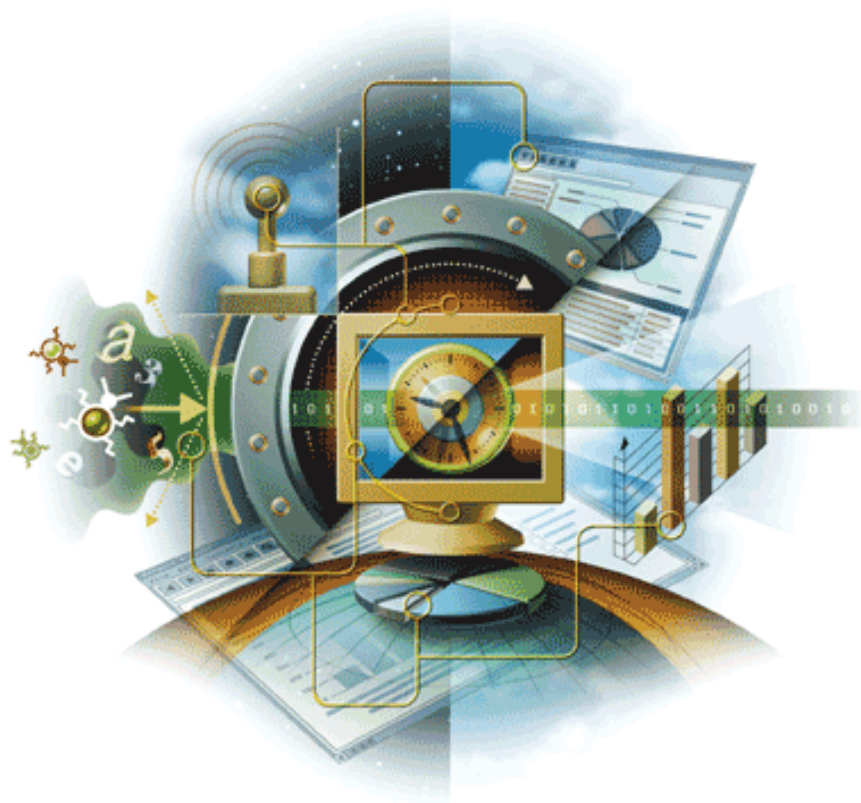


Trusted Connection™ Strategy White Paper Series

Mitigating The Risk of Rogue Systems with ePolicy Orchestrator 3.5

Second in a Series



McAfee®
System Protection

Industry-leading intrusion prevention solutions



COPYRIGHT

Copyright © 2004 Networks Associates Technology, Inc. All Rights Reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means without the written permission of Networks Associates Technology, Inc., or its suppliers or affiliate companies. To obtain this permission, write to the attention of the Network Associates legal department at: 5000 Headquarters Drive, Plano, Texas 75024, or call +1-972-963-8000.

Contents

What is the McAfee Trusted Connection strategy? 3
Introduction. 4
What is McAfee ePolicy Orchestrator 5
Rogue system detection in ePolicy Orchestrator 3.5 5
Architectural overview 6
The rogue sensor 8
The server 8
Deployment of rogue sensors 9
Rogue system response actions 10
Unmanageable systems and devices 12
Frequently asked questions 13
Acronyms used in this document 14

Legal Notes

© Copyright 2004 Networks Associates Technology, Inc. All Rights Reserved.

This document contains confidential, proprietary, or trade secret information of Network Associates, Inc. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, distributed, revised, modified or translated into any language in any form or by any means without the prior written permission of Networks Associates Technology, Inc., or its suppliers or affiliate companies. To obtain this permission, write to the attention of the Network Associates legal department at: 5000 Headquarters Drive, Plano, Texas 75024, or call +1-972-963-8000.

Network Associates, Inc. makes no representations or warranties with respect to the contents of this work, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Network Associates reserves the right to revise this work and to make changes to its content, at any time, without obligation to notify any person or entity of revisions or changes.

Mitigating The Risk of Rogue Systems with ePolicy Orchestrator 3.5

A Technical Overview

This white paper is the second in the McAfee Trusted Connection Strategy Series. This series was created to provide insight and technical detail about specific parts of the overall McAfee Trusted Connection™ Strategy.

The subject of this paper is *Mitigating The Risk of Rogue Systems with ePolicy Orchestrator 3.5*. This document reference number is White Paper 2.

The information in this document provides an overview of how to use ePolicy Orchestrator 3.5 to monitor — in real time — for rogue or unprotected systems that connect to the internal network. It is not designed to demonstrate strategies for automated blocking, which will follow in subsequent white papers or product enhancements.

What is the McAfee Trusted Connection strategy?

The McAfee Trusted Connection strategy is a way of ensuring the security compliance of systems before they connect to the corporate network. This strategy is based on multiple McAfee Security technology initiatives and key partnerships with industry-leading vendors of VPN, remote access, wireless, and networking.

The objective of this strategy is to allow McAfee users to check for security compliance on systems before they connect to the network, and to automatically remediate if necessary, from both external and internal network access points. With these solutions, IT administrators can ensure that only securely-configured systems are connected to their corporate network, giving them greater control, and enhancing proactive protection from vulnerabilities and the transfer of viruses, worms and Trojan horses.

Introduction

One challenge that any organization faces when managing its system security and ensuring complete enterprise protection is that in order to enforce policy compliance, a system must be known. This situation is compounded by the fact that in most networks, the only requirement to join the network is physical access. No further authentication is needed. Therefore, any visitor who enters a corporate building and uses an available network connection unintentionally represents a significant threat to that organization.

Such scenarios include:

- Contractors, outsourced employees, or business partners — whose computers are not managed by your security infrastructure — connecting to the network.
- Unknown or unauthorized assets or systems within the organization connecting to the network, even though they may remain unnoticed.
- Visitors in conference room facilities who plug into the network to synchronize e-mail.

A single computer lacking appropriately managed protection can be a threat to the entire network, which means that knowledge of all systems connected to the network is critical to successfully protect the enterprise. Systems connecting to the network that are not known, or not conforming to the defined security policy, are considered *rogue systems*.

There are a number of existing strategies for creating and maintaining a list of all systems connected to a network, but each of them has significant drawbacks:

| Existing strategy | Drawbacks |
|---|---|
| Manually maintaining a list of all systems on a network | Not possible for most modern networks; because the network environment is too dynamic for one person (or even several people) to keep such a list current. |
| Registering systems with logon scripts | Logon scripts are run <i>after</i> a user authenticates with some entity (for example, a Windows domain). Therefore, the logon script is not a prerequisite for a system joining the network. |
| Periodic scans using an active scanning tool (such as Nmap) or a similar vulnerability scanner | <p>If scans are periodic, only rogue systems connected at that point in time are discovered. Increasing the scan frequency to compensate is not viable because scans can be:</p> <ul style="list-style-type: none"> ■ Very time consuming. ■ Intrusive, creating a great deal of network traffic and sometimes creating traffic which adversely affects services being scanned for. <p>Scans may also be prevented from running by desktop firewall software, if it has been configured to ignore all incoming connections, making a system invisible to scanning software.</p> |
| Periodic importing from a directory (for example, Active Directory) | While greatly reducing administration time spent on directory maintenance, this shares the limitations of the two previous strategies. Like active scanning, it is periodic, and like logon scripts, it does not provide ePolicy Orchestrator visibility into systems on the network that aren't registered with Active Directory. |

To face these challenges and overcome the limitations of existing solutions, McAfee is introducing a novel approach to detecting and managing rogue systems in conjunction with the release of McAfee ePolicy Orchestrator 3.5.

What is McAfee ePolicy Orchestrator?

McAfee ePolicy Orchestrator (ePO) is the industry leading system security management solution-delivering a coordinated, proactive defense against malicious threats and attacks for the enterprise. Providing unmatched, comprehensive management of system security at the lowest cost of ownership, it ensures compliance with system security policy and the effectiveness of system protection, preventing costly business disruptions caused by malware infections and attacks. As the central hub of McAfee system protection solutions, administrators can proactively mitigate the risk of rogue, noncompliant systems, keep protection up-to-date, configure and enforce protection policies, and monitor security status, 24/7, from one centralized, and truly, enterprise-scalable console..

Rogue system detection in ePolicy Orchestrator 3.5

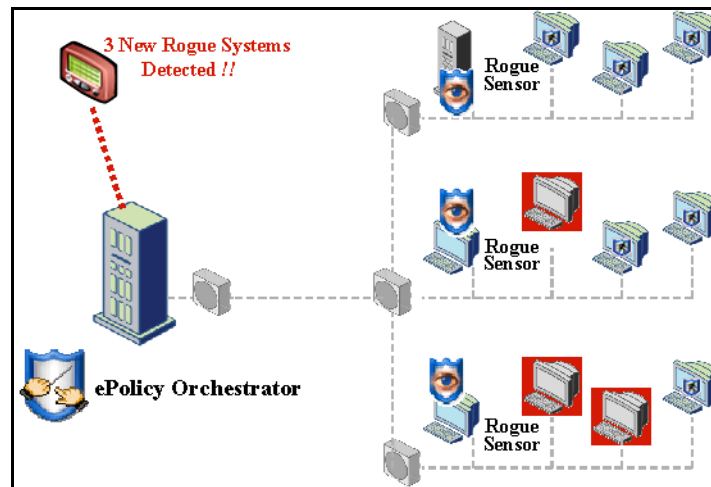
Rogue system detection is a new feature in the forthcoming release of ePolicy Orchestrator 3.5. This feature is designed to improve policy compliance within enterprises by identifying all rogue or unprotected systems and allowing ePolicy Orchestrator to invoke a policy-based response on that system.

At the heart of this solution is a software-based sensor that uses passive monitoring to detect all systems participating in the network. Specifically, the sensor listens for L2 broadcasts (see [About the OSI 7-Layer reference model on page 7](#) for more information). Computers participating on a network tend to broadcast frequently, especially when first joining a network, so new systems are usually detected by the sensor within seconds of first connecting to the network. Sensors deployed throughout the enterprise report all detected systems to the ePolicy Orchestrator server, and the server determines which of those devices are rogue.

Architectural overview

The following diagram gives an overview of rogue system detection architecture.

Figure 1 Architectural overview



At least one rogue system sensor must be deployed in each L2 segment throughout the enterprise, because sensors detect systems by broadcasts (which are only propagated through an L2 segment). As systems are detected, the sensor uses HTTPS protocol to send messages describing the systems to the ePolicy Orchestrator server. The sensor makes no attempt to classify systems as rogue or managed; it simply reports everything it sees.

When the ePolicy Orchestrator server receives a **system detected** message, it inspects the database to determine whether the system should be classified as rogue or managed. A system is considered rogue if:

- It is not present in ePolicy Orchestrator's database of managed systems, and
- The system's ePolicy Orchestrator agent is not actively communicating with the server.



ePolicy Orchestrator enforces policy on systems through a small software agent running on managed systems. Those agents are responsible for periodically checking in with the ePolicy Orchestrator server to obtain the most recent policy settings. Failure of an agent to check in and confirm its policy settings is considered a breach of policy because the ePolicy Orchestrator server cannot confirm that the system's settings are up to date.

The detected system's MAC (Media Access Control) address is used as the primary key when searching through ePolicy Orchestrator's managed system database; the hostname can also be used to reduce false-positives in cases when a system uses multiple network interfaces, such as a laptop with both wireless and Ethernet (see [Frequently asked questions on page 13](#) for details.)

A sensor reports on a given system the first time it is detected (for example, when the first broadcast packet containing that system's MAC address is received by the sensor) and then no more frequently than once per a configurable time period (set to one hour by default). Each time the server receives a **system detected** message for a previously detected system, it recalculates and updates the rogue status and other information associated with the system. This processing model has some highly desirable consequences:

- The rogue system information in the ePolicy Orchestrator database represents the current state of the network, not a snapshot from some point in the past, which is the case with active scanners.
- An administrator can tell whether a rogue system is still active, when it was last active, and how long it has been on the network. The rogue system section of the ePolicy Orchestrator console classifies **inactive** systems (systems which have not communicated in some configurable time period) separate from **active rogue** systems, so that the administrator can focus on the network's current threats.
- The server architecture is greatly simplified because each system's rogue status and other properties are kept up-to-date without any explicit background processing. The server processes events as they come in.

About the OSI 7-Layer reference model

The Open System Interconnection (OSI) reference model is a theoretical model for how networked applications communicate with each other. It describes a 7-layer protocol stack, in which each layer of the stack represents a network protocol that builds upon the capabilities of the layer below it. Applications, such as HTTP, operate in **Layer 7** while the physical media is represented in **Layer 1**. Layers important for understanding this document are as follows:

Layer 2 — Data Link Layer: Ethernet is an example of an L2 protocol. Devices are addressed using their six-byte MAC (Media Access Control) address. Ethernet supports packets sent to a single device or broadcast to all devices in the network. The set of all devices that will receive broadcast packets from each other are considered to be in the same broadcast domain; a broadcast domain is sometimes also referred to as a segment. Hubs, switches and bridges can be used to connect devices in a broadcast domain. Routers do not forward broadcast packets; thus, an L2 network can include many switches and hubs, but will not include any routers.

Layer 3 — Networking Layer: The IP (Internet Protocol) is at this layer. Devices in an IP (v4) network are addressed using a 4-byte IP address. The ARP (Address Resolution Protocol) is used to translate L3 IP addresses into L2 MAC addresses; this is necessary because within a segment, L3 relies on L2 to transmit packets to the remote host. ARP relies on the broadcast functionality provided by L2 to ask all systems on the network: *Whoever has IP address 1.2.3.4, what is your MAC address?* Multiple IP networks are connected using routers.

For additional information, you can find several useful sites on the Internet (for example, http://www.webopedia.com/quick_ref/OSI_Layers.asp).

The rogue sensor

As previously mentioned, the sensor detects systems by listening for L2 broadcasts. Some common network protocols which utilize broadcasts include ARP, which is used to translate L3 IP addresses into L2 MAC addresses, and DHCP (Dynamic Host Configuration Protocol), which is used to dynamically assign IP addresses to hosts. It is very rare for a system to connect and use a network without utilizing one of these protocols. Empirical evidence has also shown that computers that are actively in use tend to broadcast rather frequently.

The sensor is not totally passive in its information gathering. While the IP and MAC are gathered passively, the sensor makes an active query to obtain the hostname and properties. When the sensor receives a broadcast packet, it extracts both the source IP address and the MAC address. Before the system is reported to the server, the sensor gathers some additional information on the host, which is included in the **system detected** message, such as:

- The DNS name.
- The NetBIOS name.
- Various other NetBIOS properties.

The rogue system sensor is a lightweight software-based service that runs on non-dedicated systems and is deployed and managed using ePolicy Orchestrator configuration settings, defined within the ePolicy Orchestrator console and enforced on each system by the ePolicy Orchestrator agents. Sensor installation can occur in one of the following ways: through ePolicy Orchestrator, with a standalone installer, or with a copied disk image. No matter how it is installed, the sensor requires the ePolicy Orchestrator agent to be present for proper operations.

The server

Information on all detected systems and their rogue status is stored in the ePolicy Orchestrator database. The server also keeps track of which subnets the systems were found in and which sensors are actively reporting in each subnet. This information is displayed in an HTML window embedded in the ePolicy Orchestrator console. The **Rogue Machines** page provides a high-level summary view of the rogue and managed systems detected on the network, and allows drill-down to a filtered list view and details on individual systems. The list view can be organized by subnet or filtered and sorted by almost any type of information known about the systems.

Figure 2 Rogue Machines: Machine List

The screenshot shows the 'Rogue Machines' interface in the McAfee ePolicy Orchestrator console. The 'Machine List' section is active, displaying a table of 11 rogue machines for the subnet 192.168.1.0/24. All machines are listed with a status of 'Rogue' and 'No Agent'. The table columns are Status, Rogue Type, Friendly Name, IP, and Last Detect Time. Below the table, there are options to 'Check All', 'Uncheck All', and 'Add to ePO tree'.

| Status | Rogue Type | Friendly Name | IP | Last Detect Time | |
|--------------------------|------------|---------------|--------|------------------|--------------------|
| <input type="checkbox"/> | Rogue | No Agent | | | |
| <input type="checkbox"/> | Rogue | No Agent | Apple | 192.168.1.100 | 5/24/04 3:27:44 PM |
| <input type="checkbox"/> | Rogue | No Agent | Pear | 192.168.1.101 | 5/24/04 3:27:44 PM |
| <input type="checkbox"/> | Rogue | No Agent | Walnut | 192.168.1.102 | 5/24/04 3:27:44 PM |
| <input type="checkbox"/> | Rogue | No Agent | Salmon | 192.168.1.103 | 5/24/04 3:27:44 PM |
| <input type="checkbox"/> | Rogue | No Agent | Thyme | 192.168.1.104 | 5/24/04 3:27:44 PM |
| <input type="checkbox"/> | Rogue | No Agent | Cookie | 192.168.1.105 | 5/24/04 3:27:44 PM |
| <input type="checkbox"/> | Rogue | No Agent | Pork | 192.168.1.106 | 5/24/04 3:27:44 PM |
| <input type="checkbox"/> | Rogue | No Agent | Ginger | 192.168.1.107 | 5/24/04 3:27:44 PM |
| <input type="checkbox"/> | Rogue | No Agent | Egg | 192.168.1.108 | 5/24/04 3:27:44 PM |
| <input type="checkbox"/> | Rogue | No Agent | Grape | 192.168.1.109 | 5/24/04 3:27:44 PM |
| <input type="checkbox"/> | Rogue | No Agent | Salsa | 192.168.1.110 | 5/24/04 3:27:44 PM |

Deployment of rogue sensors

Given the requirements of one sensor per subnet, how can an administrator easily deploy sensors to all managed subnets, and then be sure that at any point in the future all segments continue to have at least one sensor deployed on them? The ePolicy Orchestrator console provides a number of ways to directly address this question.

- Subnets with at least one sensor deployed and actively reporting are considered covered.
- Subnets without any sensor activity in a configurable time period are considered uncovered.

The rogue system section of the ePolicy Orchestrator console displays a list of all subnets known to ePolicy Orchestrator, along with their coverage status. Administrators can configure ePolicy Orchestrator so that notification is sent when any previously covered subnets become uncovered, which could happen if a system hosting a sensor is turned off or changes subnets. This situation can also be mitigated by deploying more than one sensor per subnet.

Because a subnet must sit on a single broadcast domain (for example, a subnet cannot span two or more broadcast domains), if you ensure that all subnets are covered, you also ensure that all broadcast domains are covered. Throughout the rogue system section of the ePolicy Orchestrator console, sensor coverage is described in terms of subnets, not segments or broadcast domains.

Figure 3 Rogue Machines: Subnet List

The screenshot displays the 'Subnet List' window in the McAfee ePolicy Orchestrator interface. The window title is 'Rogue Machines' and it includes a navigation bar with 'Machines', 'Subnets', 'Events', 'Responses', and 'Configuration'. The 'Subnets' tab is active. Below the navigation bar, there are options for 'Filter: (Custom Filter)', 'Refresh (Auto)', 'Configure Table', and 'Custom Filter'. A 'Back' button is located in the top right corner. The main content is a table with the following data:

| <input type="checkbox"/> | Status | Address/Mask | Network Name | Sensors | Last Sensor Comm. |
|--------------------------|-----------|---------------|--------------|---------|--------------------|
| <input type="checkbox"/> | Uncovered | 10.13.50.0/24 | QA | 0 | |
| <input type="checkbox"/> | Covered | 10.13.51.0/24 | Dev | 1 | 5/24/04 3:41:19 PM |
| <input type="checkbox"/> | Uncovered | 10.13.52.0/24 | HR | 1 | 5/19/04 3:48:14 PM |
| <input type="checkbox"/> | Uncovered | 10.13.53.0/24 | Support | 0 | |
| <input type="checkbox"/> | Uncovered | 10.13.54.0/24 | Sales | 0 | |
| <input type="checkbox"/> | Covered | 10.13.55.0/24 | Marketing | 2 | 5/24/04 3:47:35 PM |

Below the table, there are 'Check All' and 'Uncheck All' buttons. At the bottom right, it says '176 items in 12 pages. Go to page: 1'. Below the table, there is a 'Checked subnets:' section with a 'Deploy Sensors...' button.

From the **Subnet List** window, an administrator can select subnets that are not covered, and deploy sensors to them using ePolicy Orchestrator's software deployment capabilities. Within the selected subnets, systems used to host the sensor software may be picked manually from a list or chosen automatically by the ePolicy Orchestrator server, using criteria provided by the administrator. Possible criteria for automatic system selection include operating system version, processor speed, system memory, or the time of last ePolicy Orchestrator agent communication.

Rogue system response actions

There are a number of different actions that an administrator can take in response to rogue system detections. Each of these actions can either be performed manually, by selecting the system from the user interface's list of rogue systems and then selecting the action, or by automatically implementing a pre-defined response.

Automatic actions have a set of associated conditions. As new rogue systems are detected, the actions are invoked only if the conditions are met. An action's condition can depend on any of the information known about a rogue system and can be a simple comparison of a single field or a complex compound statement. An example condition could correspond to the English sentence: *If a new rogue system's IP is in the range 192.168.1.0/24 and its operating system is Windows, push an ePolicy Orchestrator agent to that system.* An automatic action and its associated conditions are collectively referred to in the rogue system section of the ePolicy Orchestrator console as an automatic response.

Figure 4 Rogue Machines: Add or Edit Automatic Response

Rogue Machines

Machines Subnets Events Responses Configuration

Automatic Responses Help

Add or Edit Automatic Response [Back]

Name:

Event: **Rogue Machine Detected**

Enabled:

Conditions:

Match All (AND) Match Any (OR)

| Property | Comparison | Value | Delete |
|-------------|-------------|--------------------------------------|--------|
| IP | is in range | 192 .168 .1 .0 - 192 .168 .1 .255 | ✘ |
| OS Platform | contains | Windows | ✘ |

Actions:

| Method | Parameters | Delete |
|-------------------|------------|--------|
| Mark as Exception | (none) | ✘ |

Method dropdown menu:

- Mark as Exception
- Add to ePO tree
- Mark For Action
- Mark as Exception
- Push ePO Agent
- Query ePO agent
- Remove Host
- Send E-mail
- Send ePO Server Event
- Unmark For Action
- Unmark as Exception

Some of the actions that can be performed on detected rogue systems include:

- **Push ePO Agent:** This is the most direct form of rogue remediation. Once the agent has been installed, it enforces ePolicy Orchestrator's policies, and the target system is considered managed. When this happens, the server removes it from the list of rogue systems.
- **Send E-mail:** Using ePolicy Orchestrator's new alerting functionality, an administrator can receive e-mail or SNMP notification when a new rogue system is found.
- **Run an external tool:** Additional options appear in the pull-down menu if you have any external (third-party) tools installed. An external tool can be any executable installed on the ePolicy Orchestrator server. This action allows for custom remediation or notification (a feature not natively supported by ePolicy Orchestrator). It also allows the integration of third-party probing tools, so an administrator can gather additional information on a given rogue system before taking further action.
- **Mark as Exception:** Marking a system as an exception changes its state in the rogue system database to indicate that it is not manageable by ePolicy Orchestrator, and thus should not be considered a rogue. See [Unmanageable systems and devices on page 12](#) for more about false positives, and details on exceptions.
- **Mark For Action:** Marking a system as needing future action is a way for administrators to flag rogue systems that they aren't ready to remediate, but that they want to revisit later for further action. The system list can be sorted or filtered based on this whether systems are marked for action with a flag that appears next to the system name in the ePolicy Orchestrator console.

Unmanageable systems and devices

Classification of unmanageable systems as rogue will occur on every network, however, some of these classifications may be false positives. This happens because many devices participating in a network do not need to be manageable by ePolicy Orchestrator. Some examples of false positives are: routers, printers, various types of network appliances, and systems running operating systems not supported by ePolicy Orchestrator. To eliminate classification of these systems as rogue, the rogue system database includes the notion of exception systems, which can be marked in the ePolicy Orchestrator console.

When an administrator marks unmanageable rogues as exceptions, they no longer appear in ePolicy Orchestrator's list of rogue systems. Information gathered by the sensor (DNS name, NetBIOS name, and other NetBIOS information such as operating system, comments, and domain) and the IEEE OUI (Organizationally Unique Identifier) name. The OUI makes up the first three bytes of a MAC address, and is registered to a company or organization (usually the network card manufacturer), and it can aid the administrator in identifying exceptions. Third-party probing tools and other external applications can also be used to gather additional information. Their output is then captured and displayed along with the other rogue system information.

In addition to manually selecting systems as exceptions, the administrator can also create rules to mark systems as exceptions when they meet certain criteria, using the automatic response mechanism previously described. For example, if an enterprise uses Cisco routers and does not use any Cisco NICs, an automatic response could be: *If a new rogue system's OUI name contains 'Cisco' then mark it as an exception.*

Immediately after the initial deployment of rogue system sensors, it is anticipated there will be some false positives that do not need to be managed by ePolicy Orchestrator. During this tuning period, the administrator should manually mark these systems as exceptions or write automatic responses to help automate the process. After the majority of the initial exceptions are found, most new rogue system detections are legitimate. At this point, the administrator can enable more aggressive automatic remediation rules or notifications.

Frequently asked questions

Will the sensors operate correctly in a switched network?

Yes, because switches propagate L2 broadcast traffic. Switches only limit which devices in a broadcast domain can see unicast packets.

Will the sensors operate correctly on a virtual LAN (VLAN)?

Yes, from a rogue system detection perspective there is no difference between a VLAN and a LAN. Switches that support VLANs must forward broadcast packets to every other device on the VLAN, and one sensor must be deployed per VLAN.

Will sensors operate correctly with trunked switches (e.g. 802.1q)?

Yes. Trunked switches, which allow a VLAN to span multiple switches by communicating VLAN traffic over a single point-to-point link, must ensure that broadcast traffic is propagated to every device in the VLAN. From a rogue system detection perspective, a VLAN that spans trunked switches is no different than a VLAN on one switch or a regular LAN.

Will sensors operate correctly with network interface cards (NICs) that don't support promiscuous mode?

Yes. Using a network card's driver in promiscuous mode is only required for receiving unicast packets that are not destined for the local system. Because the sensor only listens for broadcast packets, it does not need to open the driver in promiscuous mode.

Will the sensor detect systems connecting through a VPN?

No. VPN servers act as routers (L3) so they do not propagate L2 broadcasts. However, as part of the McAfee Trusted Connection Strategy, McAfee has undertaken several partnerships with leading VPN providers including Check Point, Nortel, Neoteris, Cisco, and Aventail to enable detection and blocking of systems that do not meet a predefined level of security compliance. Initially, these partnerships are focused on anti-virus compliance but over time we are looking to increase the depth of those partnerships.

Will the sensor detect systems connecting through a Wireless Access Point (WAP)?

Yes, if there is a sensor on the wireless subnet, it will detect all systems connecting through the WAP. If a WAP is acting as a router or a NAT device and the sensor is on the outside, it will detect the presence of the WAP itself but will not have visibility inside the wireless network.

Does the sensor need to run on a dedicated system?

No. Although it is desirable to run sensors on systems that are not shut down regularly for extended periods of time, the sensor is not restricted to only running on servers or running on dedicated systems. The memory and processor overhead for a sensor is very minimal; typical utilization is less than 1% on a 500 MHz machine and the typical memory footprint is 5-10 megabytes. The sensor requires no special hardware.

Can rogue system detection deal with systems that regularly connect with multiple NICs, such as a laptop with a wireless card and an Ethernet jack and a docking station?

Yes. Although MAC address is the primary key used to look up detected systems in the ePolicy Orchestrator database, rogue system detection can also be configured to look up based on hostname alone, *or* on both hostname and domain. In this case, no matter whether the laptop is detected on the wireless network or the LAN, if it is managed by ePolicy Orchestrator it will be found in the database and correctly identified as managed.

How does rogue system detection handle systems that are managed by another ePolicy Orchestrator server?

These systems will be considered rogue, but rogue system detection can be configured to identify such systems as possessing *foreign agents*. The rogue system detection server will query rogues for the presence of an ePolicy Orchestrator agent, and if one is found, the system will be identified as having a foreign agent. The name of the foreign ePolicy Orchestrator server will be recorded and displayed in the rogue system detection section of the ePolicy Orchestrator console.

Acronyms used in this document

| | |
|--------------|--|
| ARP | Address Resolution Protocol |
| HTTPS | Hypertext Transfer Protocol (Secure) |
| Lx | Level x (where x is a number) of the OSI reference model |
| LAN | Local Area Network |
| MAC | Media Access Control |
| NAT | Network Address Transaction |
| NIC | Network Interface card |
| Nmap | Network Mapping Tool |
| OSI | Open System Interconnection |
| OUI | Organizationally Unique Identifier |
| VLAN | Virtual Local Area Network |
| VPN | Virtual Private Network |
| WAP | Wireless Access Point |