



Writing a Custom DLL to Use with the IVE Host Checker

The IVE Host Checker performs endpoint security checks on hosts that connect to the IVE. The Host Checker may use either *attribute checking*, in which it looks for specified application and system settings, or *NHC integration*, in which it calls either the NHC implementation of a supported endpoint security application or a custom DLL that implements NHC. This document describes the API and how to implement it for use with the IVE Host Checker. For more information about deploying the Host Checker with supported endpoint security applications or using attribute checking, please refer to the *Neoteris Instant Virtual Extranet Administration Guide* PDF.

| | |
|---|----------|
| NHC API Integration | 1 |
| Signing your custom DLL..... | 2 |
| Deploying and maintaining your custom DLL | 2 |
| Neoteris Host Checker (NHC) API | 3 |
| NHC API Definitions | 3 |
| NHC_EndpointSecure() | 3 |
| C Header file: neoterisGenericAPI.h | 3 |

NHC API Integration

NHC integration 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 NHC integration, the IVE Host Checker 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 provides a generic API library in the C programming language. This Windows¹ API is called the Neoteris Host Checker (NHC) API and contains the `NHC_EndpointSecure()` function², which checks the endpoint configuration.

NHC integration typically includes these steps:

1. An IVE administrator enables the Host Checker for the desired group. For this group, the administrator specifies an NHC integration rule on the Administrator Console's Host Checker page. This rule specifies the location of your custom DLL on a client machine.
2. The IVE downloads an ActiveX installer with the NHC package to the client machine of an authenticated user belonging to the group.
3. The NHC package loads your custom DLL from the DLL location on the client. Before calling the `NHC_EndpointSecure()` function, the NHC package calls the Windows `WinVerifyTrust()`

¹ A future release will provide support for non-Windows platforms.

² A future release will support additional functions to enable Host Checker policies to be more expressive.



function³ to validate the DLL’s digital signature. (See “Signing your Custom DLL” for information about the user experience.)

4. The NHC package calls the `NHC_EndpointSecure()` function. If the function returns `NHC_STATUS_SECURE`, the third-party product endpoint security check succeeds and the IVE presents the home page to the authenticated user. If the endpoint security check fails, 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.

Signing your custom DLL

We strongly recommend that you digitally sign your custom DLL to ensure content integrity. Prior to calling your DLL, the 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 the Host Checker calls the DLL. If the `NHC_EndpointSecure()` function returns `NHC_STATUS_SECURE`, the user is signed in to the IVE. If the user chooses to cancel the operation, the IVE does not sign in the user.
- 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 signed in to the IVE. If the user chooses to cancel the operation, the IVE does not sign in the user.
- 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, the Host Checker calls `NHC_EndpointSecure()`. If this function returns `NHC_STATUS_SECURE`, the user is signed in to the IVE.

Deploying and maintaining your custom DLL

To deploy your custom DLL, you need to:

- For the desired IVE authorization groups, 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 group that users can select if the endpoint security check fails and make the DLL available on a page configured as an IVE bookmark for that group.
- 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.

³ Please consult <http://msdn.microsoft.com/library> for information about this function.



Neoteris 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:

| | |
|--------------------------------------|---|
| <code>NHC_STATUS_SECURE</code> | The endpoint client is secure |
| <code>NHC_STATUS_UNSECURE</code> | The endpoint client is not secure |
| <code>NHC_STATUS_FAILURE</code> | The endpoint application is not running |
| <code>NHC_STATUS_BADPARAMETER</code> | 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
Neoteris IVE 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 //NEOTERISGENERICAPI_H
```