Matching Devices

Match Process

When Open-AudIT receives data about a device, either by discovering the device during an audit run or by the user importing the device, it must determine if this discovered device matches a device that already exists within its database, or if it is a new device that should be added. Open-AudIT uses a series of twelve property matches to determine this. The Match Rules work as OR comparisons, not AND. This means the first rule that matches a field in the discovered device to one in the dB resolves as an existing device. All Matching Rules have to fail in order for a device to be new and result in a new record being created.

Duplicate Devices / Missing Devices

It is important to note that when Open-AudIT determines a match any properties set to 'y' must match exactly (and not be blank) in order for Open-AudIT to determine that the discovered device matches a device already in the database. If none of the properties marked 'Y' match, then a new device entry will be created, which could result in duplicate device entries. In situations where properties are duplicated, for example a dbus_id is copied during a VM clone, then an existing device may incorrectly get overwritten/updated rather then a new entry being created resulting in missing devices.

Devices **will not be matched** if their status is set to "deleted". Any other status will allow a match to occur.

Matching Linux Devices

When matching a Linux based device, we prefer to use the Dbus id concatenated with the hostname. We can also use other options as per the below table, but we can retrieve the Dbus ID without root. To retrieve the UUID (from the motherboard), we need to run dmidecode, which does require root. Unfortunately, when you clone an ESXi guest, the Dbus ID does not get recreated - hence our concatenating this with the hostname. There is a good article linked here that details the why's of hardware IDs. http://0pointer.de/blog/projects/ids.html

Match Order

The logic for device matching is contained in the m_devices.php file, which on a Linux install can be found here: /usr/local/open-audit/code_igniter /application/models/

Matching is conducted in the following order:

1. Match the Opmantek UUID (not configurable).
2. Match the Google Cloud ID (not configurable).
3. match_hostname_uuid
4. match_hostname_dbus
5. match_hostname_serial
6. match_dbus
7. match_dns_fqdn
8. match_dns_hostname
9. match_fqdn
10. match_serial_type
11. match_serial
12. match_sysname_serial
13. match_sysname
14. match_mac (ip table)
15. match_mac (network table)
16. match_mac (addresses)
17. match_ip
18. match_hostname
19. match_ip_no_data

Matching IP Addresses

As at Open-AudIT 3.3.0 we will be implementing a match routine that essentially says "If all I have is an IP, and that IP belongs to a device in the database and that device has not been audited, match that device regardless of the match_ip rule.

The reason for this is in the case of a discovered device that we don't have credentials for, we have virtually no information except the IP and maybe the DNS Hostname. Neither are considered unique (think DHCP). But in the case where we have a device with that lack of data already preset in the database, assume it is the same device so that we don't create many false duplicates. This configuration item will be called match_ip_no_data and will be set to YES by default.

Enterprise Per Discovery Matching

For Enterprise Users, the below properties are able to be set ***per individual discovery***. If a discovery has a value of 'y' or 'n' for a match rule, that will be used. If no value is present, the discovery will use the value set in the global configuration.

Using OrgID

New for 4.3.4 - please see our match options using a device's OrgID - Matching Devices - Including OrgID.

Match Properties

These properties are stored in Open-AudIT's configuration; to access them select Admin -> Configuration -> List Configuration from Open-AudIT's menu, then use the Search box on the left to search for ***match_*** or go to Menu Discover Discoveries Discovery Match Options. The default values of 'y' and 'n' simply mean YES and NO. We will use YES and NO in the description, rather than 'y' and 'n'. The stored value should always be either a lowercase y or n.

The properties and their default values are listed below.

Property	Default Value	Description
match_dbus	n	Linux based devices only. The DBUS id is supposed to be unique on each Linux device. It is set to NO by default because ESX, upon cloning a guest virtual machine, does not tell the operating system to recreate this identifier. We were receiving reports of discovered devices overwriting one another and this was the culprit.
match_fqdn	y	Should we match a device based on its fqdn.
match_dns_fqdn	n	Should we match a device based on its DNS fqdn.
match_dns_hostname	n	Should we match a device based on its DNS hostname.
match_hostname	y	Should we match a device based on its hostname? Set to YES as hostnames should be unique to a network. This may be a candidate for changing as some users may wish to audit disparate networks (say several different customers networks) that contain hostnames that are identical to others already in Open-AudIT. Say 'web' or 'mail' or 'dns', etc. Certain hostnames are not uncommon to use.
match_hostname_dbus	y	Linux based devices only. Should we use the combination of the hostname (as determined by Open-AudIT) and DBUS id (as reported by an audit script or SSH command) to determine a device match? Set to YES as this is considered a reliable combination.
match_hostname_serial	y	Should we use the combination of the hostname (as determined by Open-AudIT) and serial (as reported by an audit script, SSH command or SNMP query) to determine uniqueness. Set to YES as this is considered a reliable combination.
match_hostname_uuid	y	Should we use the combination of the type (as determined by Open-AudIT) and serial (as reported by an audit script, SSH command or WMI command) to determine uniqueness. Set to YES as this is considered a reliable combination.
match_ip	n	Should we match based only on the device's IP address? Set to NO because DHCP will cause false positive matches. This may be acceptable to set to YES if you can guarantee no devices will change IP addresses. You may only ever audit a server network for example. In most cases, it is best to leave this to NO.
match_ip_no_data	y	Should we match a device based on its ip if we have an existing device with no data.
match_mac	y	Should we match a device based only on it's discovered MAC addresses. Set to NO prior to 3.3.0. Post 3.3.0 will be set to YES. A MAC address should be unique on a network. See below for an exception to the rule.
match_mac_vmware	n	VMware Workstation tends to use MAC addresses that are not globally unique. IE - Two different workstations may be running VMware Workstation and have two different virtual machines that have the same MAC address. These machines won't ever need to perform networking outside their hosts using this MAC address, but Open-AudIT will discover the MAC addresses upon an audit. Should we determine uniqueness based on these mac addresses? These MAC addresses typically start with one of the following: 00:0c:29, 00:50:56, 00:05:69, 00:1c:14.
match_serial	y	Should we use the serial (as reported by an audit script, SSH command, WMI command or SNMP query) to determine a device match? Set to YES as this is considered a reliable attribute.
match_serial_type	y	Should we use the combination of the type (as determined by Open-AudIT) and serial (as reported by an audit script, SSH command, WMI command or SNMP query) to determine uniqueness. Set to YES as this is considered a reliable combination.
match_sysname	y	Should we match a device based only on its SNMP sysName.

match_sysname_serial	y	Should we match a device based only on its SNMP sysName and serial.
match_uuid	y	Should we use the UUID (as reported by an audit script, SSH command or WMI command) to determine a device match? Set to YES as this is considered a reliable attribute.