Errata - 3.3.2 Discoveries

When a discovery has been created in an older version of Open-AudIT it can have certain required items missing. From the next release these will be better catered for, but in order to fix this now, add the code below and run it to automatically fix any incorrectly setup discoveries.

One consequence of this was certain discoveries would not detect the "must respond to ping" setting, and default to "no, it doesn't have to respond", and as a result give you a device entry for every IP in the discovery.

To remove these, view the device list and bulk edit the devices of type "unknown".

In the file:

Linux - /usr/local/open-audit/code_igniter/application/controllers/test.php

Windows c:\xampp\open-audit\code_igniter\application\controllers\test.php

Add the below code after line 80 (after the index function, before the base function), and continue reading below.

```
    public function discoveries_fix()
    {
        $this->load->model('m_configuration');
        $this->m_configuration->load();
        $sql = 'SELECT * FROM discovery_scan_options WHERE id = ' . intval($this->config->config
['discovery_default_scan_option']);
        $query = $this->db->query($sql);
        $result = $query->result();
        if ( ! empty($result)) {
            $scan_options = $result[0];
        } else {
            $sql = 'SELECT * FROM discovery_scan_options WHERE name LIKE "%fast" ORDER BY `name` DESC LIMIT 1';
            $query = $this->db->query($sql);
            $result = $query->result();
            if ( ! empty($result)) {
                $scan_options = $result[0];
            } else {
                echo '<h1>ERROR</h1><br /><p>There are no discovery scan options in the database we can use,
please run the below in the MySQL client or contact <a href="https://opmantek.com">Opmantek</a> if you are a
supported customer.';
                echo "<br /><br /><pre>\n";
                echo "INSERT INTO `discovery_scan_options` VALUES (1,'UltraFast',1,'Approximately 1 second per
target. Scan only the ports that Open-AudIT needs to use to talk to the device and detect an IOS device (WMI,
SSH, SNMP, Apple Sync). An open|filtered port is considered closed. Device must respond to an Nmap ping. Use
aggressive timing.','y','n','n','n',0,4,0,0,'22,135,62078','161','','','','22','','system','2000-01-01 00:00:
00');\n\n
INSERT INTO `discovery_scan_options` VALUES (2,'SuperFast',1,'Approximately 5 seconds per target. Scan the top
10 TCP and UDP ports, as well as port 62078 (Apple IOS detection). An open|filtered port is considered closed.
Device must respond to an Nmap ping. Use aggressive timing.','y','n','n','n',
0,4,10,10,'62078','','','','','22','','system','2000-01-01 00:00:00');\n\n
INSERT INTO `discovery_scan_options` VALUES (3,'Fast',1,'Approximately 40 seconds per target. Scan the top 100
TCP and UDP ports, as well as port 62078 (Apple IOS detection). An open|filtered port is considered closed.
Device must respond to an Nmap ping. Use aggressive timing.','y','n','n','n',
0,4,100,100,'62078','','','','','22','','system','2000-01-01 00:00:00');\n\n
INSERT INTO `discovery_scan_options` VALUES (4,'Medium (Classic)',1,'Approximately 90 seconds per target. As
close to a traditional Open-AudIT scan as we can make it. Scan the top 1000 TCP ports, as well as 62078 (Apple
IOS detection) and UDP 161 (SNMP). An open|filtered port is considered open (and will trigger device
detection). Devices are scanned regardless of a response to an Nmap ping. Use aggressive
timing.','n','n','y','y',0,4,1000,0,'62078','161','','','','22','','system','2000-01-01 00:00:00');\n\n
INSERT INTO `discovery_scan_options` VALUES (5,'Medium',1,'Approximately 100 seconds per target. Scan the top
1000 TCP and top 100 UDP ports, as well as port 62078 (Apple IOS detection). An open|filtered port is not
considered open. Device must respond to an Nmap ping. Use aggressive timing.','y','n','n','n',
0,4,1000,100,'62078','','','','','22','','system','2000-01-01 00:00:00');\n\n
INSERT INTO `discovery_scan_options` VALUES (6,'Slow',1,'Approximately 4 minutes per target. Scan the top 1000
TCP and top 100 UDP ports, as well as port 62078 (Apple IOS detection). Version detection enabled. An
open|filtered port is considered open (and will trigger device detection). Device must respond to an Nmap ping.
Use normal timing.','y','y','y','y',0,3,1000,100,'62078','','','','','22','','system','2000-01-01 00:00:00');
\n\n
INSERT INTO `discovery_scan_options` VALUES (7,'UltraSlow',1,'Approximately 20 minutes. Not recommended. Scan
the top 1000 TCP and UDP ports, as well as port 62078 (Apple IOS detection). Devices are scanned regardless of
a response to an Nmap ping. Version detection enabled. An open|filtered port is considered open (and will
trigger device detection). Use polite timing.','n','y','y','y',
0,2,1000,1000,'62078','','','','','22','','system','2000-01-01 00:00:00');\n\n</pre>";
```

```
                exit;
            }
        }
        echo "<h3>Please save this page for reference</h3>\n";
        echo '<hr /><pre>';
        $other = new stdCLass();
        $other->ad_domain = '';
        $other->ad_server = '';
        $other->subnet = '';
        $other->nmap = new stdClass();
        $other->nmap->discovery_scan_option_id = intval($scan_options->id);
        $other->nmap->filtered = $scan_options->filtered;
        $other->nmap->{'open|filtered'} = $scan_options->{'open|filtered'};
        $other->nmap->ping = $scan_options->ping;
        $other->nmap->service_version = $scan_options->service_version;
        $other->nmap->ssh_ports = $scan_options->ssh_ports;
        $other->nmap->tcp_ports = $scan_options->tcp_ports;
        $other->nmap->timing = $scan_options->timing;
        $other->nmap->udp_ports = $scan_options->udp_ports;
        $other->nmap->timeout = $scan_options->timeout;
        $other->nmap->nmap_tcp_ports = $scan_options->nmap_tcp_ports;
        $other->nmap->nmap_udp_ports = $scan_options->nmap_udp_ports;
        $other->match = new stdClass();
        $warning = '<b>WARNING</b>';
        $error = '<b>ERROR</b>';
        $sql = 'SELECT * FROM discoveries';
        $query = $this->db->query($sql);
        $result = $query->result();
        foreach ($result as $discovery) {
            echo 'Checking discovery named: ' . htmlentities($discovery->name);
            $original_discovery = clone $discovery;
            $original_discovery->other = @json_decode($original_discovery->other);
            $output = '';
            if (empty($discovery->other)) {
                $output .= $error . " - There is no discoveries.other attribute. This discovery will never run.
Please delete it.\n";
                $discovery->other = $other;
            } else {
                $discovery->other = json_decode($discovery->other);
            }
            if (empty($discovery->other->subnet) && $discovery->type === 'subnet') {
                $output .= $error . " - There is no discoveries.other.subnet attribute, although the
discoveries.type is subnet. This discovery will never run. Please delete it.\n";
            }
            if ((empty($discovery->other->ad_server) OR empty($discovery->other->ad_domain)) && $discovery-
>type === 'active directory') {
                $output .= $error . " - There is no discoveries.other.ad_server or discoveries.other.ad_domain
attribute, although the discoveries.type is active directory. This discovery will never run. Please delete it.
\n";
            }
            if (empty($discovery->other->nmap)) {
                $discovery->other->nmap = $other->nmap;
                $output .= $warning . " - No discoveries.other.nmap, populating with default.\n";
            }
            if (empty($discovery->other->match)) {
                $discovery->other->match = $other->match;
                $output .= $warning . " - No discoveries.other.match, populating with default.\n";
            }
            if ($output !== '') {
                echo "\n{$output}\n";
                $sql = "UPDATE discoveries SET other = '" . json_encode($discovery->other) . "' WHERE id = " .
intval($discovery->id);
                echo $sql . "\n";
                $query = $this->db->query($sql);
                echo "</pre>\n";
                echo "<table><tr><td style=\"vertical-align:text-top\"><b>Original</b>\n<pre>\n";
                print_r($original_discovery);
                echo "</pre></td><td style=\"vertical-align:text-top\"><b>Modified</b>\n<pre>\n";
                print_r($discovery);
                echo "</pre></td></tr></table>\n<pre>\n\n<hr />\n";
            } else {
```

```
                echo " - PASSED.\n\n<hr />\n";
            }
        }
    }
```

Once that code has been added, call the below URL in your browser (after logging in to Open-AudIT). Note that you must have the 'admin' role to access the below page.

http://YOUR_SERVER/open-audit/index.php/test/discoveries_fix

The page will list each discovery and for those that are OK, say PASSED.

For those not OK, it will provide a reason why and update the entry in the database.

You should see a page with output similar to below.

The discovery with bad JSON is plainly obvious. We also output the SQL statement we ran to fix it, along with before and after output.

As the page title says, you should save the page for future reference in case anything untended happens.

You may need to delete the discovery logs and re-run discoveries for them to be viewable in the GUI. To delete the logs go to menu Admin Database List Tables Discovery Log DELETE. No device data will be lost, just the discovery log data, which will be repopulated the next time the discovery is run.

Please save this page for reference

```
Checking discovery named: My Discovery - PASSED.
```

```
Checking discovery named: Subnet 192.168.0.0/24 - PASSED.
```

```
Checking discovery named: Marketing Subnet - PASSED.
```

```
Checking discovery named: Another Discovery - PASSED.
```

```
Checking discovery named: Test Bad JSON Discovery
WARNING - No discoveries.other.nmap, populating with default.
WARNING - No discoveries.other.match, populating with default.

UPDATE discoveries SET other = '{"ad_domain":"","ad_server":"","subnet":"192.168.1.0\/24","nmap":
{"discovery_scan_option_id":1,"filtered":"n","open|filtered":"n","ping":"y","service_version":"n","ssh_ports":"
22","tcp_ports":"22,135,62078","timing":"4","udp_ports":"161","timeout":"0","nmap_tcp_ports":"0","
nmap_udp_ports":"0"},"match":{}}' WHERE id = 23
```

Original

```
stdClass Object
(
    [id] => 23
    [name] => Test Bad JSON Discovery
    [org_id] => 1
    [description] => Subnet - 192.168.1.0/24
    [type] => subnet
    [devices_assigned_to_org] => 1
    [devices_assigned_to_location] => 1
    [network_address] =>
    [system_id] => 0
    [other] => stdClass Object
        (
            [ad_domain] =>
            [ad_server] =>
            [subnet] => 192.168.1.0/24
        )

    [options] =>
    [discard] => n
    [last_run] => 2001-01-01 00:00:00
    [last_finished] => 2001-01-01 00:00:00
    [duration] => 00:00:00
    [status] =>
    [ip_all_count] => 0
    [ip_responding_count] => 0
    [ip_scanned_count] => 0
    [ip_discovered_count] => 0
    [ip_audited_count] => 0
    [edited_by] => Admin
    [edited_date] => 2020-05-21 15:26:22
)
```

Modified

```
stdClass Object
(
    [id] => 23
    [name] => Test Bad JSON Discovery
    [org_id] => 1
    [description] => Subnet - 192.168.1.0/24
    [type] => subnet
    [devices_assigned_to_org] => 1
    [devices_assigned_to_location] => 1
    [network_address] =>
    [system_id] => 0
    [other] => stdClass Object
        (
            [ad_domain] =>
            [ad_server] =>
            [subnet] => 192.168.1.0/24
            [nmap] => stdClass Object
                (
                    [discovery_scan_option_id] => 1
                    [filtered] => n
                    [open|filtered] => n
                    [ping] => y
                    [service_version] => n
                    [ssh_ports] => 22
                    [tcp_ports] => 22,135,62078
                    [timing] => 4
                    [udp_ports] => 161
                    [timeout] => 0
                    [nmap_tcp_ports] => 0
                    [nmap_udp_ports] => 0
                )

            [match] => stdClass Object
                (
                )


        )

    [options] =>
    [discard] => n
    [last_run] => 2001-01-01 00:00:00
    [last_finished] => 2001-01-01 00:00:00
    [duration] => 00:00:00
    [status] =>
    [ip_all_count] => 0
    [ip_responding_count] => 0
    [ip_scanned_count] => 0
    [ip_discovered_count] => 0
    [ip_audited_count] => 0
    [edited_by] => Admin
    [edited_date] => 2020-05-21 15:26:22
)
```