Overview of Dell Embedded Systems Management Using IPMI

March 2015

Author

Josh Moore
Enterprise Master Engineer
Server Solutions

Table of Contents

Introduction

What is IPMI?

The Intelligent Platform Management Interface (IPMI) is the standards-based systems management interface used by Dell PowerEdge servers. [1]

IPMI provides both local (In-band) and remote (Out-of-band) management and monitoring capability independent of the host system operating system or CPU.

The development of this interface specification was led by Intel Corporation and is supported by more than 200 computer systems vendors, such as Cisco, Dell, Hewlett-Packard, NEC Corporation, SuperMicro and Tyan. [2]

An IPMI sub-system consists of a main controller, called the baseboard management controller (BMC) and other management controllers distributed among different system modules that are referred to as satellite controllers. The satellite controllers within the same chassis connect to the BMC via the system interface called Intelligent Platform Management Bus/Bridge (IPMB) – an enhanced implementation of I²C (Inter-Integrated Circuit). The BMC connects to satellite controllers or another BMC in another chassis via the Intelligent Platform Management Controller (IPMC) bus or bridge. It may be managed with the Remote Management Control Protocol (RMCP), a specialized wire protocol defined by this specification. RMCP+ (a UDP-based protocol with stronger authentication than RMCP) is used for IPMI over LAN. [2]

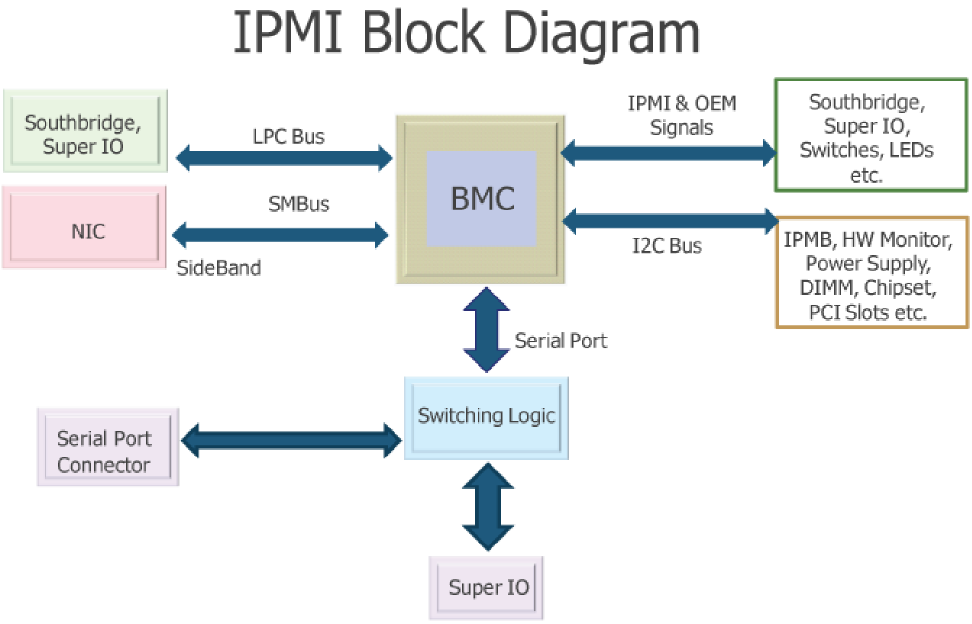


Figure 1 – Interfaces to the baseboard management controller (BMC) [2]

What can IPMI do for me?

IPMI is widely used for emergency maintenance, as well as provisioning and rollout of operating systems and applications. With features such as remote web access, SNMP alerts, LDAP support, remote console access and emulation of optical drives and other media, systems management with IPMI can provide complete control of a server even when the host operating system is down.

How does Dell implement IPMI?

- Prior to 8th generation PowerEdge servers, Embedded Server Management and Dell Remote Access Controllers implemented IPMI v1.0.
- 8th generation PowerEdge servers with BMC and DRAC4 implement IPMI v1.5.
- 9th-11th generation PowerEdge servers with iDRAC5/6 implement IPMI v2.0, while still supporting v1.5.
    - iDRAC6 firmware versions 1.98 (PowerEdge) and 3.65 (PowerEdge Blade), released Oct 2014 deprecated support for IPMI v1.5.
- 12th and 13th generation PowerEdge servers with iDRAC7/8 implement IPMI v2.0 only.
- All PowerEdge-C platforms support IPMI v1.5 and v2.0.

IPMI Specifications

The IPMI standard specification has evolved through a number of iterations:

- v1.0 was announced on September 16, 1998
- v1.5 was published on March 1, 2001

  Added features including

  - IPMI over LAN
  - IPMI over Serial/Modem
  - LAN Alerting
- v2.0 was published on February 14, 2004

  Added features including

  - Serial over LAN
  - Group Managed Systems
  - Enhanced Authentication
  - Firmware Firewall
  - VLAN Support
- v2.0 revision 1.1 was published on February 11, 2014

  Amended for errata, clarifications, and addenda, plus addition of support for IPv6

For additional information, including specification downloads and other related white papers, visit: http://www.intel.com/content/www/us/en/servers/ipmi/ipmi-home.html

IPMI Software and Drivers

In order to interact with IPMI capable devices you must install an IPMI application. Additionally for In-band management you must also install an IPMI driver.

Several IPMI applications are available. The most commonly seen are ipmitool, ipmiutil and freeipmi. OpenIPMI provides a Linux Driver and Library API. Microsoft provides a native driver for IPMI, however it may need to be manually installed.

For a comparison of the IPMI applications listed above, visit:
http://ipmiutil.sourceforge.net/docs/ipmisw-compare.htm

Each IPMI application has its own set of syntax for commands and functions. Please refer to individual application documentation for complete usage details.

For the purpose of this document we will focus on ipmitool/OpenIPMI for Linux and ipmiutil/Microsoft Generic IPMI Compliant Device driver for Windows.

Local IPMI is also available natively within VMware ESXi 5.x using esxcli, however functionality is limited to FRU, sensor and SEL data. There are other 3rd party applications, including an ipmitool for ESXi which are also available.

Installing IPMI in Linux

Most common Linux distributions include one or more IPMI applications, including those listed above, in their software repositories. Additionally rpm/deb packages and/or source code for installation are provided from the respective software websites.

The exact package version will vary by distribution.

Installing ipmitool

The official ipmitool project is maintained at:
http://sourceforge.net/projects/ipmitool/

For Red Hat Enterprise Linux (RHEL) and like derivatives such as CentOS or Scientific Linux, ipmitool can be installed from standard software repositories with the following command.

```
yum install ipmitool
```

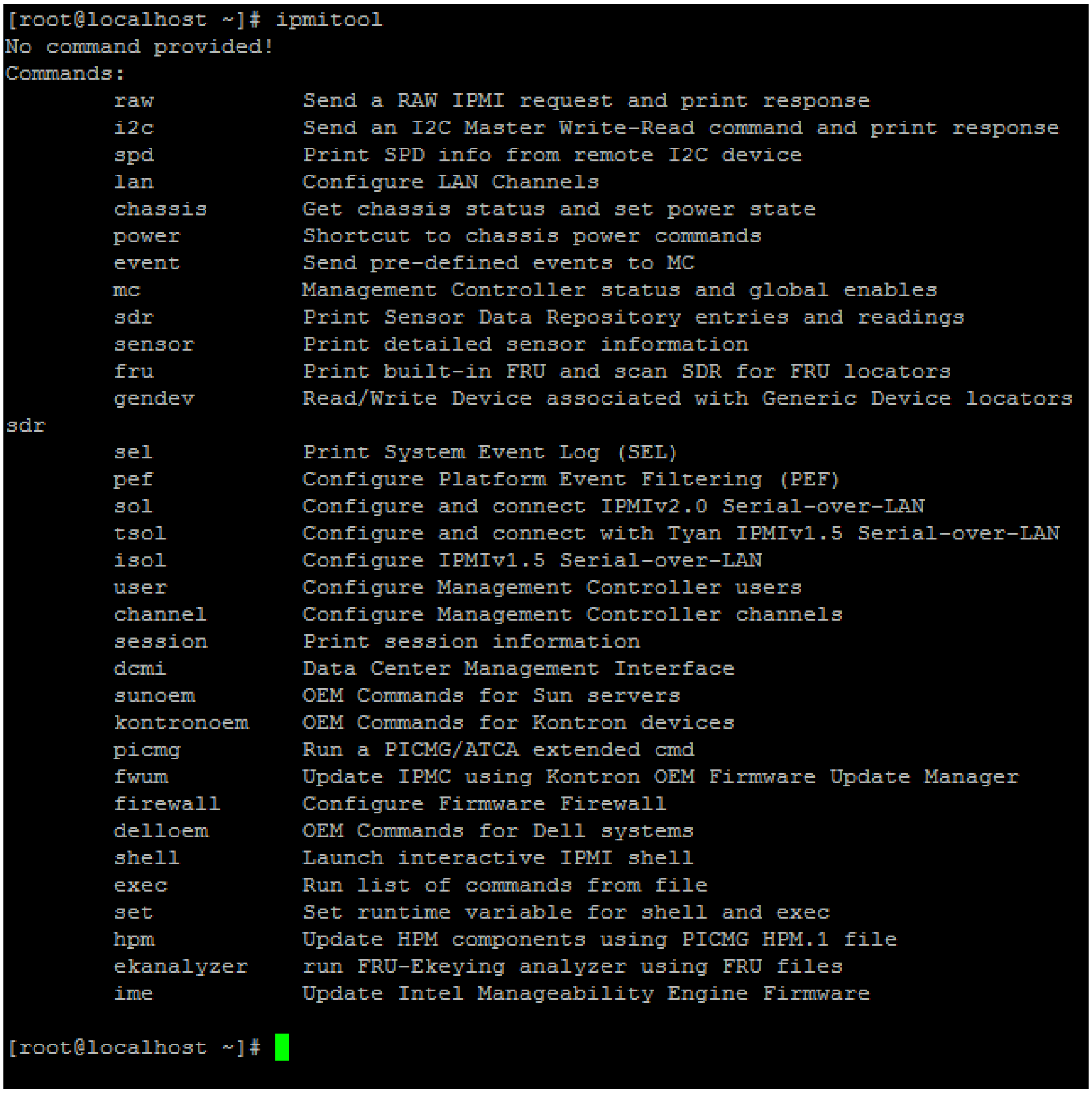
OpenIPMI is listed as a dependency of the ipmitool rpm package, and will be automatically installed if not already present on the system.

For Ubuntu, ipmitool can be installed from the standard software repositories with the following command.

```
sudo apt-get install ipmitool
```

Note for Ubuntu users, local ipmitool commands require sudo or root privileges (this does not apply to remote ipmitool commands).

Once ipmitool is installed (RHEL or Ubuntu), it will be located at:
`/usr/bin/ipmitool`



```
[root@localhost ~]# ipmitool
No command provided!
Commands:
        raw           Send a RAW IPMI request and print response
        i2c           Send an I2C Master Write-Read command and print response
        spd           Print SPD info from remote I2C device
        lan           Configure LAN Channels
        chassis       Get chassis status and set power state
        power         Shortcut to chassis power commands
        event         Send pre-defined events to MC
        mc            Management Controller status and global enables
        sdr           Print Sensor Data Repository entries and readings
        sensor        Print detailed sensor information
        fru           Print built-in FRU and scan SDR for FRU locators
        gendev        Read/Write Device associated with Generic Device locators
sdr
        sel           Print System Event Log (SEL)
        pef           Configure Platform Event Filtering (PEF)
        sol           Configure and connect IPMIv2.0 Serial-over-LAN
        tsol          Configure and connect with Tyan IPMIv1.5 Serial-over-LAN
        isol          Configure IPMIv1.5 Serial-over-LAN
        user          Configure Management Controller users
        channel       Configure Management Controller channels
        session       Print session information
        dcmi          Data Center Management Interface
        sunoem        OEM Commands for Sun servers
        kontronoem    OEM Commands for Kontron devices
        picmg         Run a PICMG/ATCA extended cmd
        fwum          Update IPMC using Kontron OEM Firmware Update Manager
        firewall      Configure Firmware Firewall
        delloem       OEM Commands for Dell systems
        shell         Launch interactive IPMI shell
        exec          Run list of commands from file
        set           Set runtime variable for shell and exec
        hpm           Update HPM components using PICMG HPM.1 file
        ekanalyzer    run FRU-Ekeying analyzer using FRU files
        ime           Update Intel Manageability Engine Firmware

[root@localhost ~]# █
```

Figure 2 – ipmitool basic usage

Note /usr/bin is generally included in $PATH environment variable, allowing us to invoke ipmitool without specifying the full path.

Installing OpenIPMI

The official OpenIPMI project is maintained at:
http://openipmi.sourceforge.net/

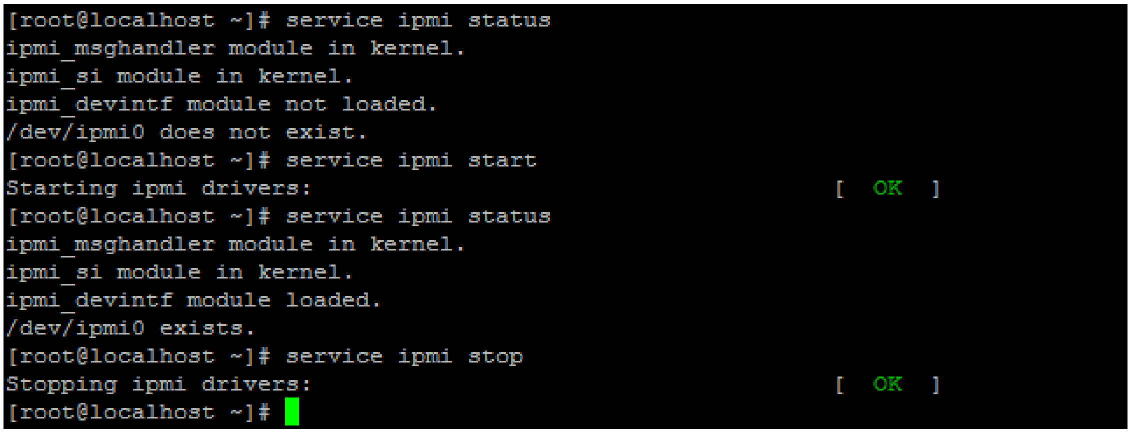
On RHEL and like derivatives, OpenIPMI can be installed from the standard software repositories with the following command.

```
yum install OpenIPMI
```

(case sensitive package name)

Once installed, you must load the driver before issuing any local IPMI commands. OpenIPMI includes startup scripts which can be controlled with the service command.

```
service ipmi start|stop|status|restart
```



```
[root@localhost ~]# service ipmi status
ipmi_msghandler module in kernel.
ipmi_si module in kernel.
ipmi_devintf module not loaded.
/dev/ipmi0 does not exist.
[root@localhost ~]# service ipmi start
Starting ipmi drivers:                                     [  OK  ]
[root@localhost ~]# service ipmi status
ipmi_msghandler module in kernel.
ipmi_si module in kernel.
ipmi_devintf module loaded.
/dev/ipmi0 exists.
[root@localhost ~]# service ipmi stop
Stopping ipmi drivers:                                     [  OK  ]
[root@localhost ~]#
```

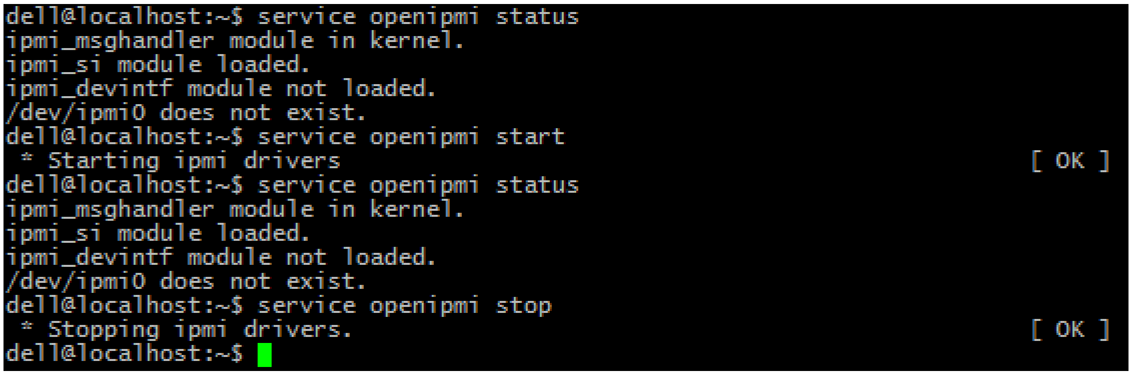
Figure 3 – Controlling IPMI driver service on RHEL

On Ubuntu, OpenIPMI can be installed from the standard software repositories with the following command.

```
sudo apt-get install openipmi
```

Once installed, you must load the driver before issuing any local IPMI commands. OpenIPMI includes startup scripts which can be controlled with the service command.

```
service openipmi start|stop|status|restart
```



```
dell@localhost:~$ service openipmi status
ipmi_msghandler module in kernel.
ipmi_si module loaded.
ipmi_devintf module not loaded.
/dev/ipmi0 does not exist.
dell@localhost:~$ service openipmi start
 * Starting ipmi drivers                                                 [ OK ]
dell@localhost:~$ service openipmi status
ipmi_msghandler module in kernel.
ipmi_si module loaded.
ipmi_devintf module not loaded.
/dev/ipmi0 does not exist.
dell@localhost:~$ service openipmi stop
 * Stopping ipmi drivers.                                                [ OK ]
dell@localhost:~$
```

Figure 4 – Controlling IPMI driver service on Ubuntu

While the IPIM driver can be configured to load automatically, using chkconfig for RHEL or sudo update-rc.d for Ubuntu, it is generally best practice to start/stop as needed.
The IPMI driver is not required to be started to send remote IPMI commands to a device or for a device to receive commands.

Installing IPMI in Windows

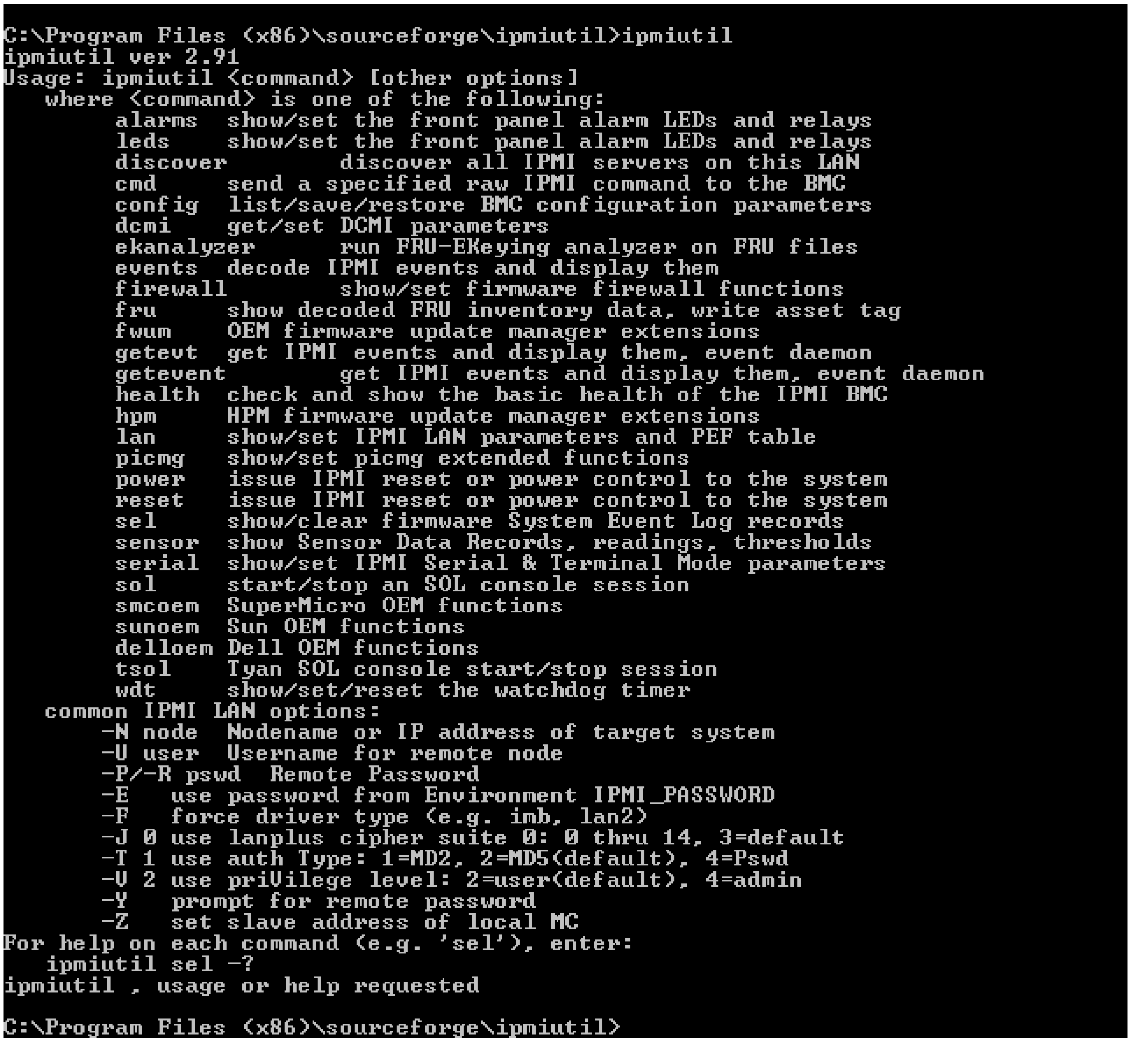
Windows users can use IPMI tools too! [4] Windows users can install ipmiutil as well as an IPMI compliant device driver for local and remote IPMI management.

Dell also provides a pre-compiled ipmitool for Windows as part of Remote Access Tools download.

Installing ipmiutil

The official ipmiutil project is maintained at:
http://ipmiutil.sourceforge.net/

The installation for ipmiutil is a standard MSI installation wizard. By default it is installed to:
`C:\Program Files (x86)\sourceforge\ipmiutil\`



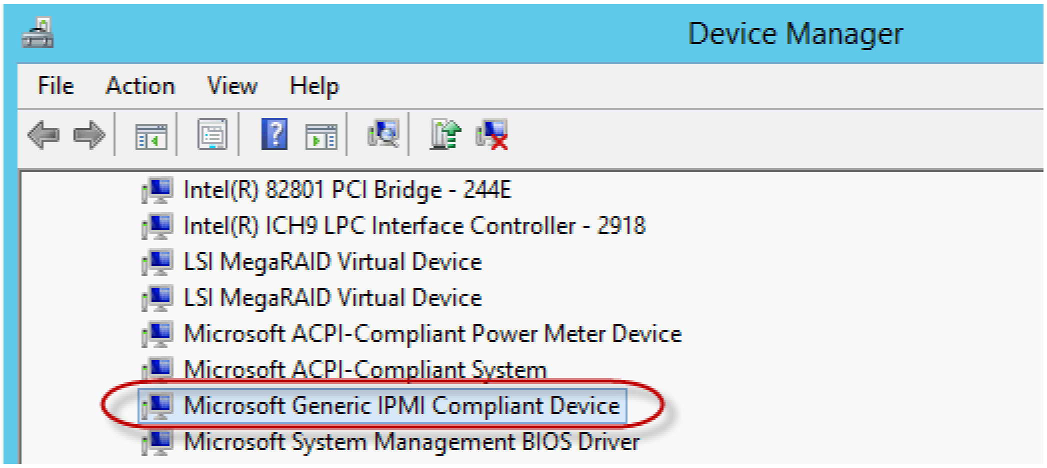
```
C:\Program Files (x86)\sourceforge\ipmiutil>ipmiutil
ipmiutil ver 2.91
Usage: ipmiutil <command> [other options]
   where <command> is one of the following:
        alarms  show/set the front panel alarm LEDs and relays
        leds    show/set the front panel alarm LEDs and relays
        discover        discover all IPMI servers on this LAN
        cmd     send a specified raw IPMI command to the BMC
        config  list/save/restore BMC configuration parameters
        dcmi    get/set DCMI parameters
        ekanalyzer      run FRU-EKeying analyzer on FRU files
        events  decode IPMI events and display them
        firewall        show/set firmware firewall functions
        fru     show decoded FRU inventory data, write asset tag
        fwum    OEM firmware update manager extensions
        getevt  get IPMI events and display them, event daemon
        getevent        get IPMI events and display them, event daemon
        health  check and show the basic health of the IPMI BMC
        hpm     HPM firmware update manager extensions
        lan     show/set IPMI LAN parameters and PEF table
        picmg   show/set picmg extended functions
        power   issue IPMI reset or power control to the system
        reset   issue IPMI reset or power control to the system
        sel     show/clear firmware System Event Log records
        sensor  show Sensor Data Records, readings, thresholds
        serial  show/set IPMI Serial & Terminal Mode parameters
        sol     start/stop an SOL console session
        smcoem  SuperMicro OEM functions
        sunoem  Sun OEM functions
        delloem Dell OEM functions
        tsol    Tyan SOL console start/stop session
        wdt     show/set/reset the watchdog timer
   common IPMI LAN options:
       -N node  Nodename or IP address of target system
       -U user  Username for remote node
       -P/-R pswd  Remote Password
       -E   use password from Environment IPMI_PASSWORD
       -F   force driver type (e.g. imb, lan2)
       -J 0 use lanplus cipher suite 0: 0 thru 14, 3=default
       -T 1 use auth Type: 1=MD2, 2=MD5(default), 4=Pswd
       -V 2 use priVilege level: 2=user(default), 4=admin
       -Y   prompt for remote password
       -Z   set slave address of local MC
For help on each command (e.g. 'sel'), enter:
   ipmiutil sel -?
ipmiutil , usage or help requested

C:\Program Files (x86)\sourceforge\ipmiutil>
```

Figure 5 – ipmiutil basic usage

Installing Microsoft Generic IPMI Compliant Device

Microsoft provides a native IPMI driver, however it may need to be manually installed.

Check Device Manager and look under System Devices to see if it is already installed.



Intel(R) 82801 PCI Bridge - 244E
Intel(R) ICH9 LPC Interface Controller - 2918
LSI MegaRAID Virtual Device
LSI MegaRAID Virtual Device
Microsoft ACPI-Compliant Power Meter Device
Microsoft ACPI-Compliant System
Microsoft Generic IPMI Compliant Device
Microsoft System Management BIOS Driver

Figure 6 – Windows Device Manager showing IPMI driver

To install the driver:

1. From Device Manager, select Action > Add legacy hardware
2. Follow the wizard to manually select from a list
3. Select System devices > Microsoft > Microsoft Generic IPMI Compliant Device

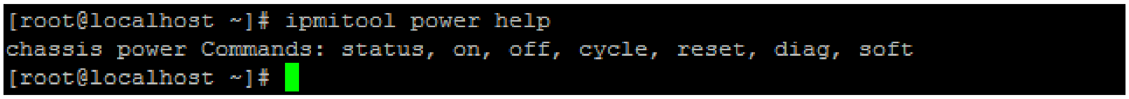
In-band Management

With the appropriate IPMI application and driver installed, we can now perform local IPMI commands to query, configure or control the BMC.

Local ipmitool Examples

As seen above in Figure 2, there are many commands available within ipmitool, each with various subcommands. For help on individual commands type:

```
ipmitool <command> help
```



```
[root@localhost ~]# ipmitool power help
chassis power Commands: status, on, off, cycle, reset, diag, soft
[root@localhost ~]#
```

Figure 7 – ipmitool command help syntax

Additional command line switches are also available, and can be viewed by typing:

```
ipmitool --help
```

Many query commands within ipmitool can generate more verbose output with -v (can use multiple times):

```
ipmitool <command> <subcommand> -v
ipmitool <command> <subcommand> -vvv
```

A man page is also available for ipmitool

Common tasks with ipmitool:

Display IPMI LAN settings

```
ipmitool lan print
```

Configure IPMI LAN settings

```
ipmitool lan set 1 ipsrc static|dhcp
ipmitool lan set 1 ipaddr <ipaddress>
ipmitool lan set 1 netmask <netmask>
ipmitool lan set 1 defgw ipaddr <gatewayip>
```

Host power control

```
ipmitool power status|on|off|reset|cycle
```

Show System Event Log (SEL)

```
ipmitool sel list
ipmitool sel list -v
```

Show sensor data

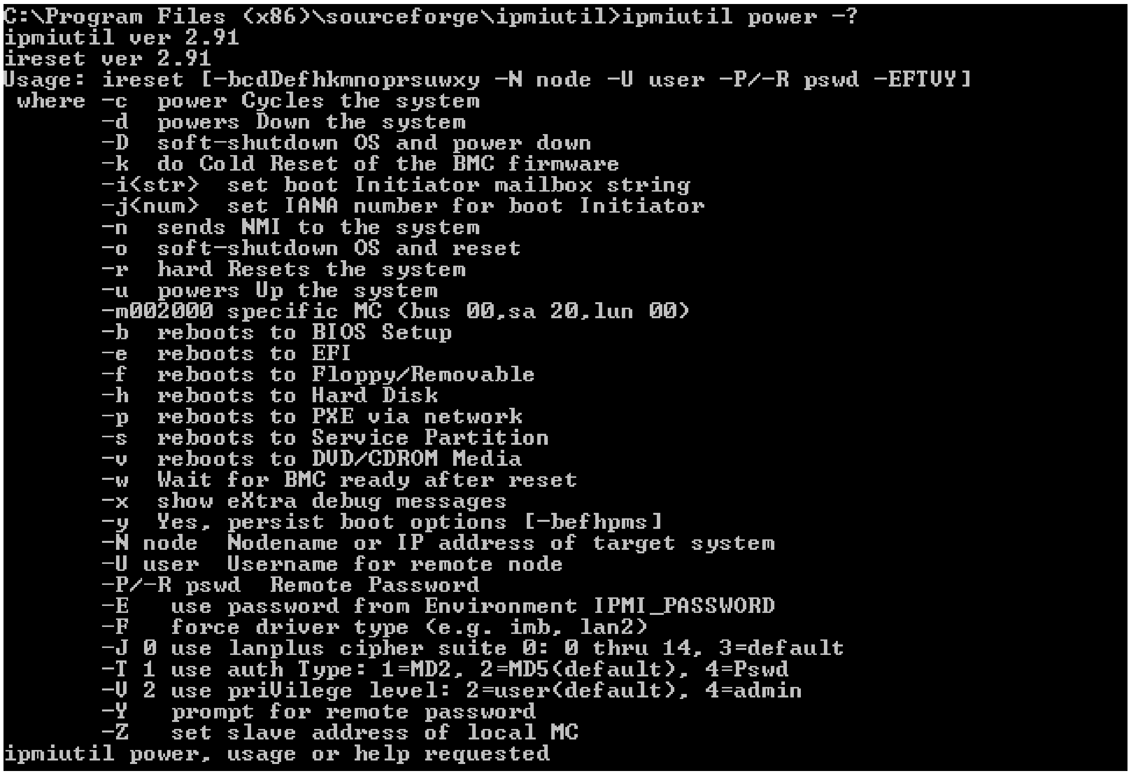
```
ipmitool sdr
ipmitool sensor
```

See the ipmitool man page or online documentation for other commands.

Local ipmiutil Examples

As seen in Figure 5, ipmiutil has many of the same command options, but the command syntax will vary from ipmitool. For help on individual command type:

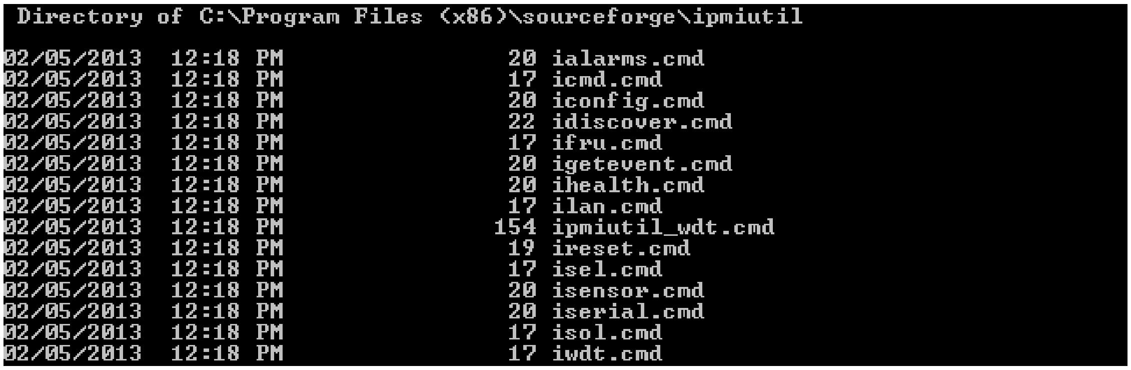
```
ipmiutil <command> -?
```



```
C:\Program Files (x86)\sourceforge\ipmiutil>ipmiutil power -?
ipmiutil ver 2.91
ireset ver 2.91
Usage: ireset [-bcdDefhkmnoprsuwxy -N node -U user -P/-R pswd -EFTVY]
 where -c  power Cycles the system
       -d  powers Down the system
       -D  soft-shutdown OS and power down
       -k  do Cold Reset of the BMC firmware
       -i<str>  set boot Initiator mailbox string
       -j<num>  set IANA number for boot Initiator
       -n  sends NMI to the system
       -o  soft-shutdown OS and reset
       -r  hard Resets the system
       -u  powers Up the system
       -m002000 specific MC (bus 00,sa 20,lun 00)
       -b  reboots to BIOS Setup
       -e  reboots to EFI
       -f  reboots to Floppy/Removable
       -h  reboots to Hard Disk
       -p  reboots to PXE via network
       -s  reboots to Service Partition
       -v  reboots to DVD/CDROM Media
       -w  Wait for BMC ready after reset
       -x  show eXtra debug messages
       -y  Yes, persist boot options [-befhpms]
       -N node  Nodename or IP address of target system
       -U user  Username for remote node
       -P/-R pswd  Remote Password
       -E   use password from Environment IPMI_PASSWORD
       -F   force driver type (e.g. imb, lan2)
       -J 0 use lanplus cipher suite 0: 0 thru 14, 3=default
       -T 1 use auth Type: 1=MD2, 2=MD5(default), 4=Pswd
       -V 2 use priVilege level: 2=user(default), 4=admin
       -Y   prompt for remote password
       -Z   set slave address of local MC
ipmiutil power, usage or help requested
```

Figure 8 – ipmiutil command help syntax

Several command shortcuts are provided by ipmiutil. For example: as seen in the usage syntax of the command help output in Figure 8, instead of typing ipmiutil power we can use ireset followed by the appropriate subcommand.



```
 Directory of C:\Program Files (x86)\sourceforge\ipmiutil

02/05/2013  12:18 PM                20 ialarms.cmd
02/05/2013  12:18 PM                17 icmd.cmd
02/05/2013  12:18 PM                20 iconfig.cmd
02/05/2013  12:18 PM                22 idiscover.cmd
02/05/2013  12:18 PM                17 ifru.cmd
02/05/2013  12:18 PM                20 igetevent.cmd
02/05/2013  12:18 PM                20 ihealth.cmd
02/05/2013  12:18 PM                17 ilan.cmd
02/05/2013  12:18 PM               154 ipmiutil_wdt.cmd
02/05/2013  12:18 PM                19 ireset.cmd
02/05/2013  12:18 PM                17 isel.cmd
02/05/2013  12:18 PM                20 isensor.cmd
02/05/2013  12:18 PM                20 iserial.cmd
02/05/2013  12:18 PM                17 isol.cmd
02/05/2013  12:18 PM                17 iwdt.cmd
```

Figure 9 – ipmiutil command shortcuts

See individual command help to see which commands support verbose output.

Common tasks with ipmiutil:

Display IPMI LAN settings

```
ipmiutil lan
```

Configure IPMI LAN settings

```
ipmiutil lan –e –I <ipaddress>
ipmiutil lan –e –S <netmask>
ipmiutil lan –e –G <gatewayip>
```

To configure for DHCP

```
ipmiutil lan –d
ipmiutil lan –e -D
```

Host power control

```
ipmiutil power –c|d|D|u…
```

Show System Event Log (SEL)

```
ipmiutil sel
```

Show sensor data

```
ipmiutil sensor
ipmiutil sensor –v
ipmiutil fru
```

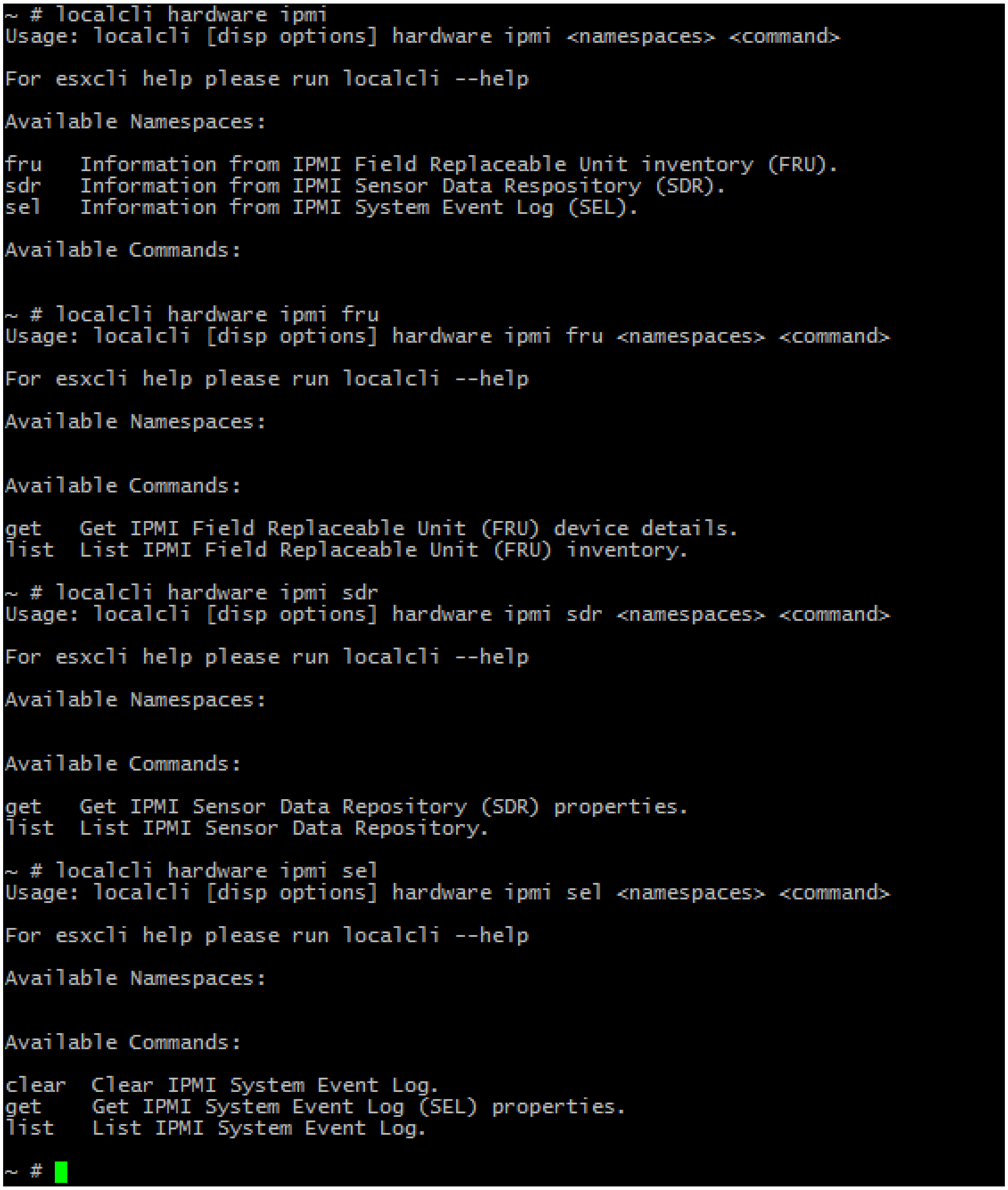
See ipmiutil user guide or online documentation for other commands.

ESXi localcli Examples

VMware provides some monitoring for IPMI information natively. For example SEL events can be seen within vSphere. Local IPMI commands are possible using esxcli.

To get IPMI information use the following syntax:

```
localcli hardware ipmi <namespace> <command>
```



```
~ # localcli hardware ipmi
Usage: localcli [disp options] hardware ipmi <namespaces> <command>

For esxcli help please run localcli --help

Available Namespaces:

fru   Information from IPMI Field Replaceable Unit inventory (FRU).
sdr   Information from IPMI Sensor Data Respository (SDR).
sel   Information from IPMI System Event Log (SEL).

Available Commands:


~ # localcli hardware ipmi fru
Usage: localcli [disp options] hardware ipmi fru <namespaces> <command>

For esxcli help please run localcli --help

Available Namespaces:


Available Commands:

get   Get IPMI Field Replaceable Unit (FRU) device details.
list  List IPMI Field Replaceable Unit (FRU) inventory.

~ # localcli hardware ipmi sdr
Usage: localcli [disp options] hardware ipmi sdr <namespaces> <command>

For esxcli help please run localcli --help

Available Namespaces:


Available Commands:

get   Get IPMI Sensor Data Repository (SDR) properties.
list  List IPMI Sensor Data Repository.

~ # localcli hardware ipmi sel
Usage: localcli [disp options] hardware ipmi sel <namespaces> <command>

For esxcli help please run localcli --help

Available Namespaces:


Available Commands:

clear  Clear IPMI System Event Log.
get    Get IPMI System Event Log (SEL) properties.
list   List IPMI System Event Log.

~ #
```

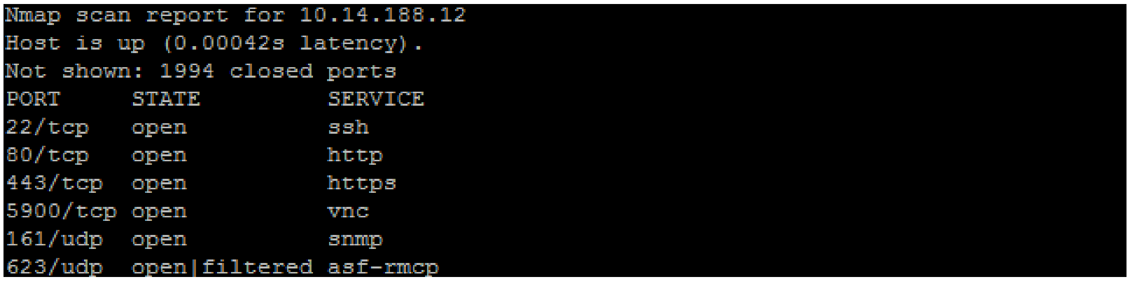
Figure 10 – esxcli IPMI examples

Out-of-Band Management

One of the great things about IPMI is that all of the commands available work whether you are issuing them against a local BMC, or remotely to a BMC over a network.

For remote IPMI commands, all that is required is to add the additional command line options to designate your remote device and authentication.

Remote IPMI uses UDP port 623. Other ports typically used on Dell BMC and DRAC are TCP 22, 80, 443, 5900 and UDP 161. Exact ports will vary by device and configuration.

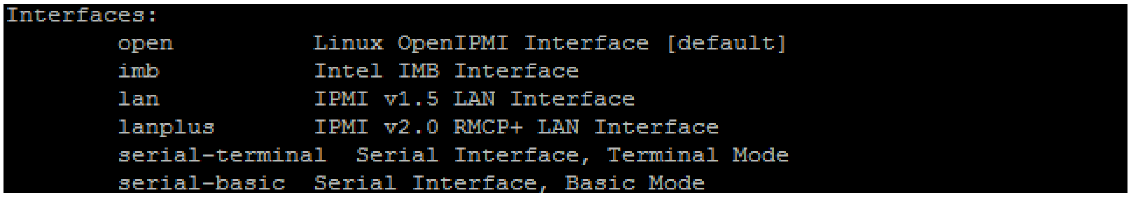


```
Nmap scan report for 10.14.188.12
Host is up (0.00042s latency).
Not shown: 1994 closed ports
PORT     STATE         SERVICE
22/tcp   open          ssh
80/tcp   open          http
443/tcp  open          https
5900/tcp open          vnc
161/udp  open          snmp
623/udp  open|filtered asf-rmcp
```

Figure 11 – example port scan of an iDRAC7

Remote ipmitool Examples

There are multiple interfaces supported by ipmitool, to work with specific specifications.



```
Interfaces:
        open          Linux OpenIPMI Interface [default]
        imb           Intel IMB Interface
        lan           IPMI v1.5 LAN Interface
        lanplus       IPMI v2.0 RMCP+ LAN Interface
        serial-terminal  Serial Interface, Terminal Mode
        serial-basic  Serial Interface, Basic Mode
```

Figure 12 – snip from ipmitool --help

By default, ipmitool uses the open interface, which will work when issuing local commands to either IPMI v1.5 or v2.0 devices.

When issuing a remote ipmitool command however it uses the lan interface, or IPMI v1.5, which will fail if the remote device in question does not support IPMI v1.5.

For Dell BMC/iDRAC devices, depending on the version of IPMI supported, use either of the following examples:

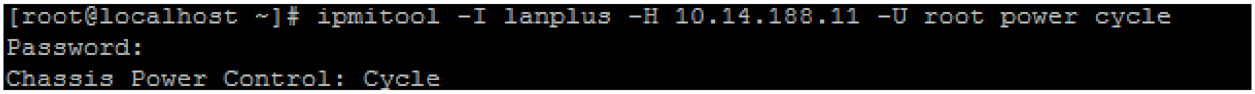
IPMI v1.5

```
ipmitool -H <ipaddress> -U <user> -P <password> <command>
```

IPMI v2.0

```
ipmitool -I lanplus -H <ipaddress> -U <user> -P <password> <command>
```

The -P may be left off the command line to be prompted for password, thus masking the password.



```
[root@localhost ~]# ipmitool -I lanplus -H 10.14.188.11 -U root power cycle
Password:
Chassis Power Control: Cycle
```

Figure 13 – Remote ipmitool example using IPMI v2.0 w/password prompt

IPMI v2.0 provides support for varying authentication, confidentiality and integrity mechanisms through cipher suites. By default, ipmitool uses cipher suite 3 for IPMI v2.0 (lanplus) commands.
You can specify a different cipher suite with -C using the following syntax:

```
ipmitool -I lanplus -C <0-14> -H <ipaddress> -U <user> <command>
```

For additional detail on cipher suites, including how to show or set cipher suite privileges, see IPMI Security section.

Remote ipmiutil Examples

Multiple interfaces are also supported by ipmiutil, however ipmiutil will attempt to detect any available interface type and use it.

The -F option is used to force a specific interface type to one of the following: imb, va, open, gnu, landesk, lan, lan2, lan2i, kcs, smb.

For Dell BMC/iDRAC devices, use the following examples:

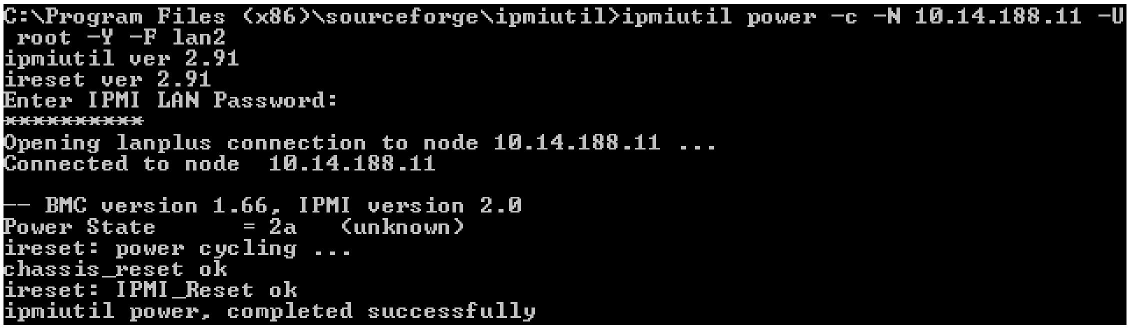
IPMI v1.5

```
ipmiutil <command> -N <ipaddress> -U <user> -P <password> -F lan
```

IPMI v2.0

```
ipmiutil <command> -N <ipaddress> -U <user> -P <password –F lanplus
```

To be prompted for the password instead of providing it on the command line, use -Y instead of –P.



```
C:\Program Files (x86)\sourceforge\ipmiutil>ipmiutil power -c -N 10.14.188.11 -U
 root -Y -F lan2
ipmiutil ver 2.91
ireset ver 2.91
Enter IPMI LAN Password:
***********
Opening lanplus connection to node 10.14.188.11 ...
Connected to node  10.14.188.11

-- BMC version 1.66, IPMI version 2.0
Power State       = 2a   (unknown)
ireset: power cycling ...
chassis_reset ok
ireset: IPMI_Reset ok
ipmiutil power, completed successfully
```

Figure 14 – Remote ipmiutil example using IPMI v2.0 w/password prompt

By default, ipmiutil uses cipher suite 3 for IPMI v2.0.
You can specify a different cipher suite with -J using the following syntax:

```
ipmiutil <command> -N <ipaddress> -U user –Y –F lanplus –J <0-14>
```

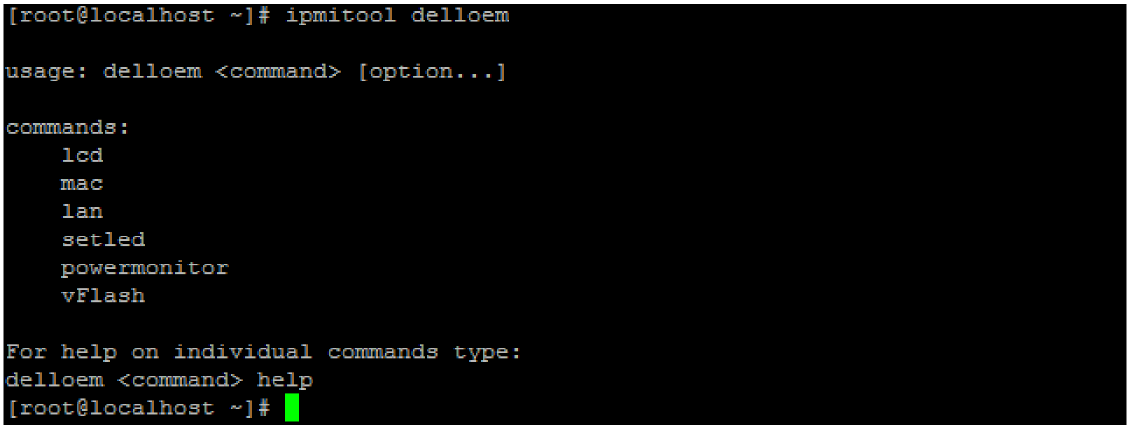
For additional detail on cipher suites, including how to show or set cipher suite privileges, see IPMI Security section.

DellOEM Specific Features

In addition to standards laid out in specifications, IPMI is also easily extensible, which allows OEMs to differentiate their products with platform specific functionality. [3]
For Dell, much of this functionality is wrapped into RACADM.

Both ipmitool and ipmiutil (as well as others like freeipmi) provide some support for various OEM extensions. To access delloem options use the following syntax:

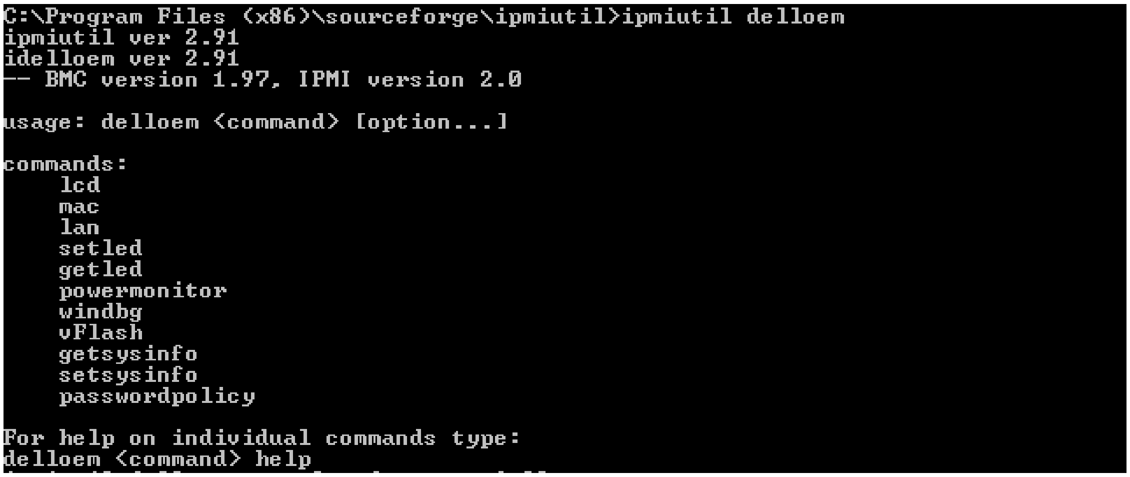
ipmitool
`ipmitool delloem`



```
[root@localhost ~]# ipmitool delloem

usage: delloem <command> [option...]

commands:
    lcd
    mac
    lan
    setled
    powermonitor
    vFlash

For help on individual commands type:
delloem <command> help
[root@localhost ~]#
```

Figure 15 – ipmitool delloem command options

ipmiutil
`ipmiutil delloem`

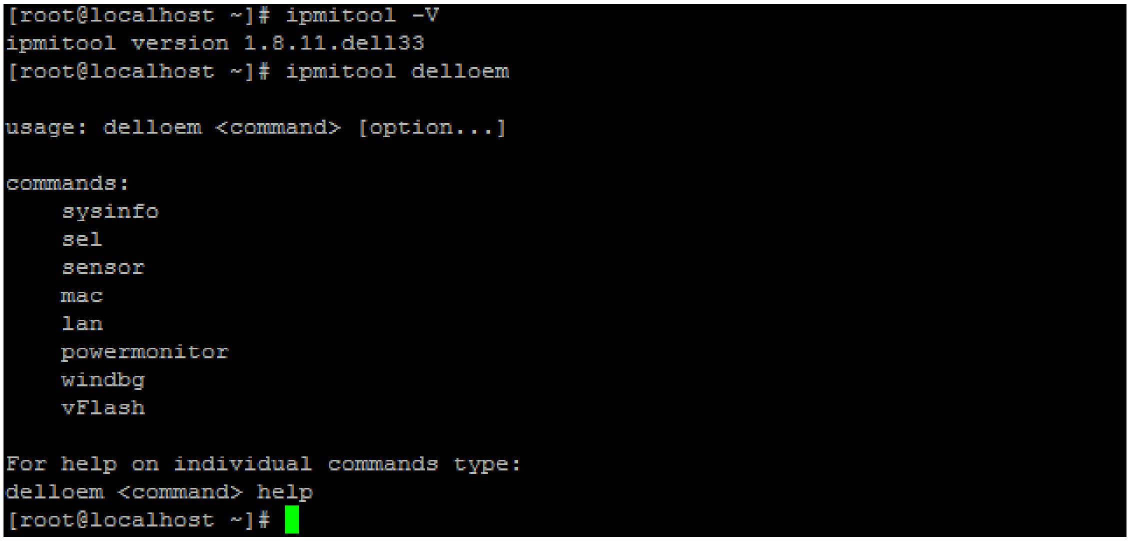


```
C:\Program Files (x86)\sourceforge\ipmiutil>ipmiutil delloem
ipmiutil ver 2.91
idelloem ver 2.91
-- BMC version 1.97, IPMI version 2.0

usage: delloem <command> [option...]

commands:
    lcd
    mac
    lan
    setled
    getled
    powermonitor
    windbg
    vFlash
    getsysinfo
    setsysinfo
    passwordpolicy

For help on individual commands type:
delloem <command> help
```

Figure 16 – ipmiutil delloem command options

Note: Exact delloem functionality will vary by application version.

In addition to the standard ipmitool application, Dell offers a modified package with additional support for delloem commands.



```
[root@localhost ~]# ipmitool -V
ipmitool version 1.8.11.dell33
[root@localhost ~]# ipmitool delloem

usage: delloem <command> [option...]

commands:
    sysinfo
    sel
    sensor
    mac
    lan
    powermonitor
    windbg
    vFlash

For help on individual commands type:
delloem <command> help
[root@localhost ~]#
```

Figure 17 – Dell modified version of ipmitool

The Dell version of ipmitool is available on Linux via the Dell Linux Repository and as part of the Dell OpenManage BMC Utility for Windows or Dell OpenManage Remote Access Utilities for Linux downloads.

IPMI Raw Hex Commands

While a fair amount of functionality is built into the IPMI applications using friendly command structure, there is a lot of additional functionality that relies upon raw hex based commands.

IPMI raw hex values are very platform specific. Even within the same OEM, the raw hex for one platform may vary from the raw hex of another for the same function. Unfortunately little documentation exists regarding the raw hex strings for the commands and their options.

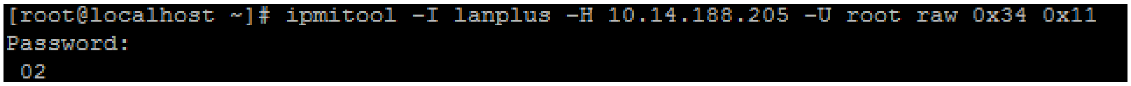
IPMI Raw Hex Examples

Some raw hex commands have been documented for very specific functions which commonly come up.

The following examples only apply to the platforms indicated and are provided to show raw hex syntax examples.

Identify PowerEdge C6100 Slot

`ipmitool raw 0x34 0x11`



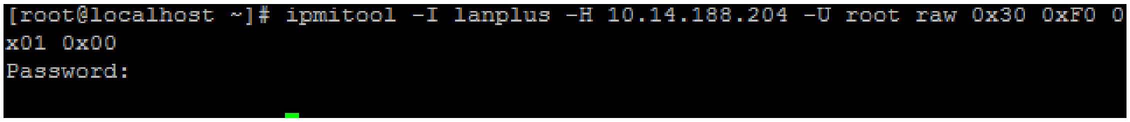
```
[root@localhost ~]# ipmitool -I lanplus -H 10.14.188.205 -U root raw 0x34 0x11
Password:
 02
```

Figure 18 – ipmitool raw command example, identify C6100 slot #

Toggle Power State of C410x GPU Slot #1

`ipmitool –H <ipaddress> -U root raw 0x30 0xF0 0x01 0x00`

The above example would flip a GPU from off to on, or on to off. To power cycle a GPU, execute the command twice with a brief pause between commands.
To toggle power on another slot, change the last two hex values, for example slot #16 0x00 0x80. For full slot hex value listing see KB article SLN244176.

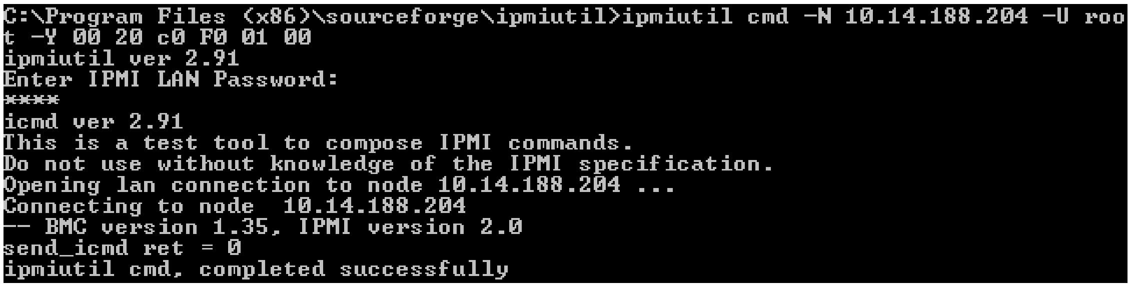


```
[root@localhost ~]# ipmitool -I lanplus -H 10.14.188.204 -U root raw 0x30 0xF0 0
x01 0x00
Password:
```

Figure 19 – ipmitool raw command example, toggle power of C410x GPU slot#1

As in the above figure, not all commands will provide output for feedback.

It is important to note that syntax with raw hex commands in ipmiutil are different.



```
C:\Program Files (x86)\sourceforge\ipmiutil>ipmiutil cmd -N 10.14.188.204 -U roo
t -Y 00 20 c0 F0 01 00
ipmiutil ver 2.91
Enter IPMI LAN Password:
****
icmd ver 2.91
This is a test tool to compose IPMI commands.
Do not use without knowledge of the IPMI specification.
Opening lan connection to node 10.14.188.204 ...
Connecting to node  10.14.188.204
-- BMC version 1.35, IPMI version 2.0
send_icmd ret = 0
ipmiutil cmd, completed successfully
```

Figure 20 – ipmiutil raw command comparison, toggle power of C410x slot#1

While raw hex is sometimes required, IPMI wrapper applications exist which generally fulfill such needs with their own friendly command structure.

IPMI Wrappers

Several wrappers exist to reduce complexity of IPMI systems management. These are particularly useful when dealing with tasks that would otherwise require knowledge of the specific raw hex for specific tasks. For Dell platforms we have RACADM for PowerEdge and BMC-tool for PowerEdge-C.

RACADM

RACADM allows for remote or local management via iDRAC, DRAC and you can also manage Dell Chassis Management Controllers (CMC) which do not support direct IPMI via the tools referenced above.

RACADM has extensive documentation. For additional information on RACADM visit the following resources:

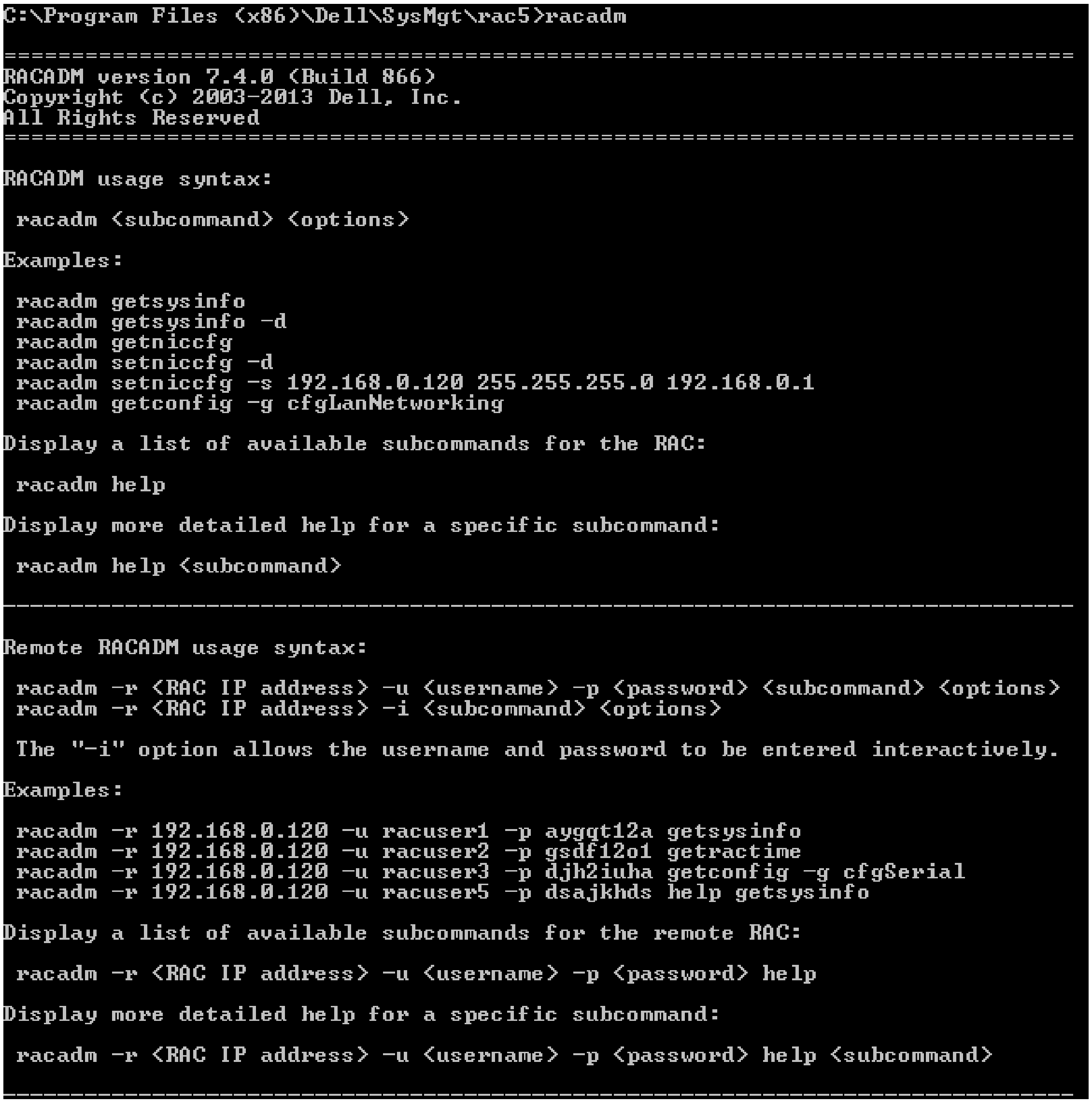
http://en.community.dell.com/techcenter/systems-management/w/wiki/3205.racadm-command-line-interface-for-drac

Additionally, consult the RACADM Command Line Reference Guide for your specific iDRAC, DRAC or CMC.

RACADM can be installed with OpenManage Server Administrator or Remote Access Utilities for Windows and Linux. Check downloads for specific PowerEdge model to get the latest supported version.

Remote RACADM commands run over port 443 and authenticates against the web service. For this reason remote RACADM can be used even while IPMI over LAN is disabled.

RACADM Examples



```
C:\Program Files (x86)\Dell\SysMgt\rac5>racadm

===============================================================================
RACADM version 7.4.0 (Build 866)
Copyright (c) 2003-2013 Dell, Inc.
All Rights Reserved
===============================================================================

RACADM usage syntax:

 racadm <subcommand> <options>

Examples:

 racadm getsysinfo
 racadm getsysinfo -d
 racadm getniccfg
 racadm setniccfg -d
 racadm setniccfg -s 192.168.0.120 255.255.255.0 192.168.0.1
 racadm getconfig -g cfgLanNetworking

Display a list of available subcommands for the RAC:

 racadm help

Display more detailed help for a specific subcommand:

 racadm help <subcommand>

-------------------------------------------------------------------------------

Remote RACADM usage syntax:

 racadm -r <RAC IP address> -u <username> -p <password> <subcommand> <options>
 racadm -r <RAC IP address> -i <subcommand> <options>

 The "-i" option allows the username and password to be entered interactively.

Examples:

 racadm -r 192.168.0.120 -u racuser1 -p aygqt12a getsysinfo
 racadm -r 192.168.0.120 -u racuser2 -p gsdf12o1 getractime
 racadm -r 192.168.0.120 -u racuser3 -p djh2iuha getconfig -g cfgSerial
 racadm -r 192.168.0.120 -u racuser5 -p dsajkhds help getsysinfo

Display a list of available subcommands for the remote RAC:

 racadm -r <RAC IP address> -u <username> -p <password> help

Display more detailed help for a specific subcommand:

 racadm -r <RAC IP address> -u <username> -p <password> help <subcommand>

-------------------------------------------------------------------------------
```

Figure 21 – RACADM use syntax and examples

BMC-tool

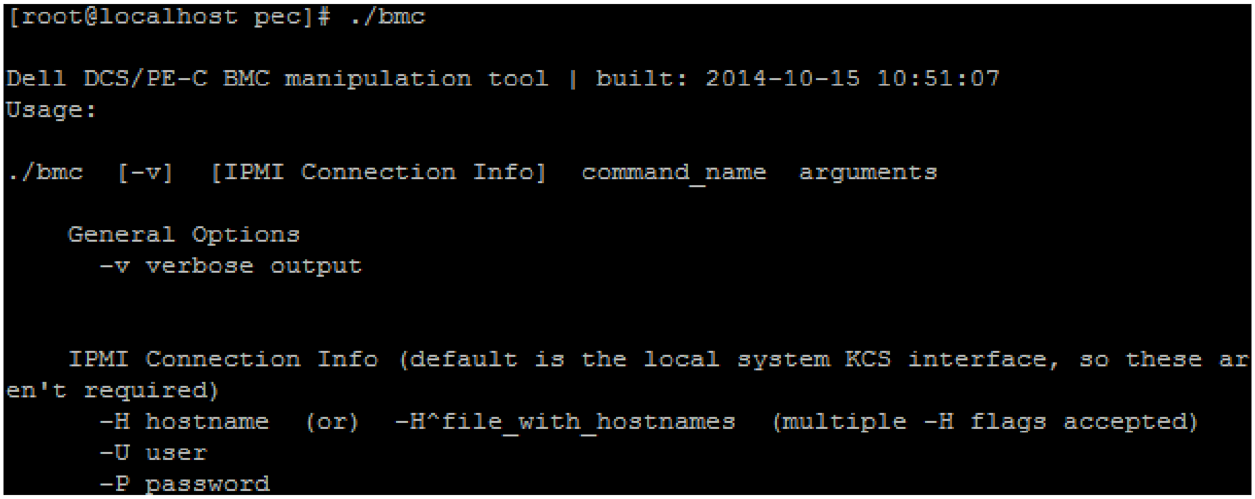
BMC-tool allows for remote or local management for PowerEdge-C BMC. BMC-tool is only available for Linux, and requires ipmitool to be installed.

Documentation for BMC-tool functionality is built within the tool. Running BMC-tool without the complete command or arguments will display the available commands or arguments that are relative.

BMC-tool can be downloaded as an RPM package for RHEL or like derivatives or as a compressed TAR file for other Linux distributions from:

http://poweredgec.dell.com/

BMC-tool Examples



```
[root@localhost pec]# ./bmc

Dell DCS/PE-C BMC manipulation tool | built: 2014-10-15 10:51:07
Usage:

./bmc  [-v]  [IPMI Connection Info]  command_name  arguments

    General Options
      -v verbose output


    IPMI Connection Info (default is the local system KCS interface, so these ar
en't required)
      -H hostname  (or)  -H^file_with_hostnames  (multiple -H flags accepted)
      -U user
      -P password
```

Figure 22 – BMC-tool usage example

IPMI Security

Properly configuring IPMI systems management is crucial for system security. The same power that makes IPMI so useful can leave the system wide open if not properly secured.

Best Practices

DRAC's are intended to be on a separate management network; they are not designed nor intended to be placed on or connected to the internet. Doing so could expose the connected system to security and other risks for which Dell is not responsible.

Along with locating DRACs on a separate management subnet, users should isolate the management subnet/vLAN with technologies such as firewalls, and limit access to the subnet/vLAN to authorized server administrators.

Dell recommends the following general networking best practices for BMC/IPMI:

- Change the default username/password.
- Segment IPMI traffic (UDP and stateless) from the rest of the network.
- Do not allow IPMI traffic from outside the network.
- If using IPMI v1.5-capable BMCs, use ACLs and strict source routing to help ensure the IPMI traffic is secure. IPMI v2.0 uses stronger encryption than IPMI v1.5.
- Segment SNMP interfaces on managed servers using virtual LANs (*VLANS*), access control lists (ACLs)
- Allows UDP port 623 (for IPMI over LAN)
- Allow TCP port 80 and 443 (HTTP and HTTPS respectively).
- Filter TCP port 25 (Use ACLs to limit port 25 traffic to the mail servers).
- Authentication should be required (*see below for steps to disallow bypassing authentication, also known as cipher suite 0*)
- A strong password should be used. Use of NULL passwords should not be allowed (*Dell DRAC/iDRACs do not allow the use of NULL passwords*).
- Anonymous logons should not be allowed *(Anonymous logons are NOT allowed by default on Dell DRAC/iDRACs. User account 1 (the anonymous user account) is disabled with no way to enable this account).*

IPMI v2.0 Cipher Suite 0

Cipher Suite 0 is often misrepresented as a serious failing of the IPMI v2.0 specification, allowing remote access to IPMI without correct authentication.
While cipher suite 0 can be exploited, it is in fact documented functionality of the IPMI v2.0 specification.

As found in the latest revision of the IPMI v2.0 Rev 1-1 specification, the characteristics of cipher suite 0 is no password, no authentication algorithm, no integrity algorithm and no confidentiality algorithm.

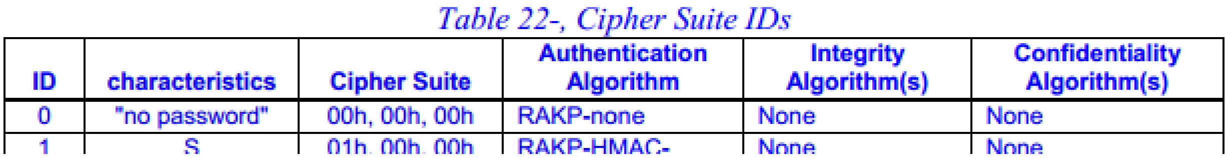
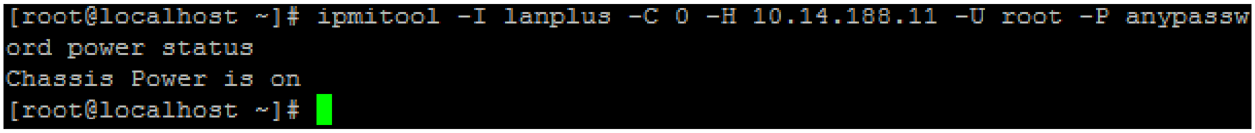


Table 22-, Cipher Suite IDs

ID	characteristics	Cipher Suite	Authentication Algorithm	Integrity Algorithm(s)	Confidentiality Algorithm(s)
0	"no password"	00h, 00h, 00h	RAKP-none	None	None
1	S	01h, 00h, 00h	RAKP-HMAC-	None	None

Figure 23 – cipher suite 0 characteristics [5]

While some IPMI v2.0 capable devices disable cipher suite 0 by default, many capable devices do not. Dell has changed default behavior from the factory, or when resetting configurations on iDRAC to be disabled. However, systems that shipped before this will still be enabled if the administrator has not manually disabled it.



```
[root@localhost ~]# ipmitool -I lanplus -C 0 -H 10.14.188.11 -U root -P anypassw
ord power status
Chassis Power is on
[root@localhost ~]#
```

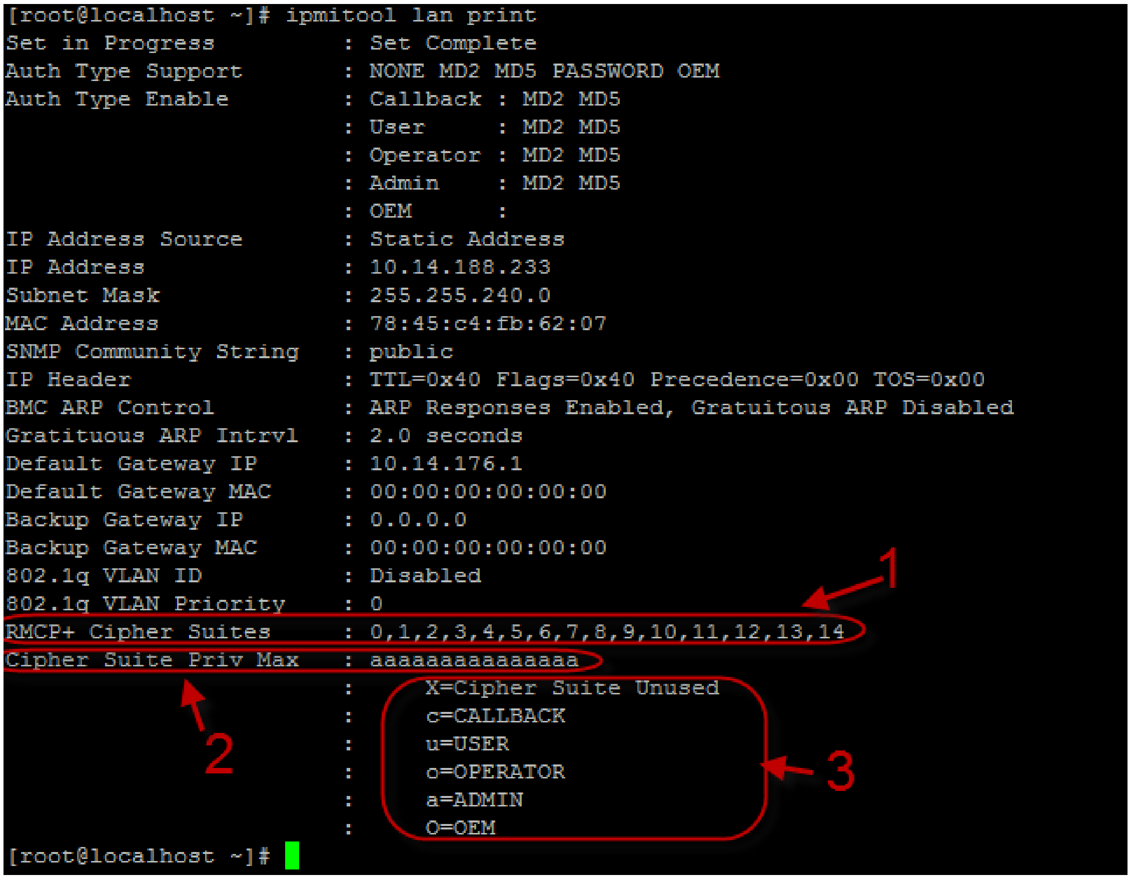
Figure 24 – ipmitool cipher suite 0 example

While ipmitool requires password input, any password string will be accepted.

Using cipher suite 0, any and all commands are possible, such as querying system information, changing power states or configuring new or existing users.
For example, an attacker could use cipher suite 0 to create a new administrative user and then access the web interface and gain full console access to the system.

You can check cipher suite privileges by pulling IPMI LAN information.

```
ipmitool lan print
```



```
[root@localhost ~]# ipmitool lan print
Set in Progress         : Set Complete
Auth Type Support       : NONE MD2 MD5 PASSWORD OEM
Auth Type Enable        : Callback : MD2 MD5
                        : User     : MD2 MD5
                        : Operator : MD2 MD5
                        : Admin    : MD2 MD5
                        : OEM      :
IP Address Source       : Static Address
IP Address              : 10.14.188.233
Subnet Mask             : 255.255.240.0
MAC Address             : 78:45:c4:fb:62:07
SNMP Community String   : public
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00 TOS=0x00
BMC ARP Control         : ARP Responses Enabled, Gratuitous ARP Disabled
Gratituous ARP Intrvl   : 2.0 seconds
Default Gateway IP      : 10.14.176.1
Default Gateway MAC     : 00:00:00:00:00:00
Backup Gateway IP       : 0.0.0.0
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14
Cipher Suite Priv Max   : aaaaaaaaaaaaaaa
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
[root@localhost ~]#
```

Figure 25 – ipmitool cipher suite privileges

From Figure 25:

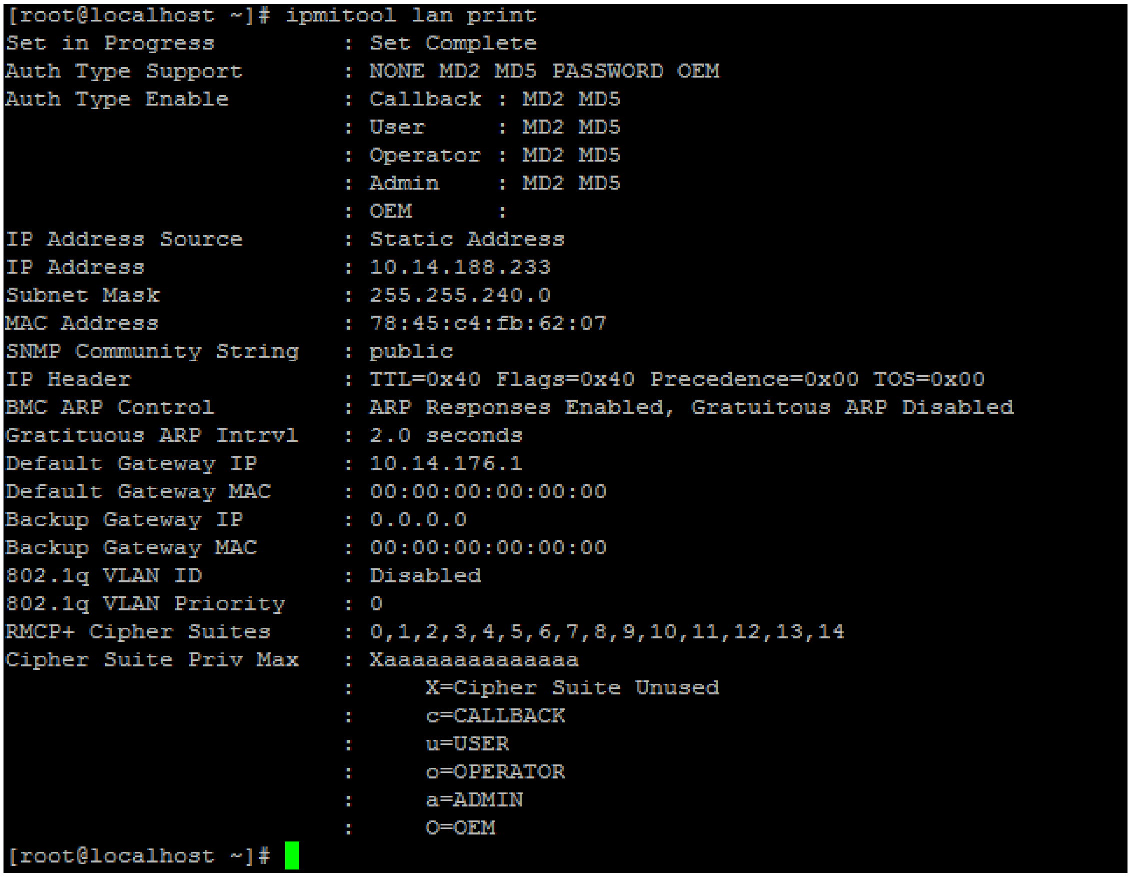
1. Shows the supported cipher suites
2. Shows the privilege level for each cipher suite
3. Defines privilege levels

To disable cipher suite 0, use the following command:

```
ipmitool lan set 1 cipher_privs Xaaaaaaaaaaaaaa
```

You must specify all 15 Cipher Suites

Once disabled, you will see cipher suite 0 designated with an X



```
[root@localhost ~]# ipmitool lan print
Set in Progress         : Set Complete
Auth Type Support       : NONE MD2 MD5 PASSWORD OEM
Auth Type Enable        : Callback : MD2 MD5
                        : User     : MD2 MD5
                        : Operator : MD2 MD5
                        : Admin    : MD2 MD5
                        : OEM      :
IP Address Source       : Static Address
IP Address              : 10.14.188.233
Subnet Mask             : 255.255.240.0
MAC Address             : 78:45:c4:fb:62:07
SNMP Community String   : public
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00 TOS=0x00
BMC ARP Control         : ARP Responses Enabled, Gratuitous ARP Disabled
Gratituous ARP Intrvl   : 2.0 seconds
Default Gateway IP      : 10.14.176.1
Default Gateway MAC     : 00:00:00:00:00:00
Backup Gateway IP       : 0.0.0.0
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14
Cipher Suite Priv Max   : Xaaaaaaaaaaaaaa
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
[root@localhost ~]#
```

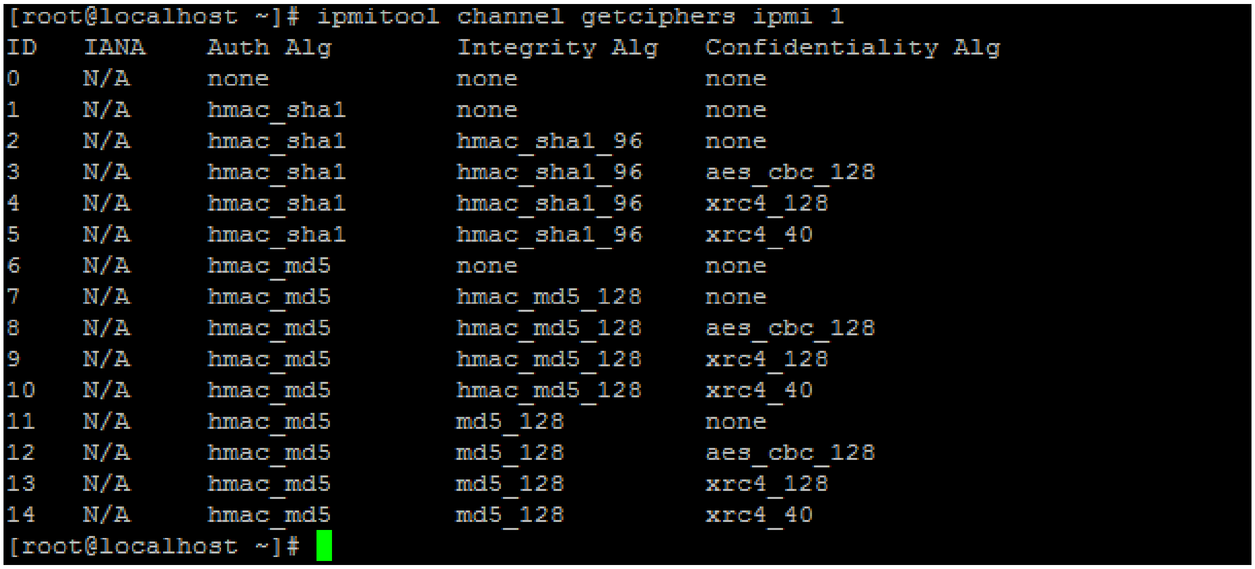
Figure 26 – ipmitool cipher suite privileges, cipher suite 0 disabled

IPMI v2.0 Weak Cipher Suites

While cipher suite 0 may provide no security by design, there are other cipher suites which are open to brute force decryption attacks following a man-in-the-middle style attack due to weak security.

The IPMI v2.0 specification download can provide details for each cipher suite, but we can also query that information from the BMC itself.
Using the following command we can display the supported cipher suites:

```
ipmitool channel getciphers ipmi 1
```



```
[root@localhost ~]# ipmitool channel getciphers ipmi 1
ID   IANA    Auth Alg        Integrity Alg   Confidentiality Alg
0    N/A     none            none            none
1    N/A     hmac_sha1       none            none
2    N/A     hmac_sha1       hmac_sha1_96    none
3    N/A     hmac_sha1       hmac_sha1_96    aes_cbc_128
4    N/A     hmac_sha1       hmac_sha1_96    xrc4_128
5    N/A     hmac_sha1       hmac_sha1_96    xrc4_40
6    N/A     hmac_md5        none            none
7    N/A     hmac_md5        hmac_md5_128    none
8    N/A     hmac_md5        hmac_md5_128    aes_cbc_128
9    N/A     hmac_md5        hmac_md5_128    xrc4_128
10   N/A     hmac_md5        hmac_md5_128    xrc4_40
11   N/A     hmac_md5        md5_128         none
12   N/A     hmac_md5        md5_128         aes_cbc_128
13   N/A     hmac_md5        md5_128         xrc4_128
14   N/A     hmac_md5        md5_128         xrc4_40
[root@localhost ~]#
```

Figure 27 – ipmitool example, show support cipher suite details

Using the same command above to disable cipher suite 0, we can customize which cipher suites can be used. From the example above, we would issue the following command to disable everything that does not offer 128bit encryption

```
ipmitool lan set 1 cipher_privs XXXXXXXXaaXXaaX
```

You should always check which cipher suites are in use and disable those that are not desired.

IPMI over LAN vs Web Services

From a security configuration standpoint, it is important to note that IPMI over LAN and web services configurations are independent.

For example, current iDRAC/CMC web interfaces default to enforcing 128-bit or higher, and there are plans to add options to increase the minimum to 168-bit – 256-bit in later firmware versions.

At the same time, IPMI over LAN supports various cipher suites, from no encryption to varying encryption strengths up to 128-bit.

Configuration changes to one will not apply to the other.

IPMI Troubleshooting

IPMI provides a lot of access and information use for troubleshooting, but the following information can be used to troubleshoot IPMI issues.

Here are some common failure conditions and the error messages seen using ipmitool:
Note that in some instances, the same or similar errors may be present in other failure conditions.

IPMI driver missing or not started

Error:
Linux - Could not open device at /dev/ipmi0 or /dev/ipmi/0 or /dev/ipmidev/0: No such file or directory
Windows - Cannot open an IPMI driver: imbdrv.sys or ipmidrv.sys

Fix:
Local IPMI commands require loaded driver, install and/or start OpenIPMI for Linux, or install Microsoft driver

Incorrect user or password

Error:
IPMI v1.5 - Activate Session command failed
IPMI v2.0 - Unable to establish IPMI v2 / RMCP+ Session

Fix:
Verify IPMI credentials

Incorrect IPMI version

Get Session Challenge command failed
Unable to establish LAN session

Fix:
Verify command syntax for correct IPMI version

Firewall or other network filtering blocking access

Error:
Unable to establish IPMI v2 / RMCP+ session
Device not present (No Response)

Fix:
Ensure UDP port 623 is not blocked

References

[1] Dell, "Intelligent Platform," [Online]. Available: http://www.dell.com/downloads/global/power/ps2q04-019.pdf.

[2] "Intelligent Platform Management Interface," Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Intelligent_Platform_Management_Interface.

[3] "Data Center Management: Windows users can use IPMI tools too!," Intel, [Online]. Available: https://communities.intel.com/community/itpeernetwork/datastack/blog/2013/10/10/windows-users-can-use-ipmi-tools-too.

[4] "What Is IPMI," Intel, [Online]. Available: http://www.intel.com/content/www/us/en/servers/ipmi/what-is-ipmi.html.

[5] "IPMI v2.0 Rev 1-1 Specification," Intel, [Online]. Available: http://www.intel.com/content/www/us/en/servers/ipmi/ipmi-second-gen-interface-spec-v2-rev1-1.html.