner runs. Valid values range from 1-60 minutes. Each time Cache Cleaner runs, it clears the browser cache, files, and folders you specify in the **Browser Cache** and **Files and Folders** sections below.
 2. In the **Status Update Frequency** field, specify how often the IVE expects Cache Cleaner to update itself. Valid values range from 1-60 minutes.
 3. Select the **Uninstall Cache Cleaner at logout** checkbox if you want the IVE to uninstall Cache Cleaner from the client machine when a user's session ends (optional).
3. Under **Browser Cache**, enter one or more hostnames or domains (wildcards are permitted). When a user session ends, Cache Cleaner removes any content in the browser cache that originates from these servers. Cache Cleaner also removes this content when it runs at the specified cleaner frequency interval.

Note: The IVE does not resolve hostnames, so enter all possible representations of a server, such as its hostname, FQDN, and IP address.

4. Under **Files and Folders**:

1 Specify either:

- the name of a file that you want Cache Cleaner to remove or
- the complete directory path to a folder whose contents you want Cache Cleaner to remove. If you specify a directory, select **Clear Subfolders** to also clear the contents of any subdirectories within this directory.

2 Select the **Clear folders only at the end of session** checkbox if you want Cache Cleaner to clear directory contents only at the end of the user session. Otherwise, Cache Cleaner also clears files and folders at the specified cleaner frequency interval.

5. Click **Save Changes** to save these settings globally.

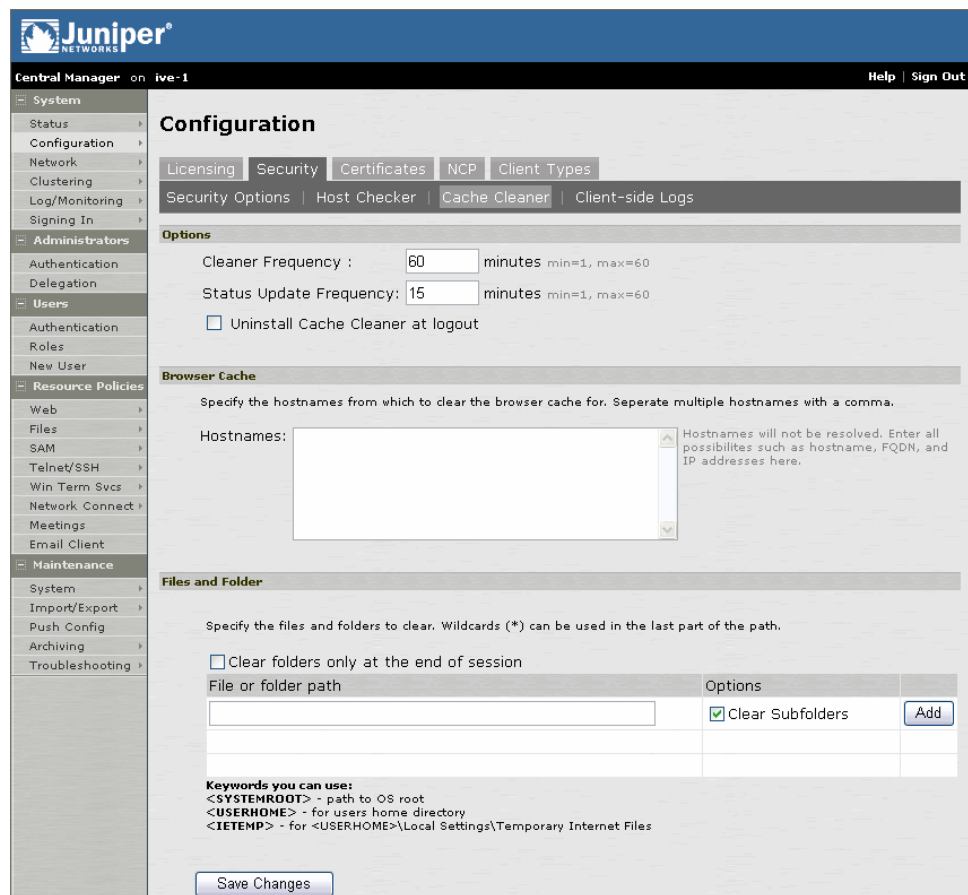


Figure 2: System > Configuration > Security > Cache Cleaner

Host Checker restrictions

Use a Host Checker restriction to require client machines to meet the specified Host Checker policies in order to access an IVE sign-in page, be mapped to a role, or access a resource policy.

Specify Host Checker restrictions

To specify Host Checker restrictions at the:

- **System level**

Navigate to: System > Configuration > Security > Host Checker

Specify global options for Host Checker to apply to any user for whom Host Checker is required in an authentication policy, a role mapping rule, or a resource policy. For more information, see “Security > Host Checker tab” on page 11.

- **Realm level**

Navigate to:

- Administrators > Authentication > *SelectRealm* > Authentication Policy > Host Checker
- Users > Authentication > *SelectRealm* > Authentication Policy > Host Checker

- **Role level**

- Administrators > Delegation > *SelectRole* > General > Restrictions > Host Checker
- Users > Authentication > *SelectRealm* > Role Mapping > *Select/CreateRule* > *CustomExpression*
- Users > Roles > *SelectRole* > General > Restrictions > Host Checker

- **Resource policy level**

Navigate to:

Resource Policies > *SelectResource* > *SelectPolicy* > Detailed Rules > *Select/CreateRule* > *ConditionField*

Then:

Choose one of the following options if you are configuring a **realm level** Host Checker requirement:

- **Allow all users** — Does not require Host Checker to be installed in order for the user to meet the access requirement.
- **Allow all users & install Host Checker** — Requires the IVE to download Host Checker to the client machine. If you choose this option for a realm’s authentication policy, then the IVE downloads Host Checker to the client machine after the user is authenticated and before the user is mapped to any roles in the system.
- **Allow only users whose workstations meet the requirements specified by the following [policies]** — Requires that Host Checker is running the specified Host Checker policies in order for the user to meet the access requirement. If you choose this option for a realm’s authentication policy, then the IVE downloads Host Checker to the client machine before the user may access the IVE sign-in page.

Choose one of the following options if you are configuring a **role level** Host Checker requirement:

- **Allow all users** — Does not require Host Checker to be installed in order for the user to meet the access requirement.
- **Allow only users whose workstations meet the requirements specified by the following [policies]** — Requires that Host Checker is running the specified Host Checker policies in order for the user to meet the access requirement.

Alternatively, you can write a custom expression for the role mapping rule to evaluate Host Checker's status using the `hostCheckerPolicy` variable. If you want to create a Host Checker restriction at the **resource policy level**, you must create a custom expression in a detailed rule.

Cache Cleaner restrictions

Use a Cache Cleaner restriction to require client machines to meet the specified Cache Cleaner requirement in order to access an IVE sign-in page, be mapped to a role, or access a resource policy.

Specify Cache Cleaner restrictions

To specify Cache Cleaner restrictions at the:

- **System level**

Navigate to: System > Configuration > Security > Cache Cleaner

Specify global options for Cache Cleaner to apply to any user for whom Cache Cleaner is required in an authentication policy, a role mapping rule, or a resource policy. For more information, see "Security > Cache Cleaner tab" on page 15.

- **Realm level**

Navigate to: Users > Authentication > *SelectRealm* > Authentication Policy > Cache Cleaner

- **Role level**

- Administrators > Delegation > *SelectRole* > General > Restrictions > Cache Cleaner
- Users > Authentication > *SelectRealm* > Role Mapping > *Select/CreateRule* > *CustomExpression*
- Users > Roles > *SelectRole* > General > Restrictions > Cache Cleaner

- **Resource policy level**

Navigate to:

Resource Policies > *SelectResource* > *SelectPolicy* > Detailed Rules > *Select/CreateRule* > *ConditionField*

Then:

Choose one of the following options if you are configuring a **realm level**

Cache Cleaner requirement:

- **Disable Cache Cleaner** — Does not require Cache Cleaner to be installed or running in order for the user to meet the access requirement.
- **Just load Cache Cleaner** — Does not require Cache Cleaner to be running in order for the user to meet the access requirement but ensures that it is available for future use. If you choose this option for a realm's authentication policy, then the IVE downloads Cache Cleaner to the client machine after the user is authenticated and before the user is mapped to any roles on the system.
- **Load and enforce Cache** — Requires the IVE to download and run Cache Cleaner in order for the user to meet the access requirement. If you choose this option for a realm's authentication policy, then the IVE downloads Cache Cleaner to the client machine before the user may access the IVE sign-in page.

Check the following option if you are configuring a *role level* Cache Cleaner requirement:

- **Enable Cache Cleaner** — Requires Cache Cleaner to be running in order for the user to meet the access requirement.

Alternatively, you can write a custom expression for the role mapping rule to evaluate Cache Cleaner's status using the `cacheCleaner` variable. If you want to create a Cache Cleaner restriction at the *resource policy level*, you must create a custom expression in a detailed rule.

Installers tab

The Installers tab provides several applications and a service for download. You can download an application or service as a Windows executable file, which enables you to:

- Distribute the file to client machines using software distribution tools. This option enables you to install an application or service on client machines whose users do not have Administrator privileges, which are required to install the application or service.
- Post the executable in a secure repository so that users with the proper administrator right may download and install the appropriate version.

These options allow you to control which version of an application or service runs on client machines.

Download an application or service

- **Juniper Installer Service** — The Juniper Installer Service allows users to download, install, upgrade, and run client-side applications without administrator privileges.

- **Host Checker** — Host Checker is a client-side agent that performs endpoint security checks on hosts that connect to the IVE.

Important: If you decide to distribute Host Checker, make sure to uncheck the **Auto-upgrade Host Checker** option on the **System > Configuration > Security > Host Checker** page (see “Specify global Host Checker options” on page 11) otherwise the IVE downloads the Host Checker application to a user’s machine, which may not be the same version as the distributed version.

To download an application or service:

1. In the Web console, choose **Maintenance > System > Installers**.
2. Click on the **Download** link to the right of the application or service you want to download. The **File Download** dialog box appears.
3. Click the **Save** button on the **File Download** dialog box. The **Save As** dialog box appears.
4. Choose an appropriate location in the **Save As** dialog box.
5. Click the **Save** button on the **Save As** dialog box.

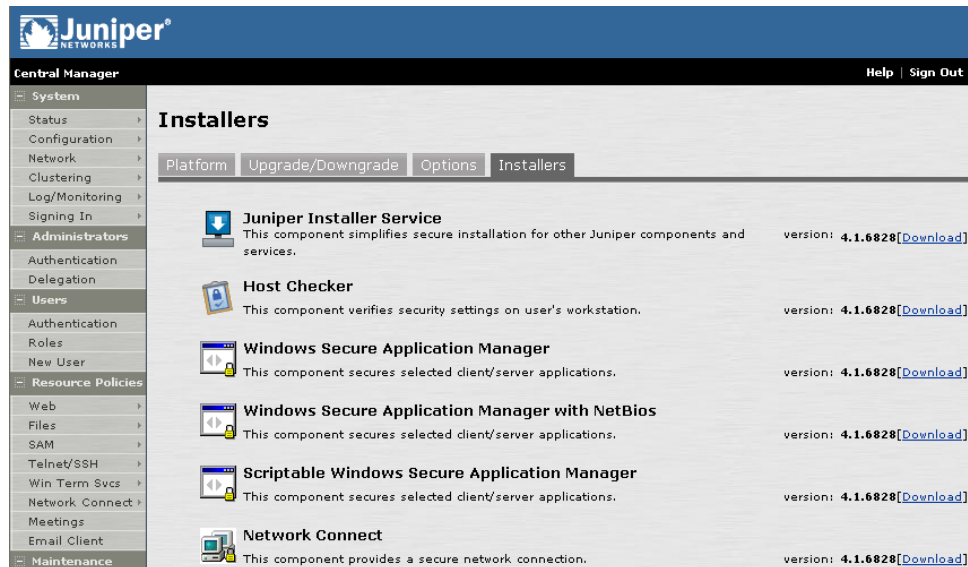


Figure 3: Maintenance > System > Installers

Host Checker interfaces

An IVE appliance provides two different APIs that you may use to integrate third party functionality:

- **Host Check Client Interface**

The Host Check Client Interface is an API that allows you to run your own DLLs using Host Checker. Through the interface, you can prompt Host Checker to run a DLL that you have already installed on the user's system or distributed as part of a corporate OS image, including programs that check compliance with corporate images, antivirus software, and personal firewall clients. Host Checker runs the specified DLL when a user signs into the IVE, and then bases its subsequent actions on the success or failure result returned by your DLL. For example, you may deny a user access to the IVE appliance if the client check software fails. For more information, see "Host Check Client Interface" on page 22.

- **Host Check Server Integration Interface**

The Host Check Server Integration Interface is an API that allows you to tightly integrate your own DLLs and corresponding files into the IVE appliance. Like the Host Check Client Interface, you can use the Host Check Server Integration Interface to prompt Host Checker to run your software on the client, including host integrity scans, malware detectors, and virtual environments. With this interface, you may also specify with extreme granularity what Host Checker should do based on the result returned by the DLL, including mapping users to different realms, roles, and resource policies based on the results of individual policies contained in your software package. For more information, see "Host Check Server Integration Interface" on page 26.

Host Check Client Interface

The Host Check Client Interface involves communicating with a third-party endpoint security application through its API and examining the return values to verify the trustworthiness of the client machine. Through the Host Check Client Interface, the IVE Host Checker feature currently supports tight integration with Sygate Enforcement API, Sygate Security Agent, Zone Labs ZoneAlarm Pro, Zone Labs Integrity, McAfee Desktop Firewall 8.0, and InfoExpress CyberGatekeeper Agent. To support other endpoint security applications or those that do not have an API, the IVE platform provides a generic API library in the C programming language. This Windows API is called the Host Check Client Interface API and contains the `NHC_EndpointSecure()` function, which checks the endpoint configuration.

Host Check Client Interface integration typically includes these steps:

1. An IVE administrator enables Host Checker for the desired realm, role, or resource. For this realm, role, or resource, the administrator specifies a 3rd party NHC check rule on the Web Console's Host Checker page. This rule specifies the location of your custom DLL on a client machine.
2. The IVE appliance downloads an ActiveX installer with the Host Check Client Interface package to the client machine of an authenticated user who is trying to access the realm, role, or resource.
3. The Host Check Client Interface package loads your custom DLL from the DLL location on the client. Before calling the `NHC_EndpointSecure()` function, the package calls the Windows `WinVerifyTrust()` function to validate the DLL's digital signature. (See "Signing your custom DLL" on page 22 for information about the user experience.)
4. The Host Check Client Interface package calls the `NHC_EndpointSecure()` function. If the function returns `NHC_STATUS_SECURE`, the third-party product endpoint security check succeeds and the IVE appliance maps the user to the realm, role, or resource. If the endpoint security check fails for a realm-level policy, the user sees an error stating that the computer does not comply with the endpoint security policy, and the user is redirected to the sign-in page. If you specify the URL to a failure page, this page opens in another browser window. If the endpoint security check fails for a role or resource level policy, the IVE appliance simply does not map the user to the role or resource.

Signing your custom DLL

We strongly recommend that you digitally sign your custom DLL to ensure content integrity. Prior to calling your DLL, Host Checker calls the Windows `WinVerifyTrust()` function to attempt to verify the trustworthiness of the DLL.

- If your DLL is not digitally signed and the user's browser security settings are Medium-to-High, the user is informed that the DLL is not signed and cannot be verified as trustworthy. The user may choose to proceed, in which case Host Checker calls the DLL. If the `NHC_EndpointSecure()` function returns `NHC_STATUS_SECURE`, the user is mapped to the realm, role, or resource. If the user chooses to cancel the operation, the IVE does not map the user to the realm, role, or resource.

- If the DLL is digitally signed but `WinVerifyTrust()` cannot verify the trustworthiness of the DLL, the user is informed that the DLL cannot be verified as trustworthy. The user may choose to proceed, in which case the Host Checker calls the `NHC_EndpointSecure()` function. If this function returns `NHC_STATUS_SECURE`, the user is mapped to the realm, role, or resource. If the user chooses to cancel the operation, the IVE appliance does not map the user to the realm, role, or resource.
- If the DLL is digitally signed and `WinVerifyTrust()` verifies the trustworthiness of the DLL, the user is notified that the content provider and its integrity have been verified and asked if he wishes to proceed. If the user proceeds, Host Checker calls `NHC_EndpointSecure()`. If this function returns `NHC_STATUS_SECURE`, the user is mapped to the realm, role, or resource.

Deploying and maintaining your custom DLL

To deploy your custom DLL, you need to:

- At the system level, configure the Host Checker feature to call your custom DLL and specify the path (including the file name) to where the DLL is stored on client machines.
- Install the DLL on the appropriate client machines or distribute the DLL as part of the corporate PC image.
- Create a low-privilege realm or role that users can access if the endpoint security check fails and make the DLL available on a page configured as an IVE bookmark for those users.
- Check the DLL timestamp to ensure that the version is current. If the endpoint security check fails, redirect users to a "safety page" that enables them to authenticate and download the current DLL.

NetScreen Host Checker (NHC) API

This section contains function definitions for the NHC API.

NHC API Definitions

```
#define NHC_STATUS_SECURE          1
#define NHC_STATUS_SUCCESS        0
#define NHC_STATUS_FAILURE        -1
#define NHC_STATUS_UNSECURE       -2
#define NHC_STATUS_BADPARAMETER   -3
```

NHC_EndpointSecure() Required

The NHC_EndpointSecure function verifies the security of the endpoint based on the criteria established by the implementer.

Syntax:

```
NHC_API int WINAPI NHC_EndpointSecure(void);
```

Parameters:

None

Return Values:

- NHC_STATUS_SECURE
The endpoint client is secure
- NHC_STATUS_UNSECURE
The endpoint client is not secure
- NHC_STATUS_FAILURE
The endpoint application is not running
- NHC_STATUS_BADPARAMETER
An internal error has occurred; the endpoint client should be judged insecure

C Header file: neoterisGenericAPI.h

Include this header file in your DLL:

```
/*
neoterisGenericAPI.h:
This header file defines the generic API for integrating with Host
Checker
*/
#ifndef NEOTERISGENERICAPI_H
#define NEOTERISGENERICAPI_H
#ifdef __cplusplus
extern "C" {
#endif
```

```

#ifdef NHC_EXPORTS

#define NHC_API __declspec(dllexport)

    #else

#define NHC_API __declspec(dllimport)

#endif

#define NHC_STATUS_SECURE          1
#define NHC_STATUS_SUCCESS        0
#define NHC_STATUS_FAILURE        -1
#define NHC_STATUS_UNSECURE       -2
#define NHC_STATUS_BADPARAMETER   -3

NHC_API int WINAPI NHC_EndpointSecure(void);

/*
    Parameters: None

    Return Values:

    NHC_STATUS_SECURE if the endpoint client is secure

    NHC_STATUS_UNSECURE if the endpoint client is not secure

    NHC_STATUS_FAILURE if the endpoint application is not
    running

    NHC_STATUS_BADPARAMETER if an internal error has occurred;
    the endpoint client should be judged insecure

*/

#ifdef __cplusplus
}

#endif

#endif //NEOTERISGENERI CAPI_H

```

Host Check Server Integration Interface

The Host Check Server Integration Interface contains a generic API library in the C++ programming language. This section contains the describes the API and how to implement it for use with Host Checker.

Deploying third party applications through Host Checker

To deploy third party applications through Host Checker, you must:

1. **Create an endpoint security package** (26)
2. **Upload the package through the IVE** (27)
3. **Configure Host Checker to use the package** (27)

Create an endpoint security package

An endpoint security package is a collection of files that comprise your third party functionality. When creating a package, you must include the following types of files:

- **Interface DLL**

An interface DLL is a program that carries out your specialized functions on the client. When creating your DLL, you must provide an interface between your modules and Host Checker using functions defined in the Host Check Server Integration Interface.

- **Package definition file**

A package definition file defines the name of your interface DLL as well as Host Checker policies that you have defined in your DLL. Within the file, you must include one definition per line using the following format:

```
HCI F-Mai n : <DLLName>
```

```
HCI F-Pol i cy : <Pol i cyName>
```

Note that if you do not include policies in your package, Host Checker simply enforces that the package has run on the client. If you do declare policies through this file, they become available through the IVE Web console where you can implement them at the realm, role, and resource policy levels.

In order for the IVE to recognize your policy definition file, you must name it MANI FEST.HCI F and include it a folder named META-INF.

- **Host Check Server Integration Interface header file**

The Host Check Server Integration Interface header file (hci f.h) defines the Host Check Server Integration Interface functions that you must use in your DLL. You must add this file to your package's i ncl ude folder in order to use the functions that are provided with the Host Check Server Integration Interface. You can find a copy of this file in the SDK provided with the product.

In addition to these required files, you may also include your own data files in the package.

Upload the package through the IVE

Once you have created your endpoint security package, you must archive it in a zip file and upload it to the IVE through the **System > Configuration > Security > Host Checker** page of the Web console. Note that once you upload a package to the IVE, you cannot modify it on the server. Instead, you must modify it on your local system, delete the server version, and upload the modified version to the IVE.

Configure Host Checker to use the package

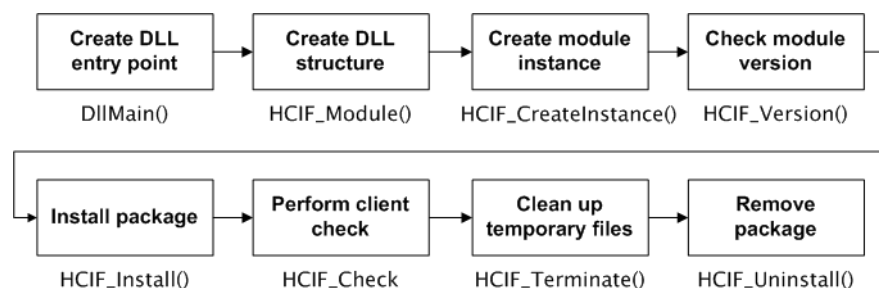
Once you have uploaded a valid package, the IVE automatically exposes the policies contained in the package in the realm, role, and resource policy configuration pages of the Web console. You may then use these pages to implement the policies. When a user tries to access a realm, role, or resource protected by one of these policies, the IVE runs the program that you have uploaded to the server.

Creating your interface DLL

When you create an interface DLL to carry out your specialized functions on user machines, you must use the functions provided in the Host Check Server Integration Interface to:

1. Provide an entry point for the DLL (28)
2. Create a module structure for your DLL (28)
3. Create an instance for your module (28)
4. Check the API version used to compile your module (28)
5. Install your endpoint security package on the user's machine (28)
6. Perform endpoint security checks (28)
7. Stop the DLL and clean up temporary files (28)
8. Remove your package from the user's machine (29)

The required steps and corresponding functions are illustrated in the following diagram:



Note:

- All of the functions described here are included in the hci f. cpp sample file provided with the SDK.
- You can test whether your package properly conforms to the standards described here using the hci ftool . exe tool provided with the SDK.

DllMain() function

Use the `DllMain()` function to define an entry point for your DLL within the main Host Checker framework. This function alerts Host Checker that it needs to execute your third party DLL. When adding the `DllMain()` function to your project, copy and use the exact version that is in the `hcf.cpp` sample that is included with the SDK.

HCIF_Module() function

Use the `HCIF_Module()` function to create a structure for your DLL. You should call all of the other functions described in this section from within the `HCIF_Module()` function.

HCIF_CreateInstance() function

Use the `HCIF_CreateInstance()` function to create an individual instance of your service and to create a DLL entry point. The Host Checker framework calls the `HCIF_CreateInstance()` function to obtain a pointer to the `HCIF_Module` structure.

HCIF_Version() function

Use the `HCIF_Version()` function to return the Host Checker API version that was used to compile the module. The Host Checker framework uses the return value (which is stored in the `HCIF_API_Version` macro) to compare the module version with the server version.

HCIF_Install() function

Use the `HCIF_Install()` function to deliver required files from the IVE to the user's system. Within the `HCIF_Install()` function, you may use the standard C `getFile` function to retrieve files that you have uploaded to the IVE and copy them to Host Checker's default directory on the user's system: `<Directory>\Neoteris\Host Checker` (where `<Directory>` is the location of your applications, such as: `C:\Program Files`).

HCIF_Check() function

Use the `HCIF_Check()` function to perform your client-side checks. You may use the `HCIF_Check()` function to run the core execution modules in your package. These modules should check the compliance of the endpoint with the configured policies and return a value of `TRUE` if the check was successful or `FALSE` if the check failed.

HCIF_Terminate() function

Use the `HCIF_Terminate()` function to perform any final clean up at the end of a session, including freeing resources, removing temporary files, or reverting registry entries. The Host Checker framework calls the `HCIF_Terminate()` function at the end of the user's IVE session.

HCIF_Uninstall() function

Use the `HCIF_Uninstall()` function to remove all traces of your module from the user's system. The Host Checker framework calls the `HCIF_Uninstall()` function after the `HCIF_Terminate()` function.