

# Secure Boot Investigation

Thursday, October 26, 2017 4:33 AM

## 1) Racadm SecureBoot

install racadm link: <http://linux.dell.com/repo/community/openmanage/>

GET /Systems/{identifier}/SecureBoot

```
{
  "@odata.context": "/redfish/v1/$metadata#SecureBoot.SecureBoot",
  "@odata.id": "/redfish/v1/Systems/59f80fa1b25c818957eda267/SecureBoot",
  "@odata.type": "#SecureBoot.v1_0_1.SecureBoot",
  "SecureBootCurrentBoot": "Disabled",
  "Id": "59f80fa1b25c818957eda267",
  "Name": "Secure Boot Current Status"
}
```

POST /Systems/{identifier}/SecureBoot

```
{
  "SecureBootEnable": true
}
```

```
{
  "Message": "Successfully Completed Request"
}
```

In fact, it failed to set SecureBoot, since it doesn't create the commit and reboot job.

```
vagrant@rackhd:~/onrack/scripts$ sudo racadm -r 192.168.188.23 -u admin -p admin set bios.SysSecurity.SecureBoot Enabled
Security Alert: Certificate is invalid - self signed certificate
Continuing execution. Use -S option for racadm to stop execution on certificate-related errors.
[Key=BIOS.Setup.1-1#SysSecurity]
RAC1017: Successfully modified the object value and the change is in
pending state.
To apply modified value, create a configuration job and reboot
the system. To create the commit and reboot jobs, use "jobqueue"
command. For more information about the "jobqueue" command, see RACADM
help.
```

## 2) Redfish

Racadm-getSecureBoot

on-http/lib/api/redfish-1.0/systems.js

```
var getSecureBoot = controller(function(req, res) {
  var identifier = req.swagger.params.identifier.value;
  var options = redfish.makeOptions(req, res, identifier);

  return wsman.isDellSystem(identifier)
    .then(function(result){
      if(result.isDell){
        return getObmSettings(identifier)
          .then(function(obmSettings) {
            return racadm.runCommand (
              obmSettings.host,
              obmSettings.user,
              obmSettings.password,
              'get bios.SysSecurity.SecureBoot'
            );
          });
      }
      .then(function(secureBoot) {
        var secureBootSetting = _.words(secureBoot, /SecureBoot=[a-zA-Z]*/);
        var statusArray = secureBootSetting[0].split('=');
        options.SecureBoot = (statusArray[1]);
        return redfish.render('redfish.1.0.0.secureboot.1.0.1.json',
          'SecureBoot.v1_0_1.json#/definitions/SecureBoot',
          options);
      });
    } else {
      return redfish.handleError("Not implemented for non-Dell hardware.", res, null, 501);
    }
  });
```

Racadm-setSecureBoot

on-http/lib/api/redfish-1.0/systems.js

```

var setSecureBoot = controller(function(req, res) {
  var identifier = req.swagger.params.identifier.value;
  var command;
  if (req.body.SecureBootEnable) {
    command = "Enabled";
  } else {
    command = "Disabled";
  }
  return getObmSettings(identifier)
    .then(function(obmSettings) {
      return racadm.runCommand (
        obmSettings.host,
        obmSettings.user,
        obmSettings.password,
        'set bios.SysSecurity.SecureBoot ' + command
      );
    })
    .then(function() {
      res.status(202).json({"Message": "Successfully Completed Request"});
    })
    .catch(function(error) {
      return redfish.handleError(error, res);
    });
});

```

### 3) Checklist/TODO

Redfish Function	Resuse Bios workflow
getSecureBoot	Graph.Dell.Wsman.GetBios
setSecureBoot	Graph.Dell.Wsman.ConfigureBios