

# What is TCG's Trusted Network Connect?

TCG's Trusted Network Connect architecture provides an outstanding way of thinking about NAC by dividing the problem into three separate entities: the Access Requestor (AR), the Policy Enforcement Point (PEP), and the Policy Decision Point (PDP).

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.

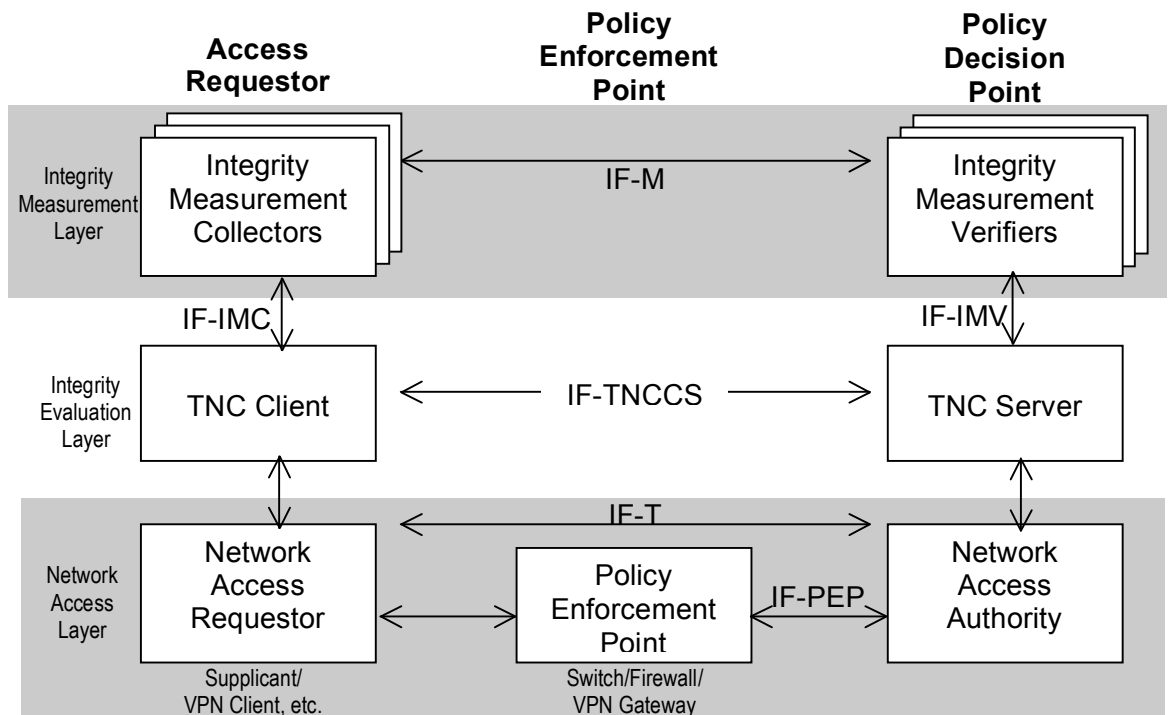
## NAC Entities in TCG-TNC

TCG's Access Requestor is a combination of the entity that's trying to gain access to the network, such as a laptop or desktop computer, as well as the software and drivers that implement the authentication and end-point security assessment parts of NAC. TCG actually divides the Access Requestor into three smaller pieces. At the bottom is a Network Access Requestor (NAR), some software that is used by the client to connect to the network, request access, and provide authentication. For example, in the world of 802.1X, an 802.1X supplicant would be a network access requester. If you were connecting using an IPsec VPN, then the IPsec VPN client would have that function.

On top of the Network Access Requestor, still on the client system and part of the Access Requestor, are Integrity Measurement Collectors (IMCs). These are software components that are responsible for evaluating the security posture of the end system. For example, if your policy is that everyone has to be running anti-virus software, then your anti-virus software vendor would provide a plug-in that provides status information on the anti-virus software. TCG's Trusted Network Connect divides this task up into two pieces: the Integrity Measurement Collectors themselves, and the TNC Client (TNCC) that collects information from the IMCs and helps to package it up for policy evaluation.

The next piece of the Trusted Network Connect NAC architecture is the Policy Enforcement Point. This is the easiest part of the whole picture to understand because it's also the least complicated: the Policy Enforcement Point is exactly what it sounds like: the point where policy is enforced. TCG's NAC doesn't actually describe what kinds of policy-based enforcement are available, although the architectural documents do specifically describe how quarantine and remediation might be part of the policy enforcement part of a NAC solution.

The last part of the Trusted Network Connect NAC architecture is the Policy Decision Point. Like the Access Requestor, TCG divides the Policy Decision Point into three parts. The bottom piece, which is in charge of talking to the authentication server and communicating decisions to the



Policy Enforcement Point, is the Network Access Authority (NAA). In a typical network, the Network Access Authority would likely be an AAA server.

Behind the Network Access Authority are Integrity Measurement Verifiers (IMVs). These Verifiers are the counterparts to the Collectors on the client. They receive the reports that the client Integrity Measurement Collectors send, and provide verification information back to the Policy Decision Point. The Verifiers (on the Policy Decision Point) and the Collectors (on the Access Requestor, or client) are a matched set. They can talk to each other, through a tunnel provided by all the other pieces, using whatever proprietary vendor-specific protocol they want, so long as when the result pops out, it is passed to the Policy Decision Point using TCG's standardized protocol.

As in the Access Requestor (client) side, the Trusted Network Connect architecture layers a thin TNC Server (TNCS) piece that is responsible for handling the interface between the Integrity Measurement Verifiers and the Network Access Authority.

### **Integrating TCG-TNC in Existing Networks**

The Trusted Network Connect NAC architecture is very obviously designed to work within an existing network access control architecture, the 802.1X authentication and authorization system (although TNC hasn't limited themselves to this environment and intends to add bindings for VPNs and other access and control methods). If you rename the client-side Network Access Requestor to "802.1X supplicant," the Policy Enforcement Point to "802.1X-compatible switch or access point," and the Network Access Authority Policy Decision Point to "802.1X RADIUS server," then Trusted Network Connect NAC is simply a bit of software that sits on top of an existing 802.1X deployment to add end-point security assessment into the mix.

This becomes even more obvious if you look at which of the protocols the Trusted Network Connect NAC team chose to publish first: those that allow the vendor-supplied Integrity Measurement Collectors to talk to a vendor-neutral TNC Client on the client system, along with those that allow the vendor-supplied Integrity Measurement Verifier to talk to the vendor-neutral TNC Server on the Policy Decision Point end. The Trusted Network Connect architecture relies heavily on existing 802.1X mechanisms such as authentication and tunneling to get all the other pieces to work.

None of this means that the Trusted Network Connect architecture won't support other kinds of Policy Enforcement Points, such as firewalls, VPN concentrators, or core switches. However, the natural first steps for TNC deployment, and the ones we are showing in the Interop iLabs, are based on 802.1X authentication and port access controls, and VLAN switching access controls based on RFC 3580.

### **Interfaces and Protocols in TCG-TNC**

**IF-IMC** – used to gather integrity measurements from IMCs so they can be communicated to IMVs and to enable messages to be exchanged between the IMCs and IMVs. This is defined in a TCG IF-IMC specification.

**IF-IMV** – used to receive integrity measurements sent from client-side IMCs to corresponding IMVs, to enable message exchanges between the IMCs and the IMVs, and to allow IMVs to supply their recommendations to the TNC Server. This is defined in a TCG IF-IMV specification.

**IF-TNCCS** – defines a protocol that conveys messages from IMCs to IMVs, and vice versa, as well as session management messages and synchronization information. This is a fairly simple multiplexing protocol, and was released this week (May 1, 2006) at Interop.

**IF-M** – a vendor-specific protocol between IMCs and IMVs, carried over the IF-TNCCS interface. Only a subset of this protocol will be standardized; the rest will be specific to each IMC/IMV pair.

**IF-T** – this transports messages between the Access Requestor (AR) as an entity and the Policy Decision Point (PDP) as an entity. The TNC will not be standardizing this as its own protocol, but will be providing bindings showing how these messages could be carried over existing protocols, such as EAP within 802.1X. The first of these bindings, a protocol for tunneled EAP methods (including the EAP-TNC transport mechanism) was released this week.

**IF-PEP** – allows the Policy Decision Point (PDP) to communicate with the Policy Enforcement Point (PEP). For example, the PDP could instruct the PEP to isolate the Access Requestor (AR) during remediation. The TNC just released this week the IF-PEP binding for RADIUS. Other protocols, such as LDAP, could be added in the future. This would be most interesting in environments where protocols other than RADIUS are being used, such as is commonly done in SSL VPNs.