Making your own console server using OpenBSD

Using only base… and serial hardware

Me

Systems Administrator at RIT Software Engineering

Using UNIX since 1990

First BSDs were 4.3 and 4.3 Tahoe also in 1990

Started using OpenBSD about 2.9

Why use serial?

Dead simple

Can continue working when most of the OS can't

Even a crippled kernel can poke characters into a UART

Why a console server?

Why do this when one has…

- IPMI?
- iLOM?
- DRAC?

Why not IPMI/iLOM/DRAC?

- Out of date firmware
  - Obsolete SSH
  - Obsolete crypto
  - Java Web Start application signed with MD5
- Security?
  - Expose that to the internet?
  - Extra hops for protection

Why not simplify while being secure?

Genesis

In the beginning, there was misc@

The host

A spare desktop

Cut it down, make it less likely to fail

- Replace HD with CF
- Remove CD-ROM
- Remove Floppy

A desktop? Really?

Had it lying around

4 PCI slots for holding serial cards

Fast boot time

Most of all: Cheap

Saves money for ...

The Serial Hardware

4 8-port Multi-port PCI Serial Cards

Oxford chipset based using puc(4)

Breakout cables

DD-78 (!!) to 8 DE-9

Cthulhu Cable

Known to cause madness when attempting comprehension

(Cable is on the right)

Cabling

Started with standard RS232 cables

All same length, made the mess worse

Switched to RJ45 to DE9M adapters

Ethernet comes in *lots* of lengths

Null modem adapters on breakout cable end

Driver issue

My cards were giving me garbage

Some cards use different speed crystals

I needed to specify my specific variant and use a multiplier

Led to my first patch in OpenBSD

Mapping ports

Can you trace them? Probably.

-10 Sanity

I mapped them on the OS side

Tip in - What responds?

Rudimentary serial console

- Log in
- Use tip(1)
- Hit enter
- Start working

No history

Along comes tmux

Why didn't I use screen? Not very scriptable

Tmux windows - Good until want to access two consoles at once

On to tmux sessions - Every session is independent

A simple script starts all sessions

Now we get history (until console server reboots)

Lots of features - no packages - less to maintain

Simple script

```
#
# Place all the hosts you want to have a session in the hosts variable
#
# All your names should exist in your /etc/remote
#
hosts="cisco cisco2 zin dynomutt norville mentok yakky \
        mightor reducto pintsize filedump eisenhower \
        openbsd devlin"

for host in $hosts; do
        /usr/bin/tmux new-session -d -s $host "cu $host"
done
```

I plan a better script with accompanying rc.d scripts

Multiple user tenancy

Student organization had server in my rack

Sometimes they want physical access - Babysitting

They don't *need* physical access, that's what this project is for

Tmux sessions under a user for student organization

Can be expanded to more users

Problems others have had

Interrupt storms

Nick Holland has talked of interrupt storms with PCI cards

I've never seen this

Are my cards that awesome? ¯_(ツ)_/¯

Features to implement

- Session history persistence
  - A reboot of console server wipes scrollback
- Logging
  - How about keeping this stuff in logs?

I have plans to do this, but haven't worked out how all the pieces fit

If I could do it all again differently

~~If~~ When I ~~could~~ do it all again differently

Better hardware now:

- PCIe (PCI is dead)
- Breakout cables can have smaller connectors with card
- USB?
- Breakout panels
- Go right to RJ45, no DE9M
  - Look out! Some have “RJ45”, but are 10P10C
- No CF, use SSD

Conclusion

Not that much effort to get a better, more secure solution

Questions

As a teacher I had in high school said:

"Questions? Comments? Cheap shots?"

Appendixes

- Pinouts
- Manufacturers of serial hardware
- Lists of interesting serial hardware
  - NOTE: I have not tested any of the listed hardware yet, they just look promising

Pinouts - RJ45 to RS232 (RS232D)

RJ45	DE9	
1	6	DSR
2	1	CD
3	4	DTR
4	5	GND
5	2	TXD
6	3	RXD
7	8	RTS
8	7	CTS

Pinouts - RJ45 to Cisco

RJ45	DE9	
1	8	RTS
2	6	DSR
3	2	TXD
4	5	GND
5	5	GND
6	3	RXD
7	4	DTR
8	7	CTS

Manufacturers of serial hardware

- SIIG
- SYBA
- Startech
- Comtrol (Rocketport)
- Moxa

Comtrol (Rocketport)

- RocketPort 32-Port RJ45 Rack Mount Interface
  - Part #: 30060-1
  - Needs controller card
- RocketPort 16-Port RJ45 Rack Mount Interface
  - Part #: 30055-7
  - Needs controller card
- RocketPort EXPRESS 32-Port (Controller card)
  - Part #: 30138-7
  - Needs breakout board (rackmount or other)
- RocketPort EXPRESS 16-Port (Controller card)
  - Part #: 30137-0
  - Needs breakout board (rackmount or other)
-
- RocketPort 16-Port DB9M Surge Interface
  - Part #: 30035-9
  - Needs controller card
  - Has surge protection
- RocketPort 16-Port DB9M Interface
  - Part #: 30030-4
  - Needs controller card
- Rocketport EXPRESS Octacable RJ45
  - Part #: 30130-1
  - RJ45 8-port Cthulu cable
- Rocketport EXPRESS Octacable DB9M
  - Part #: 30128-8
  - DB9M 8-port Cthulu cable
- Rocketport USB Serial Hub III 8-Port
  - Part #: 98296-8
- Rocketport USB Serial Hub II 4-Port
  - Part #: 98295-1

Digi

- Digi Edgeport/8 DB-9
  - USB -> 8 DB-9
  - Part #: 301-1002-08
  - $455 USD from Digi-Key
- Digi Edgeport/416 DB-9
  - USB -> 16 DB-9
  - 4x Downstream USB ports
  - Part #: 301-2000-10
  - $759 USD from Digi-Key
- Digi Neo PCI Express 8-port (w/o cable)
  - Low profile bracket
  - Part #: 77000889
- Digi Neo 8-port DB9M Cthulhu cable
  - Part #: 76000529
  - $76 USD Digi-Key
- Digi Neo 8-port DB9M Breakout box
  - Part #: 76000561
  - $99 USD Digi-key
- Digi Neo 8-port RJ45 Breakout Box
  - Part #: 76000527
  - $99 USD Digi-key

Syba

- Syba SI-PEX15041 8-Port PCI-E
- Syba SY-PEX15019 8-Port PCIe
  - $34.51 USD on Amazon(!)

Moxa

- Moxa CP-118EL-A PCIe 8-port card