

**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE**

NETWORK SECURITY
TECHNOLOGIES, LLC,

Plaintiff,

v.

PULSE SECURE, LLC,

Defendant.

Civil Action No. _____

JURY TRIAL DEMANDED

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Network Security Technologies, LLC (“Network Security” or “Plaintiff”), for its Complaint against Defendant Pulse Secure, LLC, (“Pulse Secure” or “Defendant”) alleges the following:

NATURE OF THE ACTION

1. This is an action for patent infringement arising under the Patent Laws of the United States, 35 U.S.C. § 1 *et seq.*

THE PARTIES

2. Plaintiff is a limited liability company organized under the laws of the State of Delaware can be served through its registered agent at 717 North Union Street, Wilmington, Delaware 19805.

3. Upon information and belief, Pulse Secure is a limited liability company organized and existing under the laws of Delaware, with a place of business at 2700 Zanker Road, Suite 200, San Jose, California 95134, and can be served through its registered agent, The Corporation Trust Company, Corporation Trust Center, 1209 Orange Street, Wilmington, Delaware 19801.

Upon information and belief, Pulse Secure sells and offers to sell products and services

throughout the United States, including in this judicial district, and introduces products and services that into the stream of commerce and that incorporate infringing technology knowing that they would be sold in this judicial district and elsewhere in the United States.

JURISDICTION AND VENUE

4. This is an action for patent infringement arising under the Patent Laws of the United States, Title 35 of the United States Code.

5. This Court has subject matter jurisdiction under 28 U.S.C. §§ 1331 and 1338(a).

6. Venue is proper in this judicial district under 28 U.S.C. §1400(b). On information and belief, Defendant is incorporated in the State of Delaware.

7. On information and belief, Defendant is subject to this Court's general and specific personal jurisdiction because Defendant has sufficient minimum contacts within the State of Delaware and this District, pursuant to due process and/or the Del. Code. Ann. Tit. 3, § 3104 because Defendant purposefully availed itself of the privileges of conducting business in the State of Delaware and in this District, because Defendant regularly conducts and solicits business within the State of Delaware and within this District, and because Plaintiff's causes of action arise directly from Defendant's business contacts and other activities in the State of Delaware and this District. Further, this Court has personal jurisdiction over Defendant because Defendant is incorporated in Delaware and has purposely availed itself of the privileges and benefits of the laws of the State of Delaware.

COUNT I – INFRINGEMENT OF U.S. PATENT NO. 8,234,705

8. The allegations set forth in the foregoing paragraphs 1 through 7 are incorporated into this First Count.

9. On July 31, 2012, U.S. Patent No. 8,234,705 (“the ’705 patent”), entitled “Contagion Isolation and Inoculation,” was duly and legally issued by the United States Patent and Trademark Office. A true and correct copy of the ’705 patent is attached as Exhibit 1.

10. The claims of the ’705 patent are directed to technical solutions to technical problems related to network security. Network security is a tremendous concern for modern enterprises, since a breach of an internal network can have severe repercussions, including major financial losses, data theft, disclosure of sensitive information, network disruptions, and data corruption—any of which could have devastating consequences to a business, at any scale. While a network security appliance can be extremely adept at keeping unwanted external intrusions into the network out, it is well known that perhaps the most exploitable component of a network is the human element. Information technology (IT) professionals may be able to keep on-site systems secure and up to date, but a worker’s personal laptop or mobile device is a significant security risk that would allow attackers (or just unwanted malware) access into a business’s network, bypassing IT security measures. The technology of the ’705 Patent closes this loophole by verifying that any device attempting to access a company’s network meets the company’s standards for network security and will not introduce dangerous programs into the company’s network. Through this technology, as implemented by companies including the Defendant, businesses can increase security of vulnerable elements of their networks.

11. The claims of the ’705 patent do not merely recite the performance of some business practice known before the pre-Internet world along with the requirement to perform it on the Internet. Instead, the claims of the ’705 patent are rooted in improvements to computerized network security technology in order to overcome problems specifically arising in the realm of computerized network security.

12. The claims of the '705 patent recite subject matter that is not merely the routine or conventional use of network infrastructure. Instead, the claimed inventions are directed to particularized methods of assessing and responding to an external network access request such that network integrity is maintained. The '705 patent claims specify how a secure network can assess and respond to an external network access request without jeopardizing network integrity.

13. The claims of the '705 patent do not preempt all ways of using network security technology, do not preempt any other well-known or prior technology, and do not impermissibly block future developments.

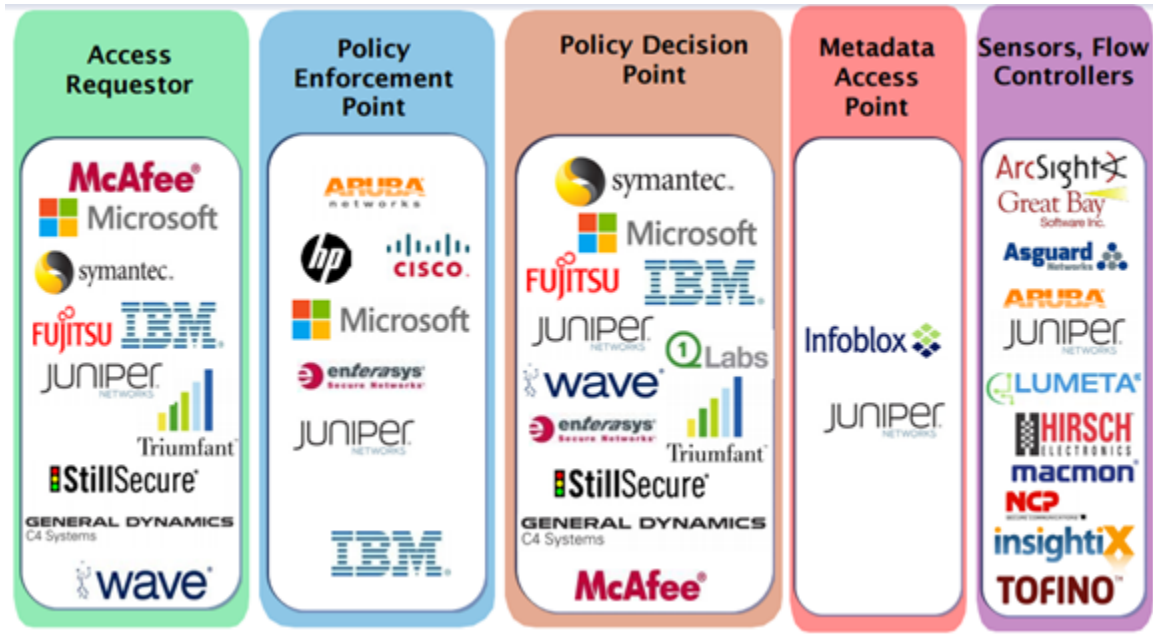
14. Each claim of the '705 patent recites a combination of elements that amounts to significantly more than a patent on an ineligible concept.

15. Plaintiff is the assignee and owner of the right, title and interest in and to the '705 patent, including the right to assert all causes of action arising under said patent and the right to any remedies for infringement of said patent.

16. Upon information and belief, Defendant has directly infringed and continues to directly infringe at least claim 1 of the '705 patent by making, using, offering for sale, selling, or importing products that perform (including but not limited to Defendant's Policy Secure solution) or services related to methods for isolating potential network contagions and inoculating networks against the isolated contagions (the "Accused Instrumentalities").

17. On information and belief, the Accused Instrumentalities leverage Trusted Network Connect (or Trusted Network Communications) (TNC) open architecture promulgated by the Trusted Network Connect Work Group of the Trusted Computing Group (TCG). *See, e.g.*, TCG Trusted Network Communications TNC Architecture for Interoperability, Specification Version 1.5, Revision 4, published May 7, 2012 (hereinafter the "TNC

Specification”) at page 9, available at https://trustedcomputinggroup.org/wp-content/uploads/TNC_Architecture_v1_5_r4.pdf. On information and belief, Defendant adopted the TNC standards at least as of 2013. *See, e.g.*, Trusted Computing Group’s Trusted Network Connect Standards for Network Security presentation available at <https://trustedcomputinggroup.org/wp-content/uploads/TNC-Briefing-2013-12-10.pdf>.



See also, Pulse Policy Secure Datasheet at page 7 available at https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjus5a-pcfVAhUHNI8KHfIeCJMQFggnMAA&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Fdatasets%2F1474%2FPulseSecure_Datasheet_PolicySecure_160224.pdf&usg=AFQjCNFw38iOU6ovDU9DCR6fFEXQYFjTbw (“Adopts and provides strong support for the TCG’s TNC open standards for network access control and security”)

Table 4: Standards-Based, Interoperable Access Control

Feature	Feature Description	Benefits
TNC open standards support, including IF-MAP support and Windows SOH and embedded NAP Agent support (optional)	<ul style="list-style-type: none"> Adopts and provides strong support for the TCG's TNC open standards for network access control and security. Adopts the TNC's open standard IF-MAP, enabling integration with third-party network and security devices, including devices that collect and (through IF-MAP) share information on the state and status of a network, user, or device. Pulse Policy Secure-enabled PSA or MAG Series Appliance can serve as Metadata Access Point (MAP) servers, enabling collected data to be used in formulating policies and appropriate access actions. Through the TNC SOH standard, leverages preinstalled Windows 10, Microsoft Windows 8.1, Windows 8, Windows RT, Windows 7, Windows Vista, clients for access control with the Pulse Access Control Service, allowing use of the Windows Security Center (WSC) SOH in access control decisions. 	<ul style="list-style-type: none"> Empowers organizations to select endpoint and network security solutions that meet their needs without concern for interoperability. Enables ease of deployment, leading to faster ROI. Integrates existing, third-party network and security devices into the access control platform. Streamlines client deployment, simplifying access control rollout and implementation.

18. Upon information and belief, Defendant makes, uses, offers for sale, sells, and/or imports the Accused Instrumentalities. *See, e.g.*, Defendant’s description of “Pulse Policy Secure Solution Overview” disclosing the following: “Policy Secure solution coordinates network security compliance and provides the control required to support network applications, manage network use, and reduce threats from unauthorized users and compromised host machines attempting to access the network. You configure rules in Host Checker policies to specify the minimum criteria for the security compliance of host machines that are allowed to enter the network.” from page 44 of 1100 of Defendant’s Pulse Policy Secure complete software guide available at https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwj76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fpps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw.

Pulse Policy Secure Solution Overview

The Pulse Policy Secure solution provides a mechanism for authenticating users and assessing the health of their host machines to control network access.

Policy Secure solution coordinates network security compliance and provides the control required to support network applications, manage network use, and reduce threats from unauthorized users and compromised host machines attempting to access the network.

You configure rules in Host Checker policies to specify the minimum criteria for the security compliance of host machines that are allowed to enter the network.

The policies that you create control access for users, the client or agent that users access the network with, and the host machine or endpoint on which the clients run.

19. Claim 1 of the '705 patent recites a method for protecting a network, comprising: detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network, wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host, receiving a response, and determining whether the response includes a valid digitally signed attestation of cleanliness, wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host; when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network, wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host, serving a quarantine notification page to the first host when the service request comprises a web server request, and in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated

with a remediation host configured to provide data usable to remedy the insecure condition; and permitting the first host to communicate with the remediation host.

20. As demonstrated in the exemplary images and text below, Defendant has directly infringed and continues to directly infringe at least claim 1 of the '705 patent by making, using, offering for sale, selling, and/or importing the Accused Instrumentalities, which perform a method for protecting a network, comprising: detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network, wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host:

The Accused Instrumentalities detect when an Access Requestor (AR) client (a first host) is attempting to connect to the protected network, but has not yet passed the integrity-verification (thus is in insecure condition). *See, e.g.*, the TNC Specification at pages 20 and 26 available at https://trustedcomputinggroup.org/wp-content/uploads/TNC_Architecture_v1_5_r4.pdf.

When a network connection attempt is triggered (automatically or by user request), the NAR at the AR (client) initiates a connection request at the link and network layers.

***Isolation:* If the AR has been authenticated and is recognized to be one that has some privileges on the network but has not passed the integrity-verification by the IMV, the PDP may return instructions to the PEP to redirect the AR to an isolation environment where the AR can obtain integrity-related updates.**

As shown in the figure below, detecting the insecure condition includes using the PTS (Platform Trust Service) Protocol to contact a trusted computing base, the Policy Decision Point (PDP), to use the PTS Integrity Measurement Collector of the first host. *See*, the TCG Attestation PTS Protocol: Binding to TNC IF-M Specification, Version 1.0, Revision 28, published August 24, 2011 (hereinafter “IFM PTS”) at page 10 available at https://www.trustedcomputinggroup.org/wp-content/uploads/IFM_PTS_v1_0_r28.pdf.

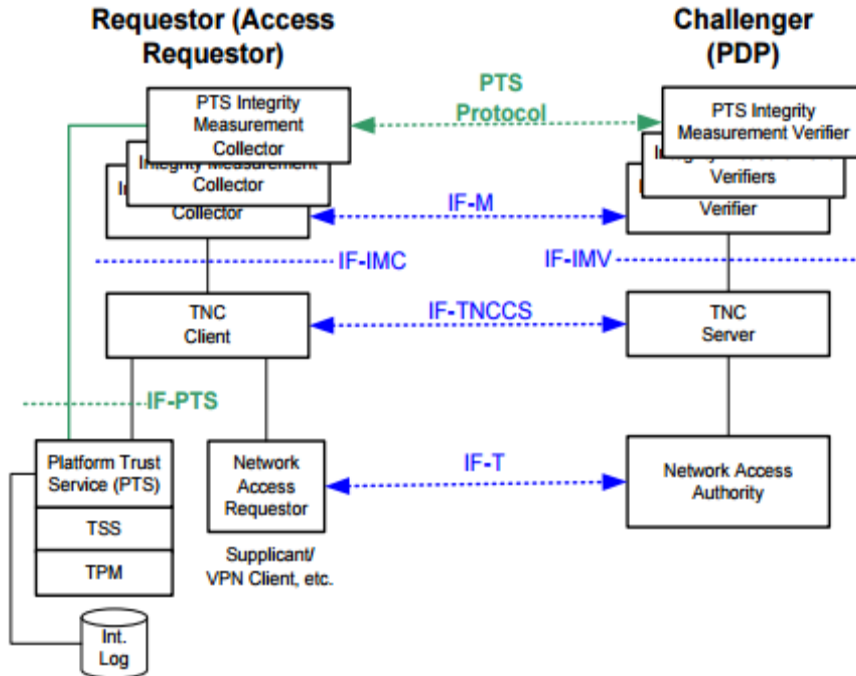


Figure 1: PTS Protocol Integration with TNC Architecture

Notice that the TNC Architecture contains several layered protocols (IF-T, IF-TNCCS and IF-M). The PTS Protocol will be carried within the payloads of the IF-M protocol, so would layer hierarchically on top of the IF-M protocol. The PTS Protocol operates between the PTS-IMC and PTS-IMV to enable PTS-based attestation leveraging the underlying TPM. The PTS-IMC uses a local IPC channel to the PTS (discussed in the IF-PTS specification) to obtain the necessary attestation evidence. Use of the IF-PTS interface and the TSS middleware stack components are optional so implementations might leverage the PTS or TPM in other ways (e.g. the PTS could have other techniques for interacting with a TPM to obtain measurements).

See also, the TCG's Trusted Platform Module Library, Part 1: Architecture, Family 2.0, Level 00 Revision 01.07 draft published March 13, 2014 (hereinafter, "TPM Library") at page 23 available at <https://www.trustedcomputinggroup.org/wp-content/uploads/TPM-Rev-2.0-Part-1-Architecture-01.07-2014-03-13.pdf>.

9.2.2 Trusted Computing Base

A trusted computing base (TCB) is the collection of system resources (hardware and software) that is responsible for maintaining the security policy of the system. An important attribute of a TCB is that it be able to prevent itself from being compromised by any hardware or software that is not part of the TCB.

receiving a response:

The response is then received in the form of "attestation information." See, e.g., the IFM PTS at pages 14-15 available at https://www.trustedcomputinggroup.org/wp-content/uploads/IFM_PTS_v1_0_r28.pdf. (Discussed further below).

and determining whether the response includes a valid digitally signed attestation of cleanliness, wherein the valid digitally signed attestation of cleanliness includes at least one of an attestation that the trusted computing base has ascertained that the first host is not infested:

If the remote challenger is clean, the “attestation information” will include a signed (and cryptographically verifiable) set of “attestation evidence” which is an attestation of cleanliness. *See, e.g.*, the IFM PTS at pages 14-15 available at https://www.trustedcomputinggroup.org/wp-content/uploads/IFM_PTS_v1_0_r28.pdf.

2.5.2 TPM Quote

This section summarizes how the TPM's ability to quote PCRs provides assurance that the attestation evidence provided is trustworthy and protected from local tampering. The TPM provides a number of shielded storage locations and ordinals to perform transformations on data using the contents of these locations. For example, the TPM houses cryptographic keys that are only usable inside the TPM. One such key is the Attestation Identity Key (AIK) which is limited by the TPM to only be used to sign the contents of the TPM's PCRs (which are protected in TPM shielded storage). On a platform where the PCRs are set to reflect the operational content of the system, this combination of protected PCRs and AIK enable an attestation mechanism to be verifiable by remote parties.

Specifically, a remote challenger can request attestation information about a system and obtain an AIK signed set of “attestation evidence” that is cryptographically verifiable as having been generated using a TPM-resident AIK and PCRs. By empowering the remote challenger to be able to retrieve this signed set of PCRs with proof that they were resident inside a TPM, this mechanism allows the challenger to have confidence that it can determine (potentially using the measurement log and policy) what software has been run on the endpoint without fear of being spoofed by malware.

and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host:

The attestation information provided by the Accused Instrumentalities includes measurements of the client's integrity (ascertained that the first host is not infested) and the software versions (the presence of a patch or a patch level associated with a software component on the first host). *See, e.g.*, the TNC Specification at page 15 available at https://trustedcomputinggroup.org/wp-content/uploads/TNC_Architecture_v1_5_r4.pdf.

Integrity Measurement Collector (IMC): The IMC function is a software component that runs on an AR, measuring security aspects of the AR's integrity. Examples include the Anti-Virus parameters on the Access Requestor, Personal Firewall status, software versions, and other security aspects of the AR. Note that the TNC Architecture is designed for multiple IMCs to interact with a single (or multiple) TNC Client/TNC Server, thereby allowing customers to deploy complex integrity policies involving a range of vendors products.

This process is performed when clients attempt to connect to the protected network, to prevent outside infections from getting in and to ensure the integrity has not fallen below expectations—such as new updates not being installed—since a previous connection. A Use Case Walkthrough provided by the Trusted

Computing Group illustrates a standard procedure for new connection attempts. See pages 55-59 of the TCG TNC IF-IMV Specification, Version 1.0, Revision 3, published May 3, 2005 (hereafter “TNC IF-IMV Specification”) available at https://www.trustedcomputinggroup.org/wp-content/uploads/TNC_IFIMV_v1_0_r3.pdf.

7.4 Network Connect

1. The endpoint's NAR attempts to connect to a network protected by a PEP, thus triggering an Integrity Check Handshake. There are other ways that an Integrity Check Handshake can be triggered, but this will probably be the most common. For those other ways, the next few steps may be significantly different.
2. The PEP sends a network access decision request to the PDP (NAA or TNCS). Depending on configuration, the PEP may contact the NAA first or the TNCS. The ordering of user authentication, platform authentication, and integrity check is also subject to configuration. Here we present what will probably be the most common order: first user authentication, then platform authentication, then integrity check.
3. The NAA performs user authentication with the NAR. Based on the NAA's policy, the user identity established through this process may be used to make immediate access decisions (like deny). If an immediate access decision has been made, skip to step 16. User authentication may also involve having the NAR authenticate the NAA.
4. The NAA informs the TNCS of the connection request, providing the user identity and other useful info (service requested, etc.).
5. The TNCS performs platform authentication with the TNCC, if required by TNCS policy. This includes verifying the IMC hashes collected during TNCC Setup. If an immediate access decision has been made, skip to step 16. Platform authentication may be mutual so the TNCC can be sure it's talking to a secure server.
6. The TNCC uses IF-IMC to fetch IMC messages.
7. The TNCS uses IF-IMV to inform each IMV that an Integrity Check Handshake has started. **[IF-IMV] If this is a new network connection, the TNCS calls TNC_IMV_NotifyConnectionChange with the newState parameter set to TNC_CONNECTION_STATE_CREATE to indicate that a new network connection has been created. Then the TNCS calls TNC_IMV_NotifyConnectionChange with the newState parameter set to TNC_CONNECTION_STATE_HANDSHAKE.**

8. The TNCC passes the IMC messages to the TNCS. This and all other TNCC-TNCS communications can be sent directly but they will often be relayed through one or more of the NAR, PEP, and NAA.
9. The TNCS passes each IMC message to the matching IMV or IMVs through IF-IMV (using message types associated with the IMC messages to find the right IMV). If there are no IMC messages, skip to step 13. **[IF-IMV] The TNCS delivers the IMC messages to the IMVs by calling `TNC_IMV_ReceiveMessage`. The IMVs may call `TNC_TNCS_SendMessage` before returning from `TNC_IMV_ReceiveMessage` if they want to send a response. When the TNCS has delivered all the IMC messages to the IMVs, it calls `TNC_IMV_BatchEnding` to inform them of this fact. The IMVs may call `TNC_TNCS_SendMessage` before returning from `TNC_IMV_BatchEnding` if they want to send a message to an IMV.**
10. Each IMV analyzes the IMC messages. If an IMV needs to exchange more messages (including remediation instructions) with an IMC, it provides a message to the TNCS and continues with step 11. If an IMV is ready to decide on an IMV Action Recommendation and IMV Evaluation Result, it gives this result to the TNCS through IF-IMV. If there are no more messages to be sent to the IMC from any of the IMVs, skip to step 13. **[IF-IMV] As described in the previous step, IMVs send messages by calling `TNC_TNCS_SendMessage` before returning from `TNC_IMV_ReceiveMessage` and `TNC_IMV_BatchEnding`. IMVs give their results to the TNCS by calling `TNC_TNCS_ProvideRecommendation` at any time.**
11. The TNCS sends the messages from the IMVs to the TNCC.
12. The TNCC sends the IMV messages on to the IMCs through IF-IMC so they can process the messages and respond. Skip to step 8.
13. If there are any IMVs that have not given an IMV Action Recommendation to the TNCS, they are prompted to do so through IF-IMV. **[IF-IMV] The TNCS gives this prompt by calling `TNC_IMV_SolicitRecommendation`. The IMVs provide their recommendations by calling `TNC_TNCS_ProvideRecommendation`.**
14. The TNCS considers the IMV Action Recommendations supplied by the IMVs and uses an integrity check combining policy to decide what its TNCS Action Recommendation should be.
15. The TNCS sends its TNCS Action Recommendation to the NAA. The NAA may ignore or modify this recommendation based on its policies but will typically abide by it.
16. The NAA sends a copy of its final network access decision response to the TNCS. The TNCS may send a copy of the network access decision to the TNCC. The TNCS also informs the IMVs of the network access decision response via IF-IMV. **[IF-IMV] The TNCS calls `TNC_IMV_NotifyConnectionChange` with the `newState` parameter set to `TNC_CONNECTION_STATE_ACCESS_ALLOWED`, `TNC_CONNECTION_STATE_ACCESS_ISOLATED`, or `TNC_CONNECTION_STATE_ACCESS_NONE`.**
17. The NAA sends its network access decision response to the PEP.
18. The PEP implements the network access decision response. During this process, the NAR is typically informed of the decision. The TNCC may be informed by the NAR or may discover that a new network has come up.
19. If step 6 was not executed, the network connect process is complete. Otherwise, the TNCC informs the IMCs of the network access decision response via IF-IMC.
20. If the IMCs need to perform remediation, they perform that remediation. Then they continue with Handshake Retry After Remediation. If no remediation was needed, the use case ends here.

The Accused Instrumentalities perform the TNC process. *See, e.g.*, Pulse Policy Secure Datasheet at page 5 available at <https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjus5a->

pcfVAhUHNI8KHfIeCJMQFggnMAA&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Fdatasets%2F1474%2FPulseSecure_Datasheet_PolicySecure_160224.pdf&usg=AFQjCNFw38iOU6ovDU9DCR6fFEXQYfjTbw.

Automated patch assessment checks and remediation (optional)	<ul style="list-style-type: none"> • Can tie access directly to the presence or absence of specific hot fixes for defined operating systems and applications, and performs role-based, predefined patch management checks according to the severity level of vulnerabilities. • Installed Systems Management Server (SMS) and/or System Center Configuration Manager (SCCM) 2007 can be leveraged to automatically check for patch updates, quarantining, remediating, and providing authorized network access once a device has been remediated. 	<ul style="list-style-type: none"> • Enables enhanced, granular endpoint device health and security state assessments. • Minimizes user interaction and downtime through automatic remediation and management of patches for endpoint devices, reducing help desk calls.
--	---	--

See also, Defendant’s description of “Host Checker and Trusted Network Computing” disclosing the following: “You can use Host Checker to perform health and security evaluations on endpoints before allowing them to connect to the network and to access protected resources. Host Checker can check for third-party applications, files, process, ports, registry keys, and custom DLLs on hosts. Based on the results of the checks, Host Checker can deny or allow access to protected resources. For example, you can check for virus detection software before allowing a user access to a realm. You can configure policies that launch the software on the user’s system, map the user to roles based on individual policies defined in your own DLL,” from page 674 of 1100 of Defendant’s Pulse Policy Secure complete software guide available at https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

Host Checker and Trusted Network Computing

Host Checker is a client-side agent that performs endpoint health and security checks for hosts that attempt to connect to Policy Secure. Host Checker is based on open standards defined by the Trusted Computing Group (TCG).

Trusted Network Connect (TNC) is a subgroup of the TCG that created an architecture and set of standards for verifying endpoint integrity and policy compliance during or after a network access request. Many of the TCG members participated in the definition and specification of the TNC architecture. The TNC defined several standard interfaces that enable components from different vendors to securely operate together. The TNC architecture is designed to build on established standards and technologies, such as 802.1X, RADIUS, IPsec, EAP, and TLS/SSL. For more information about TNC, see www.trustedcomputinggroup.org. Using technology based on the TNC architecture and standards, the Host Checker component of the Unified Access Control solution provides a comprehensive approach to assess the trustworthiness of endpoints.

You can use Host Checker to perform health and security evaluations on endpoints before allowing them to connect to the network and to access protected resources. Host Checker can check for third-party applications, files, process, ports, registry keys, and custom DLLs on hosts. Based on the results of the checks, Host Checker can deny or allow access to protected resources. For example, you can check for virus detection software before allowing a user access to a realm. You can configure policies that launch the software on the user's system, 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. When a user's computer does not meet the requirements you specify, you can configure options that display remediation instructions to users so they can bring their computers into compliance.

when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network:

The Accused Instrumentalities use the Assessment Phase to determine the client's integrity status. Once this determination is made, the Integrity Measurement Verifier (IMV) "can make one of three IMV Action-Recommendations (Allow, Isolate or Block)." *See, e.g.*, the TNC Specification at page 27 available at https://trustedcomputinggroup.org/wp-content/uploads/TNC_Architecture_v1_5_r4.pdf.

5.2 Assessment Phase

In the Assessment Phase, the TNC Client reports its current integrity status to the TNC Server. Upon receiving the client integrity status, the IMVs with the aid of the TNCS perform an assessment of the AR based on the set of policies defined by the network administrator. The IMV can make one of three IMV Action-Recommendations (Allow, Isolate or Block) or it can make no recommendation.

A client AR that fails integrity variation is isolated (quarantined) onto an "Isolation Network." "This protects the AR from the full network and vice versa, preventing the spread of viruses and worms."

5.3 Isolation Phase

An important tool in the effort to remediate ARs that fail integrity verification is the isolation of that AR to a separate network – referred to here as the Isolation Network – in order to provide remediation services to the AR. This protects the AR from the full network and vice versa, preventing the spread of viruses and worms. There are a number of technical approaches today to achieve network isolation for the AR. Two of these are as follows:

- (a) *VLAN Containment*: VLAN containment permits the AR to access the network in a limited fashion. Typically the primary purpose of the limited access is to allow the AR to access on-line sources of remediation data (e.g. virus definition file updates, worm removal software, software patches, etc). In some cases, no remediation is offered and the AR is instead offered access to limited services, in such a fashion as to limit the potential for impact to the network or other attached hosts. RADIUS provisions VLAN containment using the Tunnel-Private-Group-ID attribute, as specified in RFC3580 [21].
- (b) *IP Filters*: In the case of IP filters, the PEP is configured with a set of filters which defines network locations reachable by the isolated AR. Packets from the AR destined to other network locations are simply discarded by the PEP. RADIUS selects filter rules for application to a network access session using the Filter-ID attribute (see RFC2865 and RFC3580) [21].

wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host:

This isolation network provides “limited or quarantined access” which prevents the host from sending data to other hosts associated with the protected network. This will require “remediation instructions” be provided in place of standard responses to service requests. *See, e.g.*, pages 13-15 of the TNC IF-IMV Specification available at https://www.trustedcomputinggroup.org/wp-content/uploads/TNC_IFIMV_v1_0_r3.pdf.

2.6.3 Remediation and Handshake Retry

In several cases, it is useful to retry an Integrity Check Handshake. First, an endpoint may be isolated until remediation is complete. Once remediation is complete, an IMC can inform the TNCC of this fact and suggest that the TNCC retry the Integrity Check Handshake. Second, a TNCS can initiate a retry of an Integrity Check Handshake (if the TNCS or IMV policies change or as a periodic recheck). Third, an IMC or IMV can request a handshake retry in response to a condition detected by the IMC or IMV (suspicious activity, for instance). In any case, it's generally desirable (but not always possible) to reuse state established by the earlier handshake and to avoid disrupting network connectivity during the handshake retry.

To support handshake retries, the TNCS MAY maintain a network connection ID after an Integrity Check Handshake has been completed. This network connection ID can then be used by the TNCS to inform IMVs that it is retrying the handshake or by an IMV to request a retry (due to policy change or another reason).

Handshake retry may not always be possible due to limitations in the TNCC, NAR, PEP, or other entities. In other cases, retry may require disrupting network connectivity. For these reasons, IF-IMV supports handshake retry and requires IMVs to handle handshake retries (which is usually trivial) but does not require TNCSs to honor IMV requests for handshake retry. In fact, IF-IMV requires an IMV to provide information about the reason for requesting handshake retry so that the TNCS can decide whether it wants to retry (which may disrupt network access).

Note that remediation instructions are delivered from IMVs to IMCs through standard IMV-IMC messages (see section 2.6.4, “Message Delivery”). There is no special support in IF-IMV for this feature. IMVs SHOULD send remediation instructions to IMCs before returning an IMV Action Recommendation and IMV Evaluation Result to the TNCS so the instructions are delivered before the handshake is completed.

2.6.4 Message Delivery

One of the critical functions of the TNC architecture is conveying messages between IMCs and IMVs. Each message sent in this way consists of a message body, a message type, and a recipient type.

The message body is a sequence of octets (bytes). The TNCC and TNCS SHOULD NOT parse or interpret the message body. They only deliver it as described below. Interpretation of the message body is left to the ultimate recipients of the message, the IMCs or IMVs. A zero length message is perfectly valid and MUST be properly delivered by the TNCC and TNCS just as any other IMC-IMV message would be.

The message type is a four octet number that uniquely identifies the format and semantics of the message. The method used to ensure the uniqueness of message types while providing for vendor extensions is described below.

The recipient type is simply a flag indicating whether the message should be delivered to IMVs or IMCs. Messages sent by IMCs are delivered to IMVs and vice versa. All messages sent by an IMV through IF-IMV have a recipient type of IMC. All messages received by an IMV through IF-IMV have a recipient type of IMV. The recipient type does not show up in IF-IMC or IF-IMV, but it helps in explaining message routing.

The routing and delivery of messages is governed by message type and recipient type. Each IMC and IMV indicates through IF-IMC and IF-IMV which message types it wants to receive. The TNCC and TNCS are then responsible for ensuring that any message sent during an Integrity Check Handshake is delivered to all recipients that have a recipient type matching the message's recipient type and that have indicated the wish to receive messages whose type matches the message's message type. If no recipient has indicated a wish to receive a particular message type, the TNCC and TNCS can handle these messages as they like: ignore, log, etc.

WARNING: The message routing and delivery algorithm just described is not a one-to-one model. A single message may be received by several recipients (for example, two IMVs from a single vendor, two copies of an IMC, or nosy IMVs that monitor all messages). If several of these recipients respond, this may confuse the original sender. IMCs and IMVs MUST work properly in this environment. They MUST NOT assume that only one party will receive and/or respond to a message.

IF-IMV allows an IMV to send and receive messages using this messaging system. Note that this system should not be used to send large amounts of data. The messages will often be sent through PPP or similar protocols that do not include congestion control and are not well suited to bulk data transfer. If an IMC needs to download a patch (for instance), the IMV should indicate this by reference in the remediation instructions. The IMC will process those instructions after network access (perhaps isolated) has been established and can then download the patch via HTTP or another appropriate protocol.

All messages sent with `TNC_TNCS_SendMessage` and received with `TNC_IMV_ReceiveMessage` are between the IMC and IMV. The IMV communicates with the TNCS by calling functions (standard and vendor-specific) in the IF-IMV, not by sending messages. The TNCS should not interfere with communications between the IMC and IMVs by consuming or blocking IMC-IMV messages.

A particular example of the message delivery provided by IF-IMV is the communication of remediation instructions from the IMVs through the TNCS to the TNCC/IMCs. This is one application of IMC-IMV message delivery and in all cases follows the normal IMV-IMC communications path. IF-IMV provides support for communicating remediation instructions to an endpoint using this mechanism. Since the normal IMC-IMV communications path is used to communicate remediation instructions, this specification will not address further the details of how remediation itself is done.

2.6.7 IMV Action Recommendation

One of the assumptions of the TNC architectural model is that IF-IMV provides a means for IMVs to recommend action information to the TNCS, so that isolation can properly be supported on the network. The TNCS then will combine these IMV Action Recommendations using some logic (defined by the TNCS implementers) to come up with an overall TNCS Action Recommendation. Note that the TNCS may choose to ignore any IMV Action Recommendation, but each IMV must be able to recommend an action. Potential choices for IMV Action Recommendations include: recommend full (normal) access; recommend isolation (limited or quarantined access); and recommend denial (no access). The mandatory function `TNC_TNCS_ProvideRecommendation` is the mechanism within IF-IMV for an IMV to indicate its IMV Action Recommendation.

servicing a quarantine notification page to the first host when the service request comprises a web server request, and in the event the service request comprises a DNS query, providing in response an IP address of a quarantine server configured to serve the quarantine notification page if a host name that is the subject of the DNS query is not associated with a remediation host configured to provide data usable to remedy the insecure condition:

As an example of Defendant's implementation of this process in the Accused Instrumentalities, see the following explanation of "Host Checker and Trusted Network Computing." Notably, "based on the results of the checks, Host Checker can deny or allow access to protected resources. For example, you can check for virus detection software before allowing a user access to a realm," and "When a user's computer does not meet the requirements you specify, you can configure options that display remediation instructions to users so they can bring their computers into compliance." This illustrates the use of a quarantine notification page, provided in response when the client issues a web server request or a DNS query ("display remediation instructions"). See, e.g., the Pulse Policy Secure complete software guide at page 674 of 1100 available at https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usq=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

Host Checker and Trusted Network Computing

Host Checker is a client-side agent that performs endpoint health and security checks for hosts that attempt to connect to Policy Secure. Host Checker is based on open standards defined by the Trusted Computing Group (TCG).

Trusted Network Connect (TNC) is a subgroup of the TCG that created an architecture and set of standards for verifying endpoint integrity and policy compliance during or after a network access request. Many of the TCG members participated in the definition and specification of the TNC architecture. The TNC defined several standard interfaces that enable components from different vendors to securely operate together. The TNC architecture is designed to build on established standards and technologies, such as 802.1X, RADIUS, IPsec, EAP, and TLS/SSL. For more information about TNC, see www.trustedcomputinggroup.org. Using technology based on the TNC architecture and standards, the Host Checker component of the Unified Access Control solution provides a comprehensive approach to assess the trustworthiness of endpoints.

You can use Host Checker to perform health and security evaluations on endpoints before allowing them to connect to the network and to access protected resources. Host Checker can check for third-party applications, files, process, ports, registry keys, and custom DLLs on hosts. Based on the results of the checks, Host Checker can deny or allow access to protected resources. For example, you can check for virus detection software before allowing a user access to a realm. You can configure policies that launch the software on the user's system, 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. When a user's computer does not meet the requirements you specify, you can configure options that display remediation instructions to users so they can bring their computers into compliance.

See also, Defendant's "Understanding Host Checker Policy Remediation" which describes the Defendant system's ability to "specify general remediation actions for Host Checker to take if an endpoint does not meet the requirements of a policy. For example, you can display a remediation page to the user that contains specific instructions and links to resources to help the user bring their endpoint into compliance with Host Checker policy requirements." For example, "the user might see a remediation page that contains the following custom instructions, a link to resources, and reason strings:

Your computer's security is unsatisfactory.

Your computer does not meet the following security requirements. Follow the instructions below to fix these problems. When you are done click Try Again. If you Continue without fixing these problems, you may not have access to all of your intranet servers." from page 714 of 1100 of Defendant's Pulse Policy Secure complete software guide available at

https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

Understanding Host Checker Policy Remediation

This topic describes Host Checker policy remediation. It includes the following information:

- Remediation Options
- Remediation User Experience

Remediation Options

You can specify general remediation actions for Host Checker to take if an endpoint does not meet the requirements of a policy. For example, you can display a remediation page to the user that contains specific instructions and links to resources to help the user bring their endpoint into compliance with Host Checker policy requirements.

You can also include a message to users (called a reason string) that is returned by Host Checker or an IMV and that explains why the client machine does not meet the Host Checker policy requirements.

For example, the user might see a remediation page that contains the following custom instructions, a link to resources, and reason strings:

Your computer's security is unsatisfactory.

When specifying remediation actions for a Host Checker policy, Defendant systems allow for setting custom instructions to display to the user on the Host Checker remediation page. This is particularly disclosed in the following:

“Enable Custom Instructions—Enter the instructions to display to the user on the Host Checker remediation page. You can use the following HTML tags to format text and to add links to resources such as policy servers or web sites: `<i>`, ``, `
`, ``, and `<a href>`. For example:
You do not have the latest signature files.
`Click here to download the latest signature files.`”

“Send reason strings—Select this option to display a message to users (called a reason string) that is returned by Host Checker or IMV and that explains why the client machine does not meet the Host Checker policy requirements.”
See, Pulse Policy Secure complete software guide at page 716 of 1100 available at https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fpps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

Configuring General Host Checker Remediation

To specify remediation actions for a Host Checker policy:

1. In the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. Create or enable Host Checker policies.
3. Specify the remediation actions for Host Checker to perform if a computer does not meet the requirements of the current policy:
 - **Enable Custom Instructions**—Enter the instructions to display to the user on the Host Checker remediation page. You can use the following HTML tags to format text and to add links to resources such as policy servers or web sites: `<i>`, ``, `
`, ``, and `<a href>`. For example:

You do not have the latest signature files.

```
<a href="www.company.com">Click here to download the latest signature files.</a>
```
 - **Kill Processes**—On each line, enter the name of one or more processes to kill if the computer does not meet the policy requirements. You can include an optional MD5 checksum for the process. (You cannot use wildcards in the process name.) For example:


```
keylogger.exe
MD5: 6A7DFAF12C3183B56C44E89B12DBEF56
```
 - **Delete Files**—Enter the names of files to delete if the user's computer does not meet the policy requirements. (You cannot use wildcards in the file name.) Enter one filename per line. For example:


```
c:\temp\bad-file.txt
/temp/bad-file.txt
```
 - **Send reason strings**—Select this option to display a message to users (called a reason string) that is returned by Host Checker or IMV and that explains why the client machine does not meet the Host Checker policy requirements. This option applies to predefined

See also page 84 of 1100 available at https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

- **DNS server**—Connect if the DNS server associated with the endpoint's network properties is (or is not) set to a certain value or set of values. Specify the DNS server IP address in the IP address box. Also specify a network interface on which the condition must be satisfied:
 - **Physical**—The condition must be satisfied on the physical interfaces on the endpoint.
 - **Pulse Secure Client**—The condition must be satisfied on the virtual interface that Pulse Secure client creates when it establishes a connection.
 - **Any**—Use any interface.
- **Resolve address**—Connect if the configured hostname or set of hostnames is (or is not) resolvable by the endpoint to a particular IP address. Specify the hostname in the DNS name box and the IP address or addresses in the IP address box. Also specify a network interface on which the condition must be satisfied.



NOTE: The Pulse Secure client software evaluates IP and DNS policies on network interface changes. DNS lookups occur on DNS configuration changes or when the time-to-live setting (10 minutes) expires for a particular host record. If Pulse cannot resolve the host for any reason, it polls the configured DNS server list every 30 seconds. If the host had been resolved successfully previously and the time-to-live timer has not expired, the polling continues until the timer expires. If the host had not been resolved successfully previously, the resolution attempt fails immediately.

and permitting the first host to communicate with the remediation host:

The Accused Instrumentalities permit the client to communicate with the remediation host. This enables limited access to the network to access data and resources to enable the client to attain an acceptable state. *See, e.g.*, the TNC Specification at pages 27-28 available at https://trustedcomputinggroup.org/wp-content/uploads/TNC_Architecture_v1_5_r4.pdf.

VLAN Containment: VLAN containment permits the AR to access the network in a limited fashion. Typically the primary purpose of the limited access is to allow the AR to access on-line sources of remediation data (e.g. virus definition file updates, worm removal software, software patches, etc). In some cases, no remediation is offered and the AR is instead offered access to limited services, in such a fashion as to limit the potential for impact to the network or other attached hosts. RADIUS provisions VLAN containment using the Tunnel-Private-Group-ID attribute, as specified in RFC3580 [21].

5.4 Remediation Phase

The TNC Architecture in Figure 4 accommodates a number of schemes for remediation. The intent of remediation is generally universal, namely that of performing updates to the software and firmware of the AR to help it comply with the current network policy.

The general aim of remediation is to bring the AR up to date in all integrity-related information, as defined by the current policy for authorization. Examples include OS patches, AV updates, firmware upgrades, etc. Section 5.5 below discusses the TNC approach to remediation in further detail.

After remediation has been completed, the IMCs can ask the TNCC to retry the Integrity Check Handshake, which results in another Assessment Phase. This second phase may be shorter than the first since the IMCs may be able to send only the data that has changed (if supported by the IMVs).

In addition, *see, e.g.*, pages 13 and 15 of the TNC IF-IMV Specification available at https://www.trustedcomputinggroup.org/wp-content/uploads/TNC_IFIMV_v1_0_r3.pdf.

2.6.3 Remediation and Handshake Retry

In several cases, it is useful to retry an Integrity Check Handshake. First, an endpoint may be isolated until remediation is complete. Once remediation is complete, an IMC can inform the TNCC of this fact and suggest that the TNCC retry the Integrity Check Handshake. Second, a TNCS can initiate a retry of an Integrity Check Handshake (if the TNCS or IMV policies change or as a periodic recheck). Third, an IMC or IMV can request a handshake retry in response to a condition detected by the IMC or IMV (suspicious activity, for instance). In any case, it's generally desirable (but not always possible) to reuse state established by the earlier handshake and to avoid disrupting network connectivity during the handshake retry.

To support handshake retries, the TNCS MAY maintain a network connection ID after an Integrity Check Handshake has been completed. This network connection ID can then be used by the TNCS to inform IMVs that it is retrying the handshake or by an IMV to request a retry (due to policy change or another reason).

Handshake retry may not always be possible due to limitations in the TNCC, NAR, PEP, or other entities. In other cases, retry may require disrupting network connectivity. For these reasons, IF-IMV supports handshake retry and requires IMVs to handle handshake retries (which is usually trivial) but does not require TNCSs to honor IMV requests for handshake retry. In fact, IF-IMV requires an IMV to provide information about the reason for requesting handshake retry so that the TNCS can decide whether it wants to retry (which may disrupt network access).

Note that remediation instructions are delivered from IMVs to IMCs through standard IMV-IMC messages (see section 2.6.4, "Message Delivery"). There is no special support in IF-IMV for this feature. IMVs SHOULD send remediation instructions to IMCs before returning an IMV Action Recommendation and IMV Evaluation Result to the TNCS so the instructions are delivered before the handshake is completed.

2.6.7 IMV Action Recommendation

One of the assumptions of the TNC architectural model is that IF-IMV provides a means for IMVs to recommend action information to the TNCS, so that isolation can properly be supported on the network. The TNCS then will combine these IMV Action Recommendations using some logic (defined by the TNCS implementers) to come up with an overall TNCS Action Recommendation. Note that the TNCS may choose to ignore any IMV Action Recommendation, but each IMV must be able to recommend an action. Potential choices for IMV Action Recommendations include: recommend full (normal) access; recommend isolation (limited or quarantined access); and recommend denial (no access). The mandatory function `TNC_TNCS_ProvideRecommendation` is the mechanism within IF-IMV for an IMV to indicate its IMV Action Recommendation.

Communication with the remediation host is essential to enable remediation. This allows the host to improve its integrity and subsequently gain access to the protected network. *See id.* at page 57.

20. If the IMCs need to perform remediation, they perform that remediation. Then they continue with Handshake Retry After Remediation. If no remediation was needed, the use case ends here.

7.5 Handshake Retry After Remediation

1. When an IMC completes remediation, it informs the TNCC that its remediation is complete and requests a retry of the Integrity Check Handshake through IF-IMC.
2. The TNCC decides whether to initiate an Integrity Check Handshake retry (possibly depending on policy, user interaction, etc.). Depending on limitations of the NAR, the TNCC may need to disconnect from the network and reconnect to retry the Integrity Check Handshake. In that case (especially if the previous handshake resulted in full access), it may decide to skip the handshake retry. However, in many cases the TNCC will be able to retry the handshake without disrupting network access. It may even be able to retain the state established in the earlier handshake. If the TNCC decides to skip the retry, the use case ends here.
3. The TNCC initiates a retry of the handshake. Skip to step 1, 3, or 5 of the Network Connect section above, depending on which steps are needed to initiate the retry.

Defendant's use of the remediation server is described in the Pulse Policy Secure complete software guide at page 674 of 1100 available at

https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

Host Checker and Trusted Network Computing

Host Checker is a client-side agent that performs endpoint health and security checks for hosts that attempt to connect to Policy Secure. Host Checker is based on open standards defined by the Trusted Computing Group (TCG).

Trusted Network Connect (TNC) is a subgroup of the TCG that created an architecture and set of standards for verifying endpoint integrity and policy compliance during or after a network access request. Many of the TCG members participated in the definition and specification of the TNC architecture. The TNC defined several standard interfaces that enable components from different vendors to securely operate together. The TNC architecture is designed to build on established standards and technologies, such as 802.1X, RADIUS, IPsec, EAP, and TLS/SSL. For more information about TNC, see www.trustedcomputinggroup.org. Using technology based on the TNC architecture and standards, the Host Checker component of the Unified Access Control solution provides a comprehensive approach to assess the trustworthiness of endpoints.

You can use Host Checker to perform health and security evaluations on endpoints before allowing them to connect to the network and to access protected resources. Host Checker can check for third-party applications, files, process, ports, registry keys, and custom DLLs on hosts. Based on the results of the checks, Host Checker can deny or allow access to protected resources. For example, you can check for virus detection software before allowing a user access to a realm. You can configure policies that launch the software on the user's system, 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. When a user's computer does not meet the requirements you specify, you can configure options that display remediation instructions to users so they can bring their computers into compliance.

See also, Defendant’s description of the configuration of remediation actions when an endpoint does not meet the host policy requirements. Specifically, “Enable Custom Instructions—Enter the instructions to display to the user on the Host Checker remediation page. You can use the following HTML tags to format text and to add links to resources such as policy servers or web sites: `<i>`, ``, `
`, ``, and `<a href>`. For example:

You do not have the latest signature files.

`Click here to download the latest signature files.`” from page 716 of 1100 of Defendant’s Pulse Policy Secure complete software guide available at

https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

Configuring General Host Checker Remediation

To specify remediation actions for a Host Checker policy:

1. In the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. Create or enable Host Checker policies.
3. Specify the remediation actions for Host Checker to perform if a computer does not meet the requirements of the current policy:
 - **Enable Custom Instructions**—Enter the instructions to display to the user on the Host Checker remediation page. You can use the following HTML tags to format text and to add links to resources such as policy servers or web sites: `<i>`, ``, `
`, ``, and `<a href>`. For example:

You do not have the latest signature files.

`Click here to download the latest signature files.`
 - **Kill Processes**—On each line, enter the name of one or more processes to kill if the computer does not meet the policy requirements. You can include an optional MD5 checksum for the process. (You cannot use wildcards in the process name.) For example:

keylogger.exe

MD5: 6A7DFAF12C3183B56C44E89B12DBEF56
 - **Delete Files**—Enter the names of files to delete if the user’s computer does not meet the policy requirements. (You cannot use wildcards in the file name.) Enter one filename per line. For example:

c:\temp\bad-file.txt

/temp/bad-file.txt
 - **Send reason strings**—Select this option to display a message to users (called a reason string) that is returned by Host Checker or IMV and that explains why the client machine does not meet the Host Checker policy requirements. This option applies to predefined

21. On information and belief, the Accused Instrumentalities are used, marketed, sold, or otherwise provided by or for Defendant's partners, clients, customers and end users across the country and in this District.

22. Defendant was made aware of the '705 patent and its infringement thereof at least as early as the filing date of this Complaint.

23. Upon information and belief, since at least the filing date of this Complaint, Defendant induced and continues to induce others to infringe at least one claim of the '705 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Defendant's partners, clients, customers, and end users, whose use of the Accused Instrumentalities constitutes direct infringement of at least one claim of the '705 patent.

24. Defendant's actions that aid and abet others, such as its partners, customers, clients, and end users, to infringe include, since at least the filing date of this Complaint, advertising and distributing the Accused Instrumentalities and providing instruction materials, training, and services related to the Accused Instrumentalities. On information and belief, Defendant has, since at least the filing date of this Complaint, engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Defendant has had actual knowledge of the '705 patent and knowledge that its acts were inducing infringement of the '705 patent since at least the filing date of this Complaint.

25. Upon information and belief, Defendant is liable as a contributory infringer of the '705 patent under 35 U.S.C. § 271(c) by offering to sell, selling, and/or importing into the United States the Accused Instrumentalities for use in practicing the '705 patent knowing, at least as early as the filing date of this Complaint, that the Accused Instrumentalities are especially made

or adapted for use in an infringement of the '705 patent. The Accused Instrumentalities include a material component for use in practicing the '705 patent and are not a staple article or commodity of commerce suitable for substantial non-infringing use.

26. Plaintiff has been harmed by Defendant's infringing activities.

COUNT II – INFRINGEMENT OF U.S. PATENT NO. 9,516,048

27. The allegations set forth in the foregoing paragraphs 1 through 26 are incorporated into this Second Count.

28. On December 6, 2016, U.S. Patent No. 9,516,048 ("the '048 patent"), entitled "Contagion Isolation and Inoculation via Quarantine," was duly and legally issued by the United States Patent and Trademark Office. A true and correct copy of the '048 patent is attached as Exhibit 2.

29. The claims of the '048 patent are directed to technical solutions to technical problems related to network security. Network security is a tremendous concern for modern enterprises, since a breach of an internal network can have severe repercussions, including major financial losses, data theft, disclosure of sensitive information, network disruptions, and data corruption—any of which could have devastating consequences to a business, at any scale. While a network security appliance can be extremely adept at keeping unwanted external intrusions into the network out, it is well known that perhaps the most exploitable component of a network is the human element. IT professionals may be able to keep on-site systems secure and up to date, but a worker's personal laptop or mobile device is a significant security risk that would allow attackers (or just unwanted malware) access into a business's network, bypassing IT security measures. The technology of the '048 Patent closes this loophole by verifying that any device attempting to access a company's network meets the company's standards for network security and will not introduce dangerous programs into the company's network. Through this

technology, as implemented by companies including the Defendant, businesses can increase security of vulnerable elements of their networks.

30. The claims of the '048 patent do not merely recite the performance of some business practice known from the pre-Internet world along with the requirement to perform it on the Internet. Instead, the claims of the '048 patent are rooted in computerized network security technology in order to overcome problems specifically arising in the realm of computerized network security.

31. The claims of the '048 patent recite subject matter that is not merely the routine or conventional use of network infrastructure. Instead, the claimed inventions are directed to particularized methods of assessing and responding to an external network access request such that network integrity is maintained. The '048 patent claims specify how a secure network can assess and respond to an external network access request without jeopardizing network integrity.

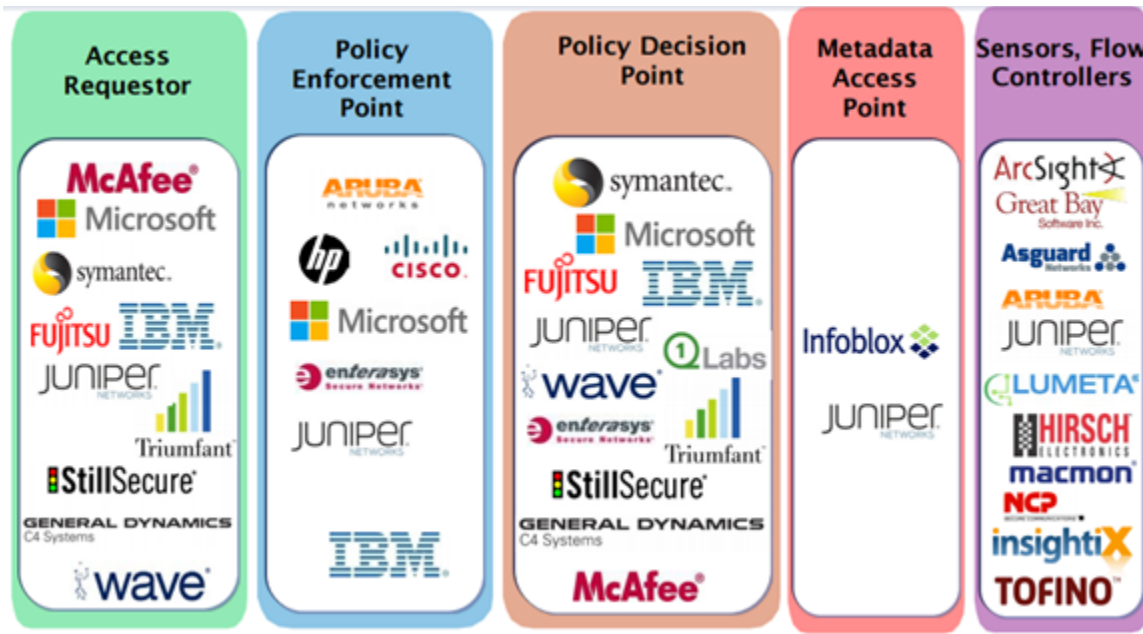
32. The claims of the '048 patent do not preempt all ways of using network security technology, do not preempt any other well-known or prior technology, and do not impermissibly block future developments.

33. Each claim of the '048 patent recites a combination of elements that amounts to significantly more than a patent on an ineligible concept.

34. Plaintiff is the assignee and owner of the right, title and interest in and to the '048 patent, including the right to assert all causes of action arising under said patent and the right to any remedies for infringement of said patent.

35. Upon information and belief, Defendant has directly infringed and continues to directly infringe at least claim 1 of the '048 patent by making, using, offering for sale, selling, and/or importing the Accused Instrumentalities.

36. On information and belief, the Accused Instrumentalities leverage Trusted Network Connect (or Trusted Network Communications) (TNC) open architecture promulgated by the Trusted Network Connect Work Group of the Trusted Computing Group (TCG). *See, e.g.*, TCG Trusted Network Communications TNC Architecture for Interoperability, Specification Version 1.5, Revision 4, published May 7, 2012 (hereinafter the “TNC Specification”) at page 9, available at https://trustedcomputinggroup.org/wp-content/uploads/TNC_Architecture_v1_5_r4.pdf. On information and belief, Defendant adopted the TNC standards at least as of 2013. *See, e.g.*, Trusted Computing Group’s Trusted Network Connect Standards for Network Security presentation available at <https://trustedcomputinggroup.org/wp-content/uploads/TNC-Briefing-2013-12-10.pdf>.



See also, Pulse Policy Secure Datasheet at page 7 available at https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjus5a-pcfVAhUHNI8KHfIeCJMQFggnMAA&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Fdatasets%2F1474%2FPulseSecure_Datasheet_PolicySecure_160224.pdf&usg=AFQjCNFw38iOU6ovDU9DCR6fFEXQYFjTbw (“Adopts and provides strong support for the TCG’s TNC open standards for network access control and security”)

Table 4: Standards-Based, Interoperable Access Control

Feature	Feature Description	Benefits
TNC open standards support, including IF-MAP support and Windows SOH and embedded NAP Agent support (optional)	<ul style="list-style-type: none"> Adopts and provides strong support for the TCG's TNC open standards for network access control and security. Adopts the TNC's open standard IF-MAP, enabling integration with third-party network and security devices, including devices that collect and (through IF-MAP) share information on the state and status of a network, user, or device. Pulse Policy Secure-enabled PSA or MAG Series Appliance can serve as Metadata Access Point (MAP) servers, enabling collected data to be used in formulating policies and appropriate access actions. Through the TNC SOH standard, leverages preinstalled Windows 10, Microsoft Windows 8.1, Windows 8, Windows RT, Windows 7, Windows Vista, clients for access control with the Pulse Access Control Service, allowing use of the Windows Security Center (WSC) SOH in access control decisions. 	<ul style="list-style-type: none"> Empowers organizations to select endpoint and network security solutions that meet their needs without concern for interoperability. Enables ease of deployment, leading to faster ROI. Integrates existing, third-party network and security devices into the access control platform. Streamlines client deployment, simplifying access control rollout and implementation.

37. Upon information and belief, Defendant makes, uses, offers for sale, sells, and/or imports the Accused Instrumentalities. *See, e.g.*, Defendant's description of "Pulse Policy Secure Solution Overview" disclosing the following: "Policy Secure solution coordinates network security compliance and provides the control required to support network applications, manage network use, and reduce threats from unauthorized users and compromised host machines attempting to access the network. You configure rules in Host Checker policies to specify the minimum criteria for the security compliance of host machines that are allowed to enter the network." from page 44 of 1100 of Defendant's Pulse Policy Secure complete software guide available at https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw.

Pulse Policy Secure Solution Overview

The Pulse Policy Secure solution provides a mechanism for authenticating users and assessing the health of their host machines to control network access.

Policy Secure solution coordinates network security compliance and provides the control required to support network applications, manage network use, and reduce threats from unauthorized users and compromised host machines attempting to access the network.

You configure rules in Host Checker policies to specify the minimum criteria for the security compliance of host machines that are allowed to enter the network.

The policies that you create control access for users, the client or agent that users access the network with, and the host machine or endpoint on which the clients run.

38. Claim 1 of the '048 patent recites a method, comprising: detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network, wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host, receiving a response, and determining whether the response includes a valid digitally signed attestation of cleanliness, wherein the valid digitally signed attestation of cleanliness includes at least one attestation selected from the group consisting of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host; when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network, wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host, determining whether the service request sent by the first host is associated with a remediation request, and when it is determined that the service request sent by the first host is not associated with a remediation request, serving a quarantine notification page that provides remediation information to the first host if the service request sent

by the first host comprises a web server request, wherein serving the quarantine notification page to the first host includes re-routing by responding to the service request sent by the first host with a redirect that causes a browser on the first host to be directed to a quarantine server configured to serve the quarantine notification page; and permitting the first host to communicate with a remediation host configured to provide data usable to remedy the insecure condition.

39. As demonstrated in the exemplary images and text below, Defendant has directly infringed and continues to directly infringe at least claim 1 of the '048 patent by making, using, offering for sale, selling, and/or importing the Accused Instrumentalities, which perform a method, comprising: detecting an insecure condition on a first host that has connected or is attempting to connect to a protected network, wherein detecting the insecure condition includes contacting a trusted computing base associated with a trusted platform module within the first host:

The Accused Instrumentalities detect when an Access Requestor (AR) client (a first host) is attempting to connect to the protected network, but has not yet passed the integrity-verification (thus is in insecure condition). *See, e.g.*, the TNC Specification at pages 20 and 26 available at https://trustedcomputinggroup.org/wp-content/uploads/TNC_Architecture_v1_5_r4.pdf.

When a network connection attempt is triggered (automatically or by user request), the NAR at the AR (client) initiates a connection request at the link and network layers.

Isolation: If the AR has been authenticated and is recognized to be one that has some privileges on the network but has not passed the integrity-verification by the IMV, the PDP may return instructions to the PEP to redirect the AR to an isolation environment where the AR can obtain integrity-related updates.

As shown in the figure below, detecting the insecure condition includes using the PTS (Platform Trust Service) Protocol to contact a trusted computing base, the Policy Decision Point (PDP), to use the PTS Integrity Measurement Collector of the first host. *See*, the TCG Attestation PTS Protocol: Binding to TNC IF-M Specification, Version 1.0, Revision 28, published August 24, 2011 (hereinafter "IFM PTS") at page 10 available at https://www.trustedcomputinggroup.org/wp-content/uploads/IFM_PTS_v1_0_r28.pdf.

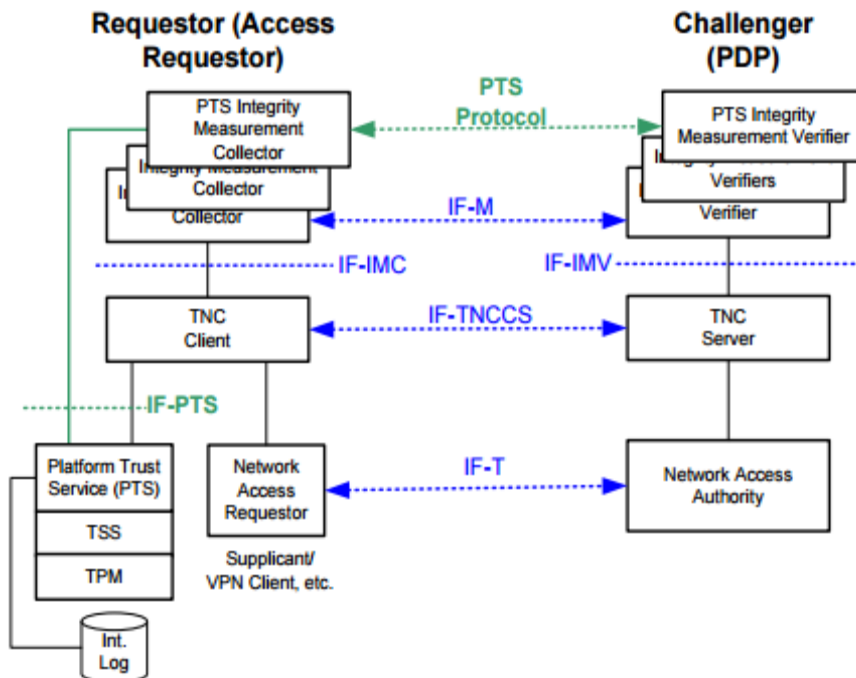


Figure 1: PTS Protocol Integration with TNC Architecture

Notice that the TNC Architecture contains several layered protocols (IF-T, IF-TNCCS and IF-M). The PTS Protocol will be carried within the payloads of the IF-M protocol, so would layer hierarchically on top of the IF-M protocol. The PTS Protocol operates between the PTS-IMC and PTS-IMV to enable PTS-based attestation leveraging the underlying TPM. The PTS-IMC uses a local IPC channel to the PTS (discussed in the IF-PTS specification) to obtain the necessary attestation evidence. Use of the IF-PTS interface and the TSS middleware stack components are optional so implementations might leverage the PTS or TPM in other ways (e.g. the PTS could have other techniques for interacting with a TPM to obtain measurements).

See also, the TCG's Trusted Platform Module Library, Part 1: Architecture, Family 2.0, Level 00 Revision 01.07 draft published March 13, 2014 (hereinafter, "TPM Library") at page 23 available at <https://www.trustedcomputinggroup.org/wp-content/uploads/TPM-Rev-2.0-Part-1-Architecture-01.07-2014-03-13.pdf>.

9.2.2 Trusted Computing Base

A trusted computing base (TCB) is the collection of system resources (hardware and software) that is responsible for maintaining the security policy of the system. An important attribute of a TCB is that it be able to prevent itself from being compromised by any hardware or software that is not part of the TCB.

receiving a response:

The response is then received in the form of "attestation information." See, e.g., the IFM PTS at pages 14-15 available at https://www.trustedcomputinggroup.org/wp-content/uploads/IFM_PTS_v1_0_r28.pdf. (Discussed further below).

and determining whether the response includes a valid digitally signed attestation of cleanliness, wherein the valid digitally signed attestation of cleanliness includes at least one attestation selected from the group consisting of an attestation that the trusted computing base has ascertained that the first host is not infested, and an attestation that the trusted computing base has ascertained the presence of a patch or a patch level associated with a software component on the first host:

If the remote challenger is clean, the “attestation information” will include a signed (and cryptographically verifiable) set of “attestation evidence” which is an attestation of cleanliness. *See, e.g.*, the IFM PTS at pages 14-15 available at https://www.trustedcomputinggroup.org/wp-content/uploads/IFM_PTS_v1_0_r28.pdf.

2.5.2 TPM Quote

This section summarizes how the TPM's ability to quote PCRs provides assurance that the attestation evidence provided is trustworthy and protected from local tampering. The TPM provides a number of shielded storage locations and ordinals to perform transformations on data using the contents of these locations. For example, the TPM houses cryptographic keys that are only usable inside the TPM. One such key is the Attestation Identity Key (AIK) which is limited by the TPM to only be used to sign the contents of the TPM's PCRs (which are protected in TPM shielded storage). On a platform where the PCRs are set to reflect the operational content of the system, this combination of protected PCRs and AIK enable an attestation mechanism to be verifiable by remote parties.

Specifically, a remote challenger can request attestation information about a system and obtain an AIK signed set of “attestation evidence” that is cryptographically verifiable as having been generated using a TPM-resident AIK and PCRs. By empowering the remote challenger to be able to retrieve this signed set of PCRs with proof that they were resident inside a TPM, this mechanism allows the challenger to have confidence that it can determine (potentially using the measurement log and policy) what software has been run on the endpoint without fear of being spoofed by malware.

The attestation information provided by the Accused Instrumentalities includes measurements of the client's integrity (ascertained that the first host is not infested) and the software versions (the presence of a patch or a patch level associated with a software component on the first host). *See, e.g.*, the TNC Specification at page 15 available at https://trustedcomputinggroup.org/wp-content/uploads/TNC_Architecture_v1_5_r4.pdf.

Integrity Measurement Collector (IMC): The IMC function is a software component that runs on an AR, measuring security aspects of the AR's integrity. Examples include the Anti-Virus parameters on the Access Requestor, Personal Firewall status, software versions, and other security aspects of the AR. Note that the TNC Architecture is designed for multiple IMCs to interact with a single (or multiple) TNC Client/TNC Server, thereby allowing customers to deploy complex integrity policies involving a range of vendors products.

This process is performed when clients attempt to connect to the protected network, to prevent outside infections from getting in and to ensure the integrity has not fallen below expectations—such as new updates not being installed—since a previous connection. A Use Case Walkthrough provided by the Trusted

Computing Group illustrates a standard procedure for new connection attempts. See pages 55-59 of the TCG TNC IF-IMV Specification, Version 1.0, Revision 3, published May 3, 2005 (hereafter “TNC IF-IMV Specification”) available at https://www.trustedcomputinggroup.org/wp-content/uploads/TNC_IFIMV_v1_0_r3.pdf.

7.4 Network Connect

1. The endpoint's NAR attempts to connect to a network protected by a PEP, thus triggering an Integrity Check Handshake. There are other ways that an Integrity Check Handshake can be triggered, but this will probably be the most common. For those other ways, the next few steps may be significantly different.
2. The PEP sends a network access decision request to the PDP (NAA or TNCS). Depending on configuration, the PEP may contact the NAA first or the TNCS. The ordering of user authentication, platform authentication, and integrity check is also subject to configuration. Here we present what will probably be the most common order: first user authentication, then platform authentication, then integrity check.
3. The NAA performs user authentication with the NAR. Based on the NAA's policy, the user identity established through this process may be used to make immediate access decisions (like deny). If an immediate access decision has been made, skip to step 16. User authentication may also involve having the NAR authenticate the NAA.
4. The NAA informs the TNCS of the connection request, providing the user identity and other useful info (service requested, etc.).
5. The TNCS performs platform authentication with the TNCC, if required by TNCS policy. This includes verifying the IMC hashes collected during TNCC Setup. If an immediate access decision has been made, skip to step 16. Platform authentication may be mutual so the TNCC can be sure it's talking to a secure server.
6. The TNCC uses IF-IMC to fetch IMC messages.
7. The TNCS uses IF-IMV to inform each IMV that an Integrity Check Handshake has started. **[IF-IMV] If this is a new network connection, the TNCS calls TNC_IMV_NotifyConnectionChange with the newState parameter set to TNC_CONNECTION_STATE_CREATE to indicate that a new network connection has been created. Then the TNCS calls TNC_IMV_NotifyConnectionChange with the newState parameter set to TNC_CONNECTION_STATE_HANDSHAKE.**

8. The TNCC passes the IMC messages to the TNCS. This and all other TNCC-TNCS communications can be sent directly but they will often be relayed through one or more of the NAR, PEP, and NAA.
9. The TNCS passes each IMC message to the matching IMV or IMVs through IF-IMV (using message types associated with the IMC messages to find the right IMV). If there are no IMC messages, skip to step 13. **[IF-IMV] The TNCS delivers the IMC messages to the IMVs by calling `TNC_IMV_ReceiveMessage`. The IMVs may call `TNC_TNCS_SendMessage` before returning from `TNC_IMV_ReceiveMessage` if they want to send a response. When the TNCS has delivered all the IMC messages to the IMVs, it calls `TNC_IMV_BatchEnding` to inform them of this fact. The IMVs may call `TNC_TNCS_SendMessage` before returning from `TNC_IMV_BatchEnding` if they want to send a message to an IMV.**
10. Each IMV analyzes the IMC messages. If an IMV needs to exchange more messages (including remediation instructions) with an IMC, it provides a message to the TNCS and continues with step 11. If an IMV is ready to decide on an IMV Action Recommendation and IMV Evaluation Result, it gives this result to the TNCS through IF-IMV. If there are no more messages to be sent to the IMC from any of the IMVs, skip to step 13. **[IF-IMV] As described in the previous step, IMVs send messages by calling `TNC_TNCS_SendMessage` before returning from `TNC_IMV_ReceiveMessage` and `TNC_IMV_BatchEnding`. IMVs give their results to the TNCS by calling `TNC_TNCS_ProvideRecommendation` at any time.**
11. The TNCS sends the messages from the IMVs to the TNCC.
12. The TNCC sends the IMV messages on to the IMCs through IF-IMC so they can process the messages and respond. Skip to step 8.
13. If there are any IMVs that have not given an IMV Action Recommendation to the TNCS, they are prompted to do so through IF-IMV. **[IF-IMV] The TNCS gives this prompt by calling `TNC_IMV_SolicitRecommendation`. The IMVs provide their recommendations by calling `TNC_TNCS_ProvideRecommendation`.**
14. The TNCS considers the IMV Action Recommendations supplied by the IMVs and uses an integrity check combining policy to decide what its TNCS Action Recommendation should be.
15. The TNCS sends its TNCS Action Recommendation to the NAA. The NAA may ignore or modify this recommendation based on its policies but will typically abide by it.
16. The NAA sends a copy of its final network access decision response to the TNCS. The TNCS may send a copy of the network access decision to the TNCC. The TNCS also informs the IMVs of the network access decision response via IF-IMV. **[IF-IMV] The TNCS calls `TNC_IMV_NotifyConnectionChange` with the `newState` parameter set to `TNC_CONNECTION_STATE_ACCESS_ALLOWED`, `TNC_CONNECTION_STATE_ACCESS_ISOLATED`, or `TNC_CONNECTION_STATE_ACCESS_NONE`.**
17. The NAA sends its network access decision response to the PEP.
18. The PEP implements the network access decision response. During this process, the NAR is typically informed of the decision. The TNCC may be informed by the NAR or may discover that a new network has come up.
19. If step 6 was not executed, the network connect process is complete. Otherwise, the TNCC informs the IMCs of the network access decision response via IF-IMC.
20. If the IMCs need to perform remediation, they perform that remediation. Then they continue with Handshake Retry After Remediation. If no remediation was needed, the use case ends here.

The Accused Instrumentalities perform the TNC process. *See, e.g.,* Pulse Policy Secure Datasheet at page 5 available at <https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=>

0ahUKEwjus5a-
pcfVAhUHNI8KHfIeCJMqFggnMAA&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Fdatasets%2F1474%2FPulseSecure_Datasheet_PolicySecure_160224.pdf&usg=AFQjCNFw38iOU6ovDU9DCR6fFEXQYfjTbw.

Automated patch assessment checks and remediation (optional)	<ul style="list-style-type: none"> • Can tie access directly to the presence or absence of specific hot fixes for defined operating systems and applications, and performs role-based, predefined patch management checks according to the severity level of vulnerabilities. • Installed Systems Management Server (SMS) and/or System Center Configuration Manager (SCCM) 2007 can be leveraged to automatically check for patch updates, quarantining, remediating, and providing authorized network access once a device has been remediated. 	<ul style="list-style-type: none"> • Enables enhanced, granular endpoint device health and security state assessments. • Minimizes user interaction and downtime through automatic remediation and management of patches for endpoint devices, reducing help desk calls.
--	---	--

See also, Defendant’s description of “Host Checker and Trusted Network Computing” disclosing the following: “You can use Host Checker to perform health and security evaluations on endpoints before allowing them to connect to the network and to access protected resources. Host Checker can check for third-party applications, files, process, ports, registry keys, and custom DLLs on hosts. Based on the results of the checks, Host Checker can deny or allow access to protected resources. For example, you can check for virus detection software before allowing a user access to a realm. You can configure policies that launch the software on the user’s system, map the user to roles based on individual policies defined in your own DLL,” from page 674 of 1100 of Defendant’s Pulse Policy Secure complete software guide available at https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwj76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpiZgoMFvb0wDxqOK1_zZw

Host Checker and Trusted Network Computing

Host Checker is a client-side agent that performs endpoint health and security checks for hosts that attempt to connect to Policy Secure. Host Checker is based on open standards defined by the Trusted Computing Group (TCG).

Trusted Network Connect (TNC) is a subgroup of the TCG that created an architecture and set of standards for verifying endpoint integrity and policy compliance during or after a network access request. Many of the TCG members participated in the definition and specification of the TNC architecture. The TNC defined several standard interfaces that enable components from different vendors to securely operate together. The TNC architecture is designed to build on established standards and technologies, such as 802.1X, RADIUS, IPsec, EAP, and TLS/SSL. For more information about TNC, see www.trustedcomputinggroup.org. Using technology based on the TNC architecture and standards, the Host Checker component of the Unified Access Control solution provides a comprehensive approach to assess the trustworthiness of endpoints.

You can use Host Checker to perform health and security evaluations on endpoints before allowing them to connect to the network and to access protected resources. Host Checker can check for third-party applications, files, process, ports, registry keys, and custom DLLs on hosts. Based on the results of the checks, Host Checker can deny or allow access to protected resources. For example, you can check for virus detection software before allowing a user access to a realm. You can configure policies that launch the software on the user's system, 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. When a user's computer does not meet the requirements you specify, you can configure options that display remediation instructions to users so they can bring their computers into compliance.

when it is determined that the response does not include a valid digitally signed attestation of cleanliness, quarantining the first host, including by preventing the first host from sending data to one or more other hosts associated with the protected network:

The Accused Instrumentalities use the Assessment Phase to determine the client's integrity status. Once this determination is made, the Integrity Measurement Verifier (IMV) "can make one of three IMV Action-Recommendations (Allow, Isolate or Block)." *See, e.g.*, the TNC Specification at page 27 available at https://trustedcomputinggroup.org/wp-content/uploads/TNC_Architecture_v1_5_r4.pdf.

5.2 Assessment Phase

In the Assessment Phase, the TNC Client reports its current integrity status to the TNC Server. Upon receiving the client integrity status, the IMVs with the aid of the TNCS perform an assessment of the AR based on the set of policies defined by the network administrator. The IMV can make one of three IMV Action-Recommendations (Allow, Isolate or Block) or it can make no recommendation.

A client AR that fails integrity variation is isolated (quarantined) onto an "Isolation Network." "This protects the AR from the full network and vice versa, preventing the spread of viruses and worms."

5.3 Isolation Phase

An important tool in the effort to remediate ARs that fail integrity verification is the isolation of that AR to a separate network – referred to here as the Isolation Network – in order to provide remediation services to the AR. This protects the AR from the full network and vice versa, preventing the spread of viruses and worms. There are a number of technical approaches today to achieve network isolation for the AR. Two of these are as follows:

- (a) *VLAN Containment*: VLAN containment permits the AR to access the network in a limited fashion. Typically the primary purpose of the limited access is to allow the AR to access on-line sources of remediation data (e.g. virus definition file updates, worm removal software, software patches, etc). In some cases, no remediation is offered and the AR is instead offered access to limited services, in such a fashion as to limit the potential for impact to the network or other attached hosts. RADIUS provisions VLAN containment using the Tunnel-Private-Group-ID attribute, as specified in RFC3580 [21].
- (b) *IP Filters*: In the case of IP filters, the PEP is configured with a set of filters which defines network locations reachable by the isolated AR. Packets from the AR destined to other network locations are simply discarded by the PEP. RADIUS selects filter rules for application to a network access session using the Filter-ID attribute (see RFC2865 and RFC3580) [21].

wherein preventing the first host from sending data to one or more other hosts associated with the protected network includes receiving a service request sent by the first host, determining whether the service request sent by the first host is associated with a remediation request:

This isolation network provides “limited or quarantined access” which prevents the host from sending data to other hosts associated with the protected network. This will require “remediation instructions” be provided in place of standard responses to service requests. *See, e.g.*, pages 13-15 of the TNC IF-IMV Specification available at https://www.trustedcomputinggroup.org/wp-content/uploads/TNC_IFIMV_v1_0_r3.pdf.

2.6.3 Remediation and Handshake Retry

In several cases, it is useful to retry an Integrity Check Handshake. First, an endpoint may be isolated until remediation is complete. Once remediation is complete, an IMC can inform the TNCC of this fact and suggest that the TNCC retry the Integrity Check Handshake. Second, a TNCS can initiate a retry of an Integrity Check Handshake (if the TNCS or IMV policies change or as a periodic recheck). Third, an IMC or IMV can request a handshake retry in response to a condition detected by the IMC or IMV (suspicious activity, for instance). In any case, it's generally desirable (but not always possible) to reuse state established by the earlier handshake and to avoid disrupting network connectivity during the handshake retry.

To support handshake retries, the TNCS MAY maintain a network connection ID after an Integrity Check Handshake has been completed. This network connection ID can then be used by the TNCS to inform IMVs that it is retrying the handshake or by an IMV to request a retry (due to policy change or another reason).

Handshake retry may not always be possible due to limitations in the TNCC, NAR, PEP, or other entities. In other cases, retry may require disrupting network connectivity. For these reasons, IF-IMV supports handshake retry and requires IMVs to handle handshake retries (which is usually trivial) but does not require TNCSs to honor IMV requests for handshake retry. In fact, IF-IMV requires an IMV to provide information about the reason for requesting handshake retry so that the TNCS can decide whether it wants to retry (which may disrupt network access).

Note that remediation instructions are delivered from IMVs to IMCs through standard IMV-IMC messages (see section 2.6.4, "Message Delivery"). There is no special support in IF-IMV for this feature. IMVs SHOULD send remediation instructions to IMCs before returning an IMV Action Recommendation and IMV Evaluation Result to the TNCS so the instructions are delivered before the handshake is completed.

2.6.4 Message Delivery

One of the critical functions of the TNC architecture is conveying messages between IMCs and IMVs. Each message sent in this way consists of a message body, a message type, and a recipient type.

The message body is a sequence of octets (bytes). The TNCC and TNCS SHOULD NOT parse or interpret the message body. They only deliver it as described below. Interpretation of the message body is left to the ultimate recipients of the message, the IMCs or IMVs. A zero length message is perfectly valid and MUST be properly delivered by the TNCC and TNCS just as any other IMC-IMV message would be.

The message type is a four octet number that uniquely identifies the format and semantics of the message. The method used to ensure the uniqueness of message types while providing for vendor extensions is described below.

The recipient type is simply a flag indicating whether the message should be delivered to IMVs or IMCs. Messages sent by IMCs are delivered to IMVs and vice versa. All messages sent by an IMV through IF-IMV have a recipient type of IMC. All messages received by an IMV through IF-IMV have a recipient type of IMV. The recipient type does not show up in IF-IMC or IF-IMV, but it helps in explaining message routing.

The routing and delivery of messages is governed by message type and recipient type. Each IMC and IMV indicates through IF-IMC and IF-IMV which message types it wants to receive. The TNCC and TNCS are then responsible for ensuring that any message sent during an Integrity Check Handshake is delivered to all recipients that have a recipient type matching the message's recipient type and that have indicated the wish to receive messages whose type matches the message's message type. If no recipient has indicated a wish to receive a particular message type, the TNCC and TNCS can handle these messages as they like: ignore, log, etc.

WARNING: The message routing and delivery algorithm just described is not a one-to-one model. A single message may be received by several recipients (for example, two IMVs from a single

vendor, two copies of an IMC, or nosy IMVs that monitor all messages). If several of these recipients respond, this may confuse the original sender. IMCs and IMVs MUST work properly in this environment. They MUST NOT assume that only one party will receive and/or respond to a message.

IF-IMV allows an IMV to send and receive messages using this messaging system. Note that this system should not be used to send large amounts of data. The messages will often be sent through PPP or similar protocols that do not include congestion control and are not well suited to bulk data transfer. If an IMC needs to download a patch (for instance), the IMV should indicate this by reference in the remediation instructions. The IMC will process those instructions after network access (perhaps isolated) has been established and can then download the patch via HTTP or another appropriate protocol.

All messages sent with `TNC_TNCS_SendMessage` and received with `TNC_IMV_ReceiveMessage` are between the IMC and IMV. The IMV communicates with the TNCS by calling functions (standard and vendor-specific) in the IF-IMV, not by sending messages. The TNCS should not interfere with communications between the IMC and IMVs by consuming or blocking IMC-IMV messages.

A particular example of the message delivery provided by IF-IMV is the communication of remediation instructions from the IMVs through the TNCS to the TNCC/IMCs. This is one application of IMC-IMV message delivery and in all cases follows the normal IMV-IMC communications path. IF-IMV provides support for communicating remediation instructions to an endpoint using this mechanism. Since the normal IMC-IMV communications path is used to communicate remediation instructions, this specification will not address further the details of how remediation itself is done.

2.6.7 IMV Action Recommendation

One of the assumptions of the TNC architectural model is that IF-IMV provides a means for IMVs to recommend action information to the TNCS, so that isolation can properly be supported on the network. The TNCS then will combine these IMV Action Recommendations using some logic (defined by the TNCS implementers) to come up with an overall TNCS Action Recommendation. Note that the TNCS may choose to ignore any IMV Action Recommendation, but each IMV must be able to recommend an action. Potential choices for IMV Action Recommendations include: recommend full (normal) access; recommend isolation (limited or quarantined access); and recommend denial (no access). The mandatory function `TNC_TNCS_ProvideRecommendation` is the mechanism within IF-IMV for an IMV to indicate its IMV Action Recommendation.

and when it is determined that the service request sent by the first host is not associated with a remediation request, serving a quarantine notification page that provides remediation information to the first host if the service request sent by the first host comprises a web server request, wherein serving the quarantine notification page to the first host includes re-routing by responding to the service request sent by the first host with a redirect that causes a browser on the first host to be directed to a quarantine server configured to serve the quarantine notification page:

As an example of Defendant's implementation of this process in the Accused Instrumentalities, see the following explanation of "Host Checker and Trusted Network Computing." Notably, "based on the results of the checks, Host Checker can deny or allow access to protected resources. For example, you can check for virus detection software before allowing a user access to a realm," and "When a

user's computer does not meet the requirements you specify, you can configure options that display remediation instructions to users so they can bring their computers into compliance." This illustrates the use of a quarantine notification page, provided in response when the client issues a web server request or a DNS query ("display remediation instructions"). See, e.g., the Pulse Policy Secure complete software guide at page 674 of 1100 available at https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

Host Checker and Trusted Network Computing

Host Checker is a client-side agent that performs endpoint health and security checks for hosts that attempt to connect to Policy Secure. Host Checker is based on open standards defined by the Trusted Computing Group (TCG).

Trusted Network Connect (TNC) is a subgroup of the TCG that created an architecture and set of standards for verifying endpoint integrity and policy compliance during or after a network access request. Many of the TCG members participated in the definition and specification of the TNC architecture. The TNC defined several standard interfaces that enable components from different vendors to securely operate together. The TNC architecture is designed to build on established standards and technologies, such as 802.1X, RADIUS, IPsec, EAP, and TLS/SSL. For more information about TNC, see www.trustedcomputinggroup.org. Using technology based on the TNC architecture and standards, the Host Checker component of the Unified Access Control solution provides a comprehensive approach to assess the trustworthiness of endpoints.

You can use Host Checker to perform health and security evaluations on endpoints before allowing them to connect to the network and to access protected resources. Host Checker can check for third-party applications, files, process, ports, registry keys, and custom DLLs on hosts. Based on the results of the checks, Host Checker can deny or allow access to protected resources. For example, you can check for virus detection software before allowing a user access to a realm. You can configure policies that launch the software on the user's system, 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. When a user's computer does not meet the requirements you specify, you can configure options that display remediation instructions to users so they can bring their computers into compliance.

See also, Defendant's "Understanding Host Checker Policy Remediation" which describes the Defendant system's ability to "specify general remediation actions for Host Checker to take if an endpoint does not meet the requirements of a policy. For example, you can display a remediation page to the user that contains specific instructions and links to resources to help the user bring their endpoint into compliance with Host Checker policy requirements." For example, "the user might see a remediation page that contains the following custom instructions, a link to resources, and reason strings:

Your computer's security is unsatisfactory.

Your computer does not meet the following security requirements. Follow the instructions below to fix these problems. When you are done click Try Again. If

you Continue without fixing these problems, you may not have access to all of your intranet servers.” from page 714 of 1100 of Defendant’s Pulse Policy Secure complete software guide available at https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

Understanding Host Checker Policy Remediation

This topic describes Host Checker policy remediation. It includes the following information:

- Remediation Options
- Remediation User Experience

Remediation Options

You can specify general remediation actions for Host Checker to take if an endpoint does not meet the requirements of a policy. For example, you can display a remediation page to the user that contains specific instructions and links to resources to help the user bring their endpoint into compliance with Host Checker policy requirements.

You can also include a message to users (called a reason string) that is returned by Host Checker or an IMV and that explains why the client machine does not meet the Host Checker policy requirements.

For example, the user might see a remediation page that contains the following custom instructions, a link to resources, and reason strings:

Your computer’s security is unsatisfactory.

When specifying remediation actions for a Host Checker policy, Defendant systems allow for setting custom instructions to display to the user on the Host Checker remediation page. This is particularly disclosed in the following:

“Enable Custom Instructions—Enter the instructions to display to the user on the Host Checker remediation page. You can use the following HTML tags to format text and to add links to resources such as policy servers or web sites: `<i>`, ``, `
`, ``, and `<a href>`. For example:

You do not have the latest signature files.

`Click here to download the latest signature files.`”

“Send reason strings—Select this option to display a message to users (called a reason string) that is returned by Host Checker or IMV and that explains why the client machine does not meet the Host Checker policy requirements.”

See, Pulse Policy Secure complete software guide at page 716 of 1100 available at <https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent>

%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

Configuring General Host Checker Remediation

To specify remediation actions for a Host Checker policy:

1. In the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. Create or enable Host Checker policies.
3. Specify the remediation actions for Host Checker to perform if a computer does not meet the requirements of the current policy:
 - **Enable Custom Instructions**—Enter the instructions to display to the user on the Host Checker remediation page. You can use the following HTML tags to format text and to add links to resources such as policy servers or web sites: `<i>`, ``, `
`, ``, and `<a href>`. For example:

You do not have the latest signature files.

`Click here to download the latest signature files.`
 - **Kill Processes**—On each line, enter the name of one or more processes to kill if the computer does not meet the policy requirements. You can include an optional MD5 checksum for the process. (You cannot use wildcards in the process name.) For example:

keylogger.exe

MD5: 6A7DFAF12C3183B56C44E89B12DBEF56
 - **Delete Files**—Enter the names of files to delete if the user's computer does not meet the policy requirements. (You cannot use wildcards in the file name.) Enter one filename per line. For example:

c:\temp\bad-file.txt

/temp/bad-file.txt
 - **Send reason strings**—Select this option to display a message to users (called a reason string) that is returned by Host Checker or IMV and that explains why the client machine does not meet the Host Checker policy requirements. This option applies to predefined

See also page 84 of 1100 available at
https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

- **DNS server**—Connect if the DNS server associated with the endpoint's network properties is (or is not) set to a certain value or set of values. Specify the DNS server IP address in the IP address box. Also specify a network interface on which the condition must be satisfied:
 - **Physical**—The condition must be satisfied on the physical interfaces on the endpoint.
 - **Pulse Secure Client**—The condition must be satisfied on the virtual interface that Pulse Secure client creates when it establishes a connection.
 - **Any**—Use any interface.
- **Resolve address**—Connect if the configured hostname or set of hostnames is (or is not) resolvable by the endpoint to a particular IP address. Specify the hostname in the DNS name box and the IP address or addresses in the IP address box. Also specify a network interface on which the condition must be satisfied.



NOTE: The Pulse Secure client software evaluates IP and DNS policies on network interface changes. DNS lookups occur on DNS configuration changes or when the time-to-live setting (10 minutes) expires for a particular host record. If Pulse cannot resolve the host for any reason, it polls the configured DNS server list every 30 seconds. If the host had been resolved successfully previously and the time-to-live timer has not expired, the polling continues until the timer expires. If the host had not been resolved successfully previously, the resolution attempt fails immediately.

and permitting the first host to communicate with a remediation host configured to provide data usable to remedy the insecure condition:

The Accused Instrumentalities permit the client to communicate with the remediation host. This enables limited access to the network to access data and resources to enable the client to attain an acceptable state. *See, e.g.*, the TNC Specification at pages 27-28 available at https://trustedcomputinggroup.org/wp-content/uploads/TNC_Architecture_v1_5_r4.pdf.

VLAN Containment: VLAN containment permits the AR to access the network in a limited fashion. Typically the primary purpose of the limited access is to allow the AR to access on-line sources of remediation data (e.g. virus definition file updates, worm removal software, software patches, etc). In some cases, no remediation is offered and the AR is instead offered access to limited services, in such a fashion as to limit the potential for impact to the network or other attached hosts. RADIUS provisions VLAN containment using the Tunnel-Private-Group-ID attribute, as specified in RFC3580 [21].

5.4 Remediation Phase

The TNC Architecture in Figure 4 accommodates a number of schemes for remediation. The intent of remediation is generally universal, namely that of performing updates to the software and firmware of the AR to help it comply with the current network policy.

The general aim of remediation is to bring the AR up to date in all integrity-related information, as defined by the current policy for authorization. Examples include OS patches, AV updates, firmware upgrades, etc. Section 5.5 below discusses the TNC approach to remediation in further detail.

After remediation has been completed, the IMCs can ask the TNCC to retry the Integrity Check Handshake, which results in another Assessment Phase. This second phase may be shorter than

the first since the IMCs may be able to send only the data that has changed (if supported by the IMVs).

In addition, *see, e.g.*, pages 13 and 15 of the TNC IF-IMV Specification available at https://www.trustedcomputinggroup.org/wp-content/uploads/TNC_IFIMV_v1_0_r3.pdf.

2.6.3 Remediation and Handshake Retry

In several cases, it is useful to retry an Integrity Check Handshake. First, an endpoint may be isolated until remediation is complete. Once remediation is complete, an IMC can inform the TNCC of this fact and suggest that the TNCC retry the Integrity Check Handshake. Second, a TNCS can initiate a retry of an Integrity Check Handshake (if the TNCS or IMV policies change or as a periodic recheck). Third, an IMC or IMV can request a handshake retry in response to a condition detected by the IMC or IMV (suspicious activity, for instance). In any case, it's generally desirable (but not always possible) to reuse state established by the earlier handshake and to avoid disrupting network connectivity during the handshake retry.

To support handshake retries, the TNCS MAY maintain a network connection ID after an Integrity Check Handshake has been completed. This network connection ID can then be used by the TNCS to inform IMVs that it is retrying the handshake or by an IMV to request a retry (due to policy change or another reason).

Handshake retry may not always be possible due to limitations in the TNCC, NAR, PEP, or other entities. In other cases, retry may require disrupting network connectivity. For these reasons, IF-IMV supports handshake retry and requires IMVs to handle handshake retries (which is usually trivial) but does not require TNCSs to honor IMV requests for handshake retry. In fact, IF-IMV requires an IMV to provide information about the reason for requesting handshake retry so that the TNCS can decide whether it wants to retry (which may disrupt network access).

Note that remediation instructions are delivered from IMVs to IMCs through standard IMV-IMC messages (see section 2.6.4, "Message Delivery"). There is no special support in IF-IMV for this feature. IMVs SHOULD send remediation instructions to IMCs before returning an IMV Action Recommendation and IMV Evaluation Result to the TNCS so the instructions are delivered before the handshake is completed.

2.6.7 IMV Action Recommendation

One of the assumptions of the TNC architectural model is that IF-IMV provides a means for IMVs to recommend action information to the TNCS, so that isolation can properly be supported on the network. The TNCS then will combine these IMV Action Recommendations using some logic (defined by the TNCS implementers) to come up with an overall TNCS Action Recommendation. Note that the TNCS may choose to ignore any IMV Action Recommendation, but each IMV must be able to recommend an action. Potential choices for IMV Action Recommendations include: recommend full (normal) access; recommend isolation (limited or quarantined access); and recommend denial (no access). The mandatory function `TNC_TNCS_ProvideRecommendation` is the mechanism within IF-IMV for an IMV to indicate its IMV Action Recommendation.

Communication with the remediation host is essential to enable remediation. This allows the host to improve its integrity and subsequently gain access to the protected network. *See id.* at page 57.

20. If the IMCs need to perform remediation, they perform that remediation. Then they continue with Handshake Retry After Remediation. If no remediation was needed, the use case ends here.

7.5 Handshake Retry After Remediation

1. When an IMC completes remediation, it informs the TNCC that its remediation is complete and requests a retry of the Integrity Check Handshake through IF-IMC.
2. The TNCC decides whether to initiate an Integrity Check Handshake retry (possibly depending on policy, user interaction, etc.). Depending on limitations of the NAR, the TNCC may need to disconnect from the network and reconnect to retry the Integrity Check Handshake. In that case (especially if the previous handshake resulted in full access), it may decide to skip the handshake retry. However, in many cases the TNCC will be able to retry the handshake without disrupting network access. It may even be able to retain the state established in the earlier handshake. If the TNCC decides to skip the retry, the use case ends here.
3. The TNCC initiates a retry of the handshake. Skip to step 1, 3, or 5 of the Network Connect section above, depending on which steps are needed to initiate the retry.

Defendant's use of the remediation server is described in the Pulse Policy Secure complete software guide at page 674 of 1100 available at

https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

Host Checker and Trusted Network Computing

Host Checker is a client-side agent that performs endpoint health and security checks for hosts that attempt to connect to Policy Secure. Host Checker is based on open standards defined by the Trusted Computing Group (TCG).

Trusted Network Connect (TNC) is a subgroup of the TCG that created an architecture and set of standards for verifying endpoint integrity and policy compliance during or after a network access request. Many of the TCG members participated in the definition and specification of the TNC architecture. The TNC defined several standard interfaces that enable components from different vendors to securely operate together. The TNC architecture is designed to build on established standards and technologies, such as 802.1X, RADIUS, IPsec, EAP, and TLS/SSL. For more information about TNC, see www.trustedcomputinggroup.org. Using technology based on the TNC architecture and standards, the Host Checker component of the Unified Access Control solution provides a comprehensive approach to assess the trustworthiness of endpoints.

You can use Host Checker to perform health and security evaluations on endpoints before allowing them to connect to the network and to access protected resources. Host Checker can check for third-party applications, files, process, ports, registry keys, and custom DLLs on hosts. Based on the results of the checks, Host Checker can deny or allow access to protected resources. For example, you can check for virus detection software before allowing a user access to a realm. You can configure policies that launch the software on the user's system, 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. When a user's computer does not meet the requirements you specify, you can configure options that display remediation instructions to users so they can bring their computers into compliance.

See also, Defendant’s description of the configuration of remediation actions when an endpoint does not meet the host policy requirements. Specifically, “Enable Custom Instructions—Enter the instructions to display to the user on the Host Checker remediation page. You can use the following HTML tags to format text and to add links to resources such as policy servers or web sites: `<i>`, ``, `
`, ``, and `<a href>`. For example:

You do not have the latest signature files.

`Click here to download the latest signature files.`” from page 716 of 1100 of Defendant’s Pulse Policy Secure complete software guide available at

https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&uact=8&ved=0ahUKEwja76Kw5b3VAhUGwI8KHciXD8sQFghIMAY&url=https%3A%2F%2Fwww.pulsesecure.net%2Fdownload%2Ftechpubs%2Fcurrent%2F588%2Fpulse-policy-secure%2Fpps%2F5.3rx%2Fps-pps-ic-5.3-admin-guide.pdf&usg=AFQjCNHbLpLnzpIZgoMFvb0wDxqOK1_zZw

Configuring General Host Checker Remediation

To specify remediation actions for a Host Checker policy:

1. In the admin console, select **Authentication > Endpoint Security > Host Checker**.
2. Create or enable Host Checker policies.
3. Specify the remediation actions for Host Checker to perform if a computer does not meet the requirements of the current policy:
 - **Enable Custom Instructions**—Enter the instructions to display to the user on the Host Checker remediation page. You can use the following HTML tags to format text and to add links to resources such as policy servers or web sites: `<i>`, ``, `
`, ``, and `<a href>`. For example:

You do not have the latest signature files.

`Click here to download the latest signature files.`
 - **Kill Processes**—On each line, enter the name of one or more processes to kill if the computer does not meet the policy requirements. You can include an optional MD5 checksum for the process. (You cannot use wildcards in the process name.) For example:

keylogger.exe

MD5: 6A7DFAF12C3183B56C44E89B12DBEF56
 - **Delete Files**—Enter the names of files to delete if the user’s computer does not meet the policy requirements. (You cannot use wildcards in the file name.) Enter one filename per line. For example:

c:\temp\bad-file.txt

/temp/bad-file.txt
 - **Send reason strings**—Select this option to display a message to users (called a reason string) that is returned by Host Checker or IMV and that explains why the client machine does not meet the Host Checker policy requirements. This option applies to predefined

40. On information and belief, the Accused Instrumentalities are used, marketed, sold, or otherwise provided by or for Defendant's partners, clients, customers and end users across the country and in this District.

41. Defendant was made aware of the '048 patent and its infringement thereof at least as early as the filing date of this Complaint.

42. Upon information and belief, since at least the filing date of this Complaint, Defendant induced and continues to induce others to infringe at least one claim of the '048 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Defendant's partners, clients, customers, and end users, whose use of the Accused Instrumentalities constitutes direct infringement of at least one claim of the '048 patent.

43. Defendant's actions that aid and abet others such as its partners, customers, clients, and end users to infringe include, since at least the filing date of this Complaint, advertising and distributing the Accused Instrumentalities and providing instruction materials, training, and services related to the Accused Instrumentalities. On information and belief, Defendant has, since at least the filing date of this Complaint, engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Defendant has had actual knowledge of the '048 patent and knowledge that its acts were inducing infringement of the '048 patent since at least the filing date of this Complaint.

44. Upon information and belief, Defendant is liable as a contributory infringer of the '048 patent under 35 U.S.C. § 271(c) by offering to sell, selling, and/or importing into the United States the Accused Instrumentalities for use in practicing the '048 patent knowing, at least as early as the filing date of this Complaint, that the Accused Instrumentalities are especially made

or adapted for use in an infringement of the '048 patent. The Accused Instrumentalities include a material component for use in practicing the '048 patent and are not a staple article or commodity of commerce suitable for substantial non-infringing use.

45. Plaintiff has been harmed by Defendant's infringing activities.

JURY DEMAND

Pursuant to Rule 38 of the Federal Rules of Civil Procedure, Plaintiff demands a trial by jury on all issues triable as such.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff demands judgment for itself and against Defendant as follows:

- A. An adjudication that Defendant has infringed the '705 and '048 patents;
- B. An award of damages to be paid by Defendant adequate to compensate Plaintiff for Defendant's past infringement of the '705 and '048 patents, and any continuing or future infringement through the date such judgment is entered, including interest, costs, expenses and an accounting of all infringing acts including, but not limited to, those acts not presented at trial;
- C. A declaration that this case is exceptional under 35 U.S.C. § 285, and an award of Plaintiff's reasonable attorneys' fees; and
- D. An award to Plaintiff of such further relief at law or in equity as the Court deems just and proper.

Dated: October 24, 2017

DEVLIN LAW FIRM LLC

/s/ Timothy Devlin

Timothy Devlin (#4241)
tdevlin@devlinlawfirm.com
1306 N. Broom St., 1st Floor
Wilmington, Delaware 19806

Telephone: (302) 449-9010

Facsimile: (302) 353-4251

TOLER LAW GROUP, PC

Jeffrey G. Toler (*Pro Hac Vice* motion to be filed)

jtoler@tligiplaw.com

Craig S. Jepson (*Pro Hac Vice* motion to be filed)

cjepson@tligiplaw.com

8500 Bluffstone Cove

Suite A201

Austin, TX 78759

Telephone: (512) 327-5515

Attorneys for Plaintiff

NETWORK SECURITY TECHNOLOGIES, LLC