Illuminating the Security Issues with Lights-Out Server Management

Anthony J. Bonkoski J. Alex Halderman

University of Michigan

What is IPMI?

Need to manage a massive cluster of servers?

OS installs, monitoring, power-cycle, etc.

How?

Intel introduces Intelligent Platform Management Interface (IPMI) Specification:

Adds a second computer

Always on

Integrated directly into the system buses (e.g. I^2C)

OEM Names:

HP iLo

Dell iDrac

Oracle iLOM

Lenovo/IBM IMM

SuperMicro IPMI

ATEN IPMI

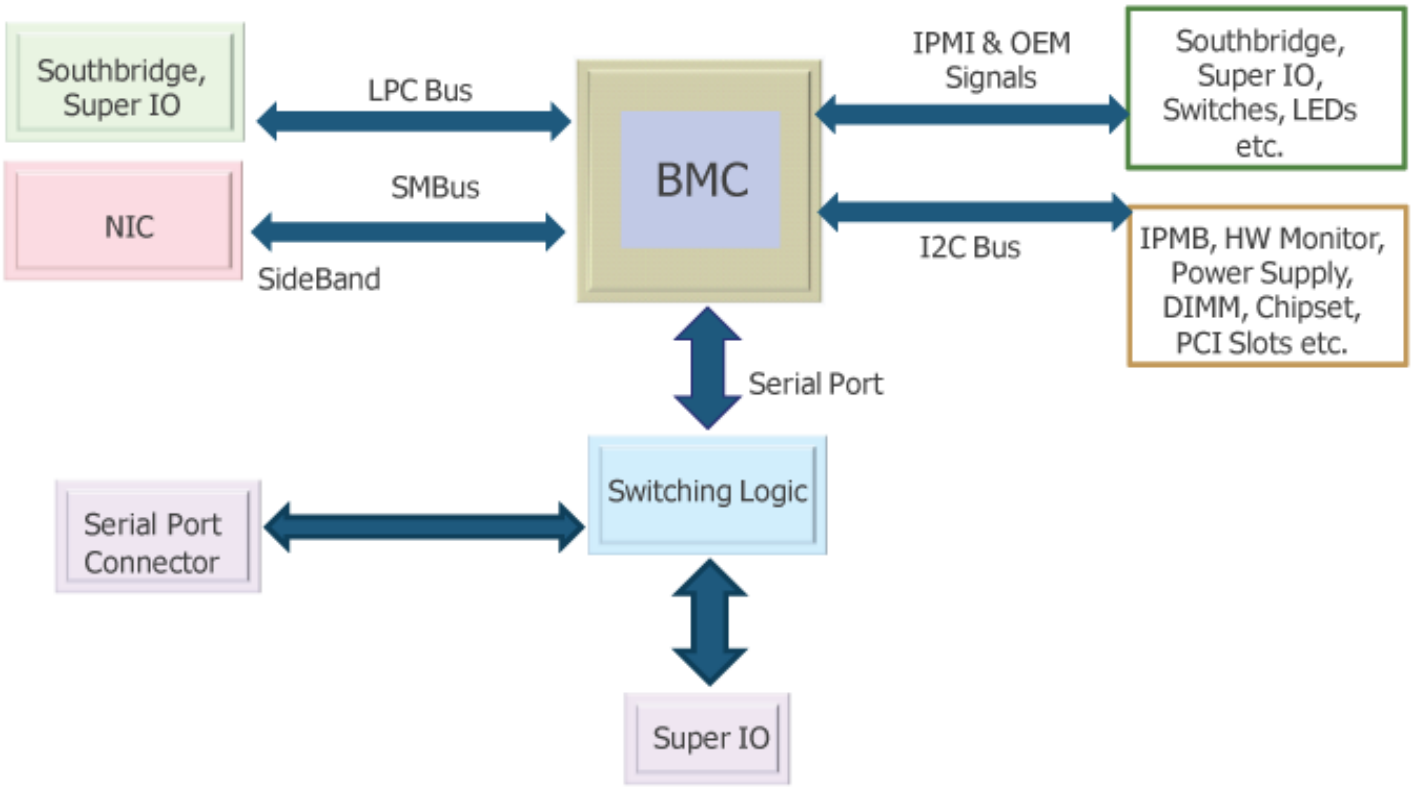
MegaRAC

Avocent IPMI

What is IPMI?

Baseboard Management Controller (BMC)

The embedded micro-controller: the second CPU



Typical IPMI Implementation

System

Embedded on Motherboard or Expansion card
CPU: ARM/MIPS or other low power embedded CPU
OS: Linux is common

Extra OEM Features

Remote Virtual Console
Remote Media
High network connectivity incl. HTTP and SSH.

Why do we care?

In short: IPMI is the perfect spying backdoor

- Always on and often pre-enabled.
- NIC failover*
- Powerful Remote Tools
- Widespread deployment: 100,000+ on public IPs

It's an embedded system…

- …often, security is an after-thought!

*As seen on our SuperMicro ATEN-based IPMI

Known Problems

Authentication Risks:

Many vendors ship default passwords

`root`/`calvin`[†]

Anonymous undocumented accounts[*]

Passwords stored in plain-text[*]

* SuperMicro ATEN-based IPMI

† Dell iDRAC

Recent Developments

Dan Farmer

- January 2013: Starts publicly denouncing IPMI
- Criticisms are largely just conjectures

- Finds some negligent flaws:
  - Hidden backdoor debugging web page on Dell iDRAC
  - Could gain root over ssh

Our Work

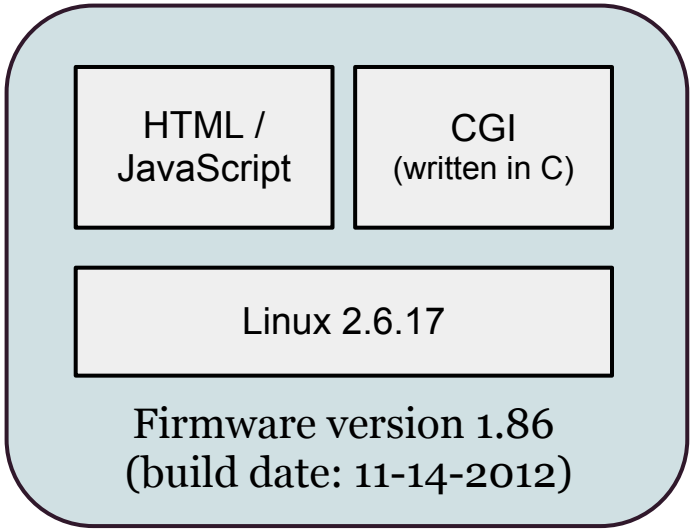
Is IPMI security actually a problem?

Supermicro IPMI

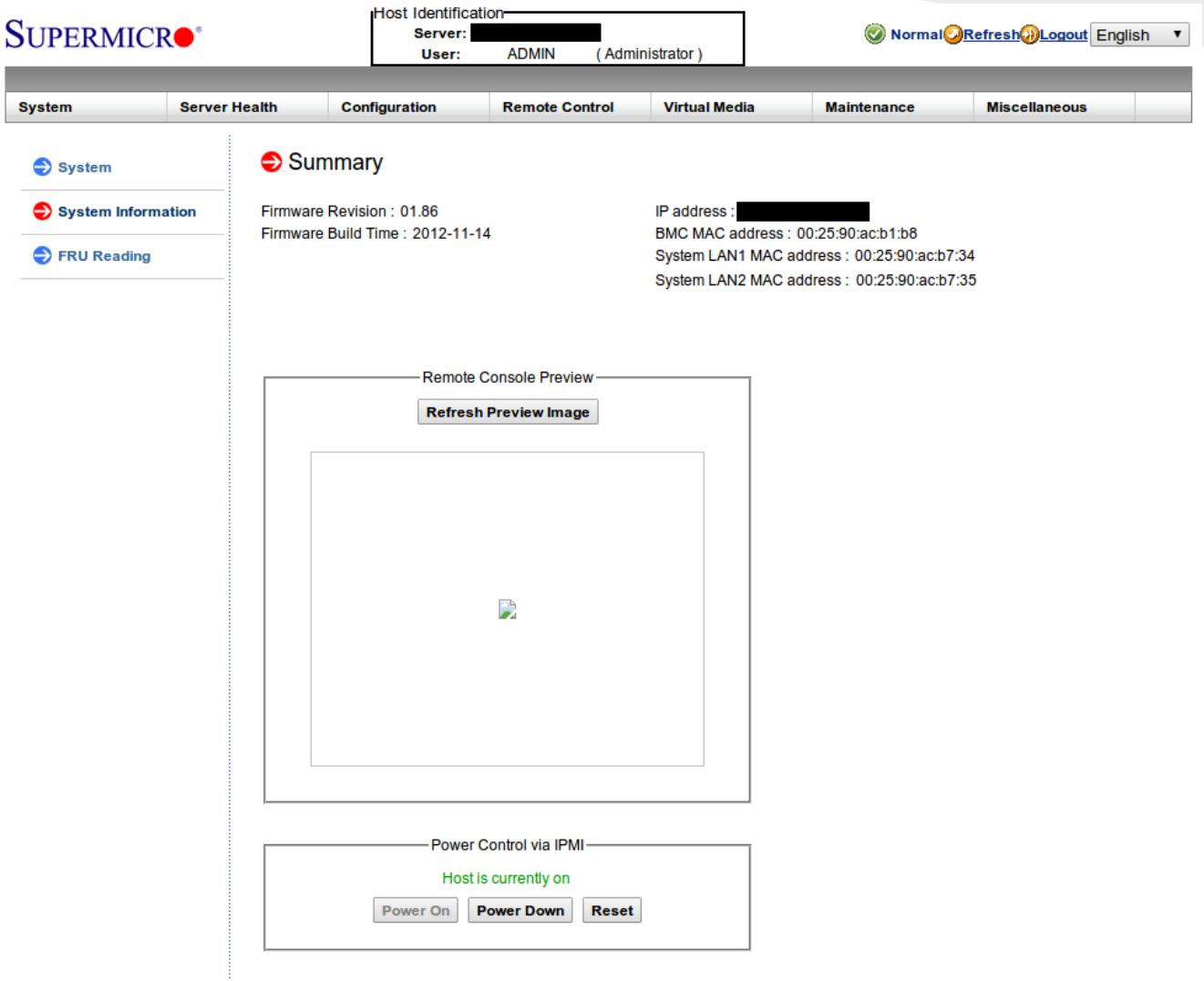
Supermicro SYS-5017C-LF

Nuvoton WPCM450
ARM-based BMC

IPMI Firmware by
ATEN Technology



Supermicro Web Interface



Supermicro SSH Interface

Backend: Highly modified fork of Dropbear

Frontend: Systems Management Architecture for Server Hardware Command-Line Protocol (SMASH)*

Notice: a system admin has no access to underlying Unix shell

```
ATEN SMASH-CLP System Management Shell, version 1.04
Copyright (c) 2008-2009 by ATEN International CO., Ltd.
All Rights Reserved


-> help
/

  The managed element is the root

  Verbs :
        cd
        show
        help
        version
        exit


->
```

*Distributed Management Task Force (DMTF) specification: dmtf.org/standards/smash

Reverse Engineering Approach

Fetch firmware from OEM website.

Scan and unpack: binwalk

```
DECIMAL       HEX             DESCRIPTION
--------------------------------------------------------------------------------
59700         0xE934          Copyright string:
60835         0xEDA3          Copyright string:
1572864       0x180000        CramFS filesystem,
9961472       0x980000        Zip archive data,
11086483      0xA92A93        End of Zip archive
12058624      0xB80000        CramFS filesystem,
```

Mount filesystems

Objdump and IDA Pro

What to Look For?

Begin with Classics:

1. Insecure Input Validation
2. Shell Injection
3. Buffer Overflows

Input Validation

All input validation is done in client-side javascript …

… and so is permission checking:

```
function PrivilegeCallBack(Privilege){
    // full access
    if(Privilege == '04'){
        isSuperUser = 1;
     }
     // only view
     else if(Privilege == '03') {
         var_save_btn.disabled = true;
     }
     // no access
     else {
         alert(lang.LANG_NOPRIVI);
     }
}
```

Server-side?

No permission checking.

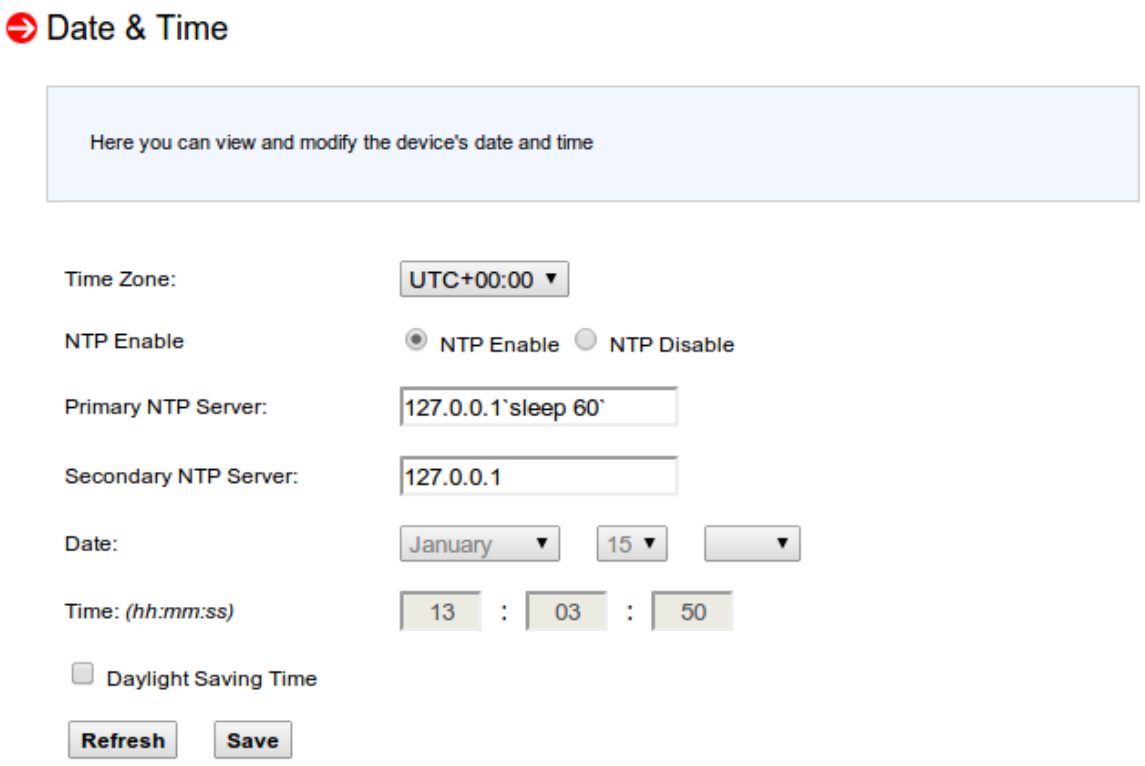
No escaping of input passed to shell.

No string length checking in CGI.

Shell Injection

15 of 67 CGI programs made calls to `system()`.

Confirmed shell injection in `config_date_time.cgi`:



Shell Injection

15 of 67 CGI programs made calls to `system()`.

Confirmed shell injection in `config_date_time.cgi`:

Getting command output

- Redirect to `/nv/system_log`.
- Issue GET request to `system_log.cgi`.

Create a psuedo-terminal

- Wraps GET ands POST request in a python script.

```
root@localhost #
```

Buffer Overflows

Server backend:

- … CGI programs.
- … written in C.
- … running as root.

Buffer Overflows

Server backend:

- ... CGI programs.
- ... written in C.
- ... running as root.

```
// login.cgi
int main(void)
{
  char name[128], pwd[24];
  char *temp ;
  // ... initialize ...
  temp = cgiGetVariable("name");
  strcpy(name, temp);
  temp = cgiGetVariable("pwd");
  strcpy(pwd, temp);
  // ... authenticate user ...
}
```

Buffer Overflows

Server backend:

- ... CGI programs.
- ... written in C.
- ... running as root.


```
// login.cgi
int main(void)
{
  char name[128], pwd[24];
  char *temp ;
  // ... initialize ...
  temp = cgiGetVariable("name");
  strcpy(name, temp);
  temp = cgiGetVariable("pwd");
  strcpy(pwd, temp);
  // ... authenticate user ...
}
```

Buffer Overflows

No length validation?

```
<input name="name" size="20" maxlength="64"
```

Buffer Overflows

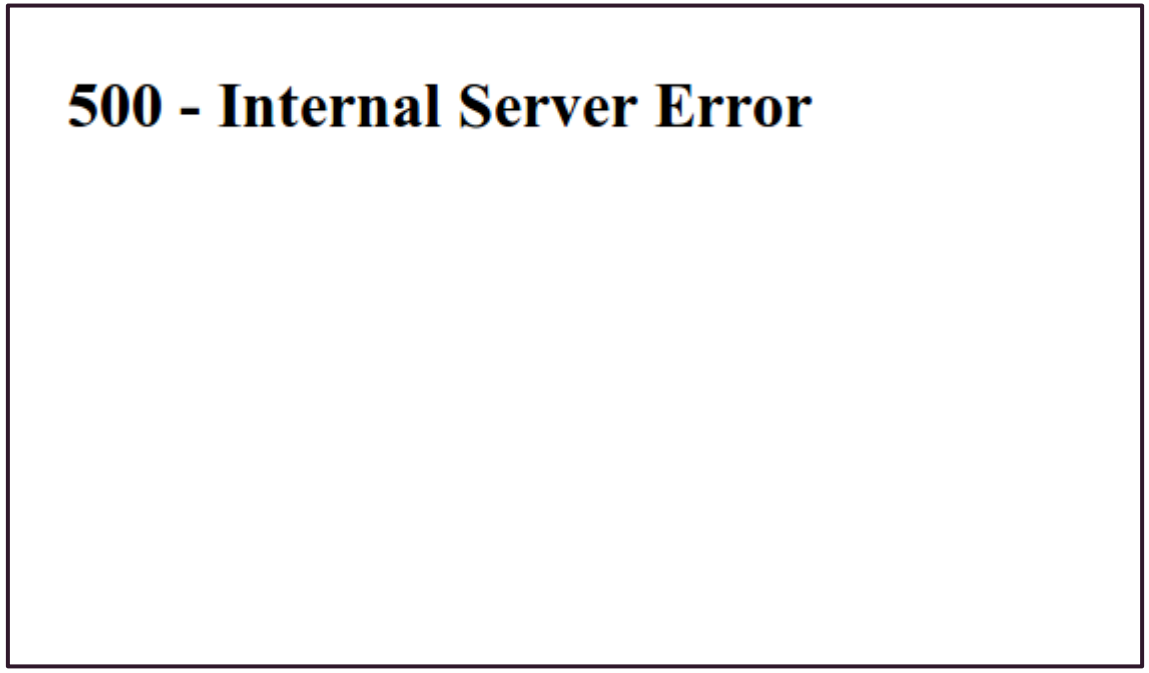
No length validation?

```
<input name="name" size="20" maxlength="1000"
```

Buffer Overflows

No length validation?

```
<input name="name" size="20" maxlength="1000"
```



Buffer Overflow Exploitability

Buffer-overflow defenses?

No DEP (Stack and Heap are executable).

No Stack Canaries.

Limited ASLR.

(Stack/Heap base addresses are randomized, but dynamic libraries are **not**. Return-to-libc works.)

Exploitation Challenges

Stack is randomized (ASLR).

...but, only 12 bits are random. Just 4096 possibilities.

We gain control on the return from main().

Stack is small: shellcode must be compact.

BMC crashes and reboots if pounded too hard with requests.

Buffer Overflow Exploit

Solutions

Store the shell command in the `name` buffer.

Brute force through the stack randomization.

Limit the time between brute-force iterations.

Avg. search time: ~7 min.

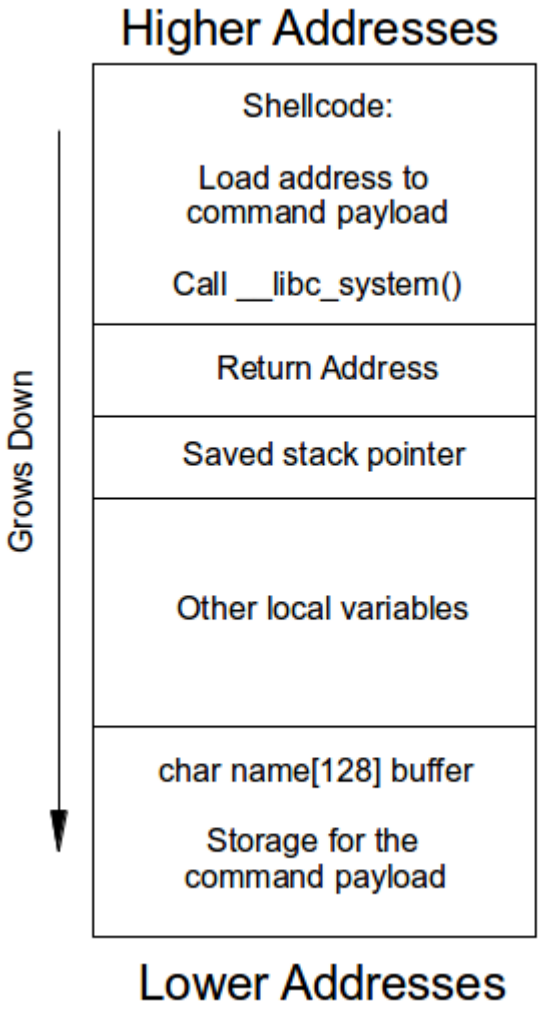
Payload

Fetch (wget) and install modified SSH daemon.

Forks root shell on *incorrect* password.

Only 2 instructions changed!

```
root@localhost #
```



Vulnerable Models?

Cursory check of all Supermicro IPMI firmware downloads as of May 23, 2013.

- 30 of 64 images appear vulnerable.
- 135 device models.

Supermicro says they're working on a fix.

Possibly affects other ATEN-based products.

The Impact

So, rooting this device is *easy*!
But, what are the implications?

Yet another broken embedded system?

The Impact

Only as *secure* as our weakest component.
Entire system is now vulnerable!
Adding an entire computer only weakens.

IPMI for Evil

BMC-based spyware and botnets

Rooted BMC → Rooted host system

Mount a custom OS and reboot.

Rooted host system → Rooted BMC

Re-flash the BMC with malicious code.

BMC rootkits

A backdoor that survives potentially forever.

A scary thought

IPMI meets Matrix → Is your IPMI just emulated? How do you know?

Network Measurements

Scanned all public IPs on May 7, 2013 using ZMap*.

Downloaded all X.509 certs from HTTPS servers.

Used identifying characteristics of default certificates.†

Platform	Devices on Public IPs
Supermicro IPMI	41,545
Dell iDARC	40,413
HP iLO	23,376
Total	105,334

Could root all these in parallel in minutes!

* *ZMap: Fast Internet-wide Scanning and its Security Applications*. Paper and tool coming this FRIDAY at Usenix Security.

† Details on "identifying characteristics" may be found in our paper

Defenses

For System Operators

Never attach your IPMI device directly to the Internet.

- Use an isolated management network or VLAN.

Change default passwords and certificates.

Disable IPMI if you don't need it.

Unfortunately: we're at the will of the Vendor

Defenses

For IPMI Vendors

These are textbook vulns. *You have to do better*.

Apply security engineering practices.

Sign and verify firmware when flashing.

Make devices hard to deploy on public IPs.

Lessons

A Culture Clash?

Embedded

Internet

IPMI: hopefully a climax

Future Work

Analysis of other vendors' implementations

Dell, HP, Lenovo, Oracle, etc.

Firmware update exploitation

Can an attacker inject a backdoor that persists?
Across BMC reboot? Across BMC flashes? Forever?

IPMI honeypot

Unclear whether attackers are exploiting these devices in the wild.
Some anecdotal evidence of their use as spambots.
Are they being used for other malicious purposes?

Conclusions

IPMI serves a vital role for system management.

Carries elevated risks, potential for powerful attacks.

At least some vendors are getting it badly wrong.

Farmer is correct: IPMI *is* a serious concern.

Our work: A call to arms .

Illuminating the Security Issues with Lights-Out Server Management

Anthony J. Bonkoski
abonkosk@umich.edu

J. Alex Halderman
jhalderm@umich.edu

University of Michigan

Zmap Scan Details

Vendor	Identifying Characteristics
SuperMicro	Subjects containing “linda.wu@supermicro.com” or “doris@aten.com.tw”
Dell	Subject containing iDRAC
HP	Subjects containing “CN=ILO” and issuers containing “iLO3 Default Issuer” or “Hewlett Packard”

*Landing pages spot-checked for false positives