

What is Microsoft's Network Access Protection?

The most significant differences between Microsoft's Network Access Protection architecture and other NAC architectures you see in the iLabs come because Microsoft does not make switches or routers. Therefore, the path for handling enforcement is different, focusing on server enforcement and standards-based switch enforcement. The original intent of MS-NAP was not security, but to find and quarantine non-compliant clients in the enterprise LAN. As the interest in NAC has increased, Microsoft has adjusted their architecture to include more enforcement mechanisms, and it's the 802.1x portion of MS-NAP that we tested for interoperability in the iLabs.

It's important to realize that MS-NAP is part of Microsoft's Vista/Longhorn product. While Vista has been released, Longhorn is still in beta. This means that any MS-NAP testing you do is complicated by the client and server platforms and dealing with pre-release code. Microsoft has stated that it will back-port NAP to Windows XP as they have done with other security technologies bound to "new" versions of Windows. (The 802.1X supplicant and IPsec client are good examples of this). But if this happens, it likely won't be until Longhorn (the server platform) ships.

When reading this white paper, you may find it helpful to have at hand our companion white paper, "Network Access Control Architecture Alphabet Soup," with the diagram showing the different parts of a NAC architecture.

Access Requestor in MS-NAP

Following the common NAC architecture, the Microsoft client side is broken into three parts. At the top are the System Health Agents, taking on the function of collecting end-point security information about the client, such as the state of the anti-virus software or whether the firewall has the right policy. Microsoft has provided a System Health Validator, and many 3rd parties have declared their intent to provide System Health Validators of their own. These agents are responsible for generating Statements of Health that can be used to assess end-point security. Tying the System Health Agents into the rest of the architecture is Microsoft's Network Access Protection Agent, analogous to the IETF's Client Broker component. Below the Network Access Protection Agent are Microsoft's Enforcement Clients, which match up to the Network Access Requestor. MS-NAP includes 802.1X supplicant and VPN enforcement clients as typically found in other architectures, but also includes DHCP clients as an enforcement option.

More importantly, though, is that Microsoft has defined the API connecting its three layers of Network Access Protection on the client. By creating an API that describes how the three pieces of the client will fit together, Microsoft eliminates an enormous amount of risk and variability in the entire Network Access Control space. The Microsoft API provides a defined method for third party vendors to integrate their products into the MS-NAP solution. Even if Microsoft's entire Network Access Protection product plans were jettisoned internally, the contribution of having these defined APIs shipping with Windows cannot be underestimated.

Of course, the trick will be convincing every other NAC architect in the industry that Microsoft's API is both necessary to a good NAC design and sufficient for the task. No vendor is proposing to make this middleware piece a moneymaking differentiator. It simply exists to let desktop security vendors have a way of communicating the status of their products back to the Policy Decision Points. By simply adopting Microsoft's model, which happens to mesh almost perfectly with the other important NAC models, IT managers won't have to worry about interoperability or vendor lock-in at that point in the scheme.

Policy Enforcement in MS-NAP

The enforcement mechanisms for MS-NAP differ based on the approach used. Within the 802.1X version, the role of Policy Enforcement Point is delegated to the edge switches and access points. Because Microsoft doesn't make switch or router hardware, MS-NAP utilizes the communication capabilities of the standardized RADIUS protocol. The benefit of this model is that it ensures an open, cross-vendor separation between policy determination and policy enforcement. The drawback is that enforcement in MS-NAP is limited to the switch capabilities which are dynamically controllable by RADIUS.

At the back end Policy Decision Point, Microsoft offers up its new Network Policy Server (to ship with Longhorn server), a RADIUS-based server replacing Microsoft's older Internet Authentication Service. The Network Policy Server contains the functionality of the Network Access Authority, including authentication and policy management, with a separate Network Access Protection Administration Server which handles the same functions of the IETF's Server Broker component, gluing the authentication server to third-party health verifier plug-ins. On top of the Administration Server, using a Microsoft-defined API, are System Health Validators, the equivalent of IETF-defined Posture Verifiers, which receive Statements of Health from System Health Agents on the client and provide answers back to the Administration Server.

Within the 802.1X approach to MS-NAP, the client-to-server communication is based on the PEAP protocol and the Policy Decision Point provides enforcement information using attributes within the RADIUS response.

Microsoft's Big Picture

The diagram below, taken from one of the MS-NAP architecture white papers, summarizes the components and protocols that Microsoft sees as part of their NAC architecture. An interesting twist on other NAC architectures is the Health Certificate Server device marked "HCS". This is key to the concept of a "Health Certificate." Using a combination of existing products to create a Web-based PKI server (called the Health Certificate Server), MS-NAP supports the idea of creating a digital certificate that can be used in place of Statements of Health. Rather than try and send Statements of Health around at authentication time, a client proves its health to the Health Certificate Server using normal System Health Agents and Statements of Health over an HTTP/S connection. It then receives a digital certificate that it can use as a statement of health (instead of normal user credentials for authentication) when using VPN connections, thus providing a faster connection to the end user.

