

Offset	Topic
00:17	<ul style="list-style-type: none"> • Intro <ul style="list-style-type: none"> • Con season <ul style="list-style-type: none"> • Penguicon 6.0 4-18 through 4-20 • http://penguicon.org • RavenCon 2008 <ul style="list-style-type: none"> • 4-25 through 4-27 • http://ravencon.com
04:05	<ul style="list-style-type: none"> • Word of the Week: daemon <ul style="list-style-type: none"> • http://catb.org/jargon/html/D/daemon.html
05:53	<ul style="list-style-type: none"> • Inner Chapter: Stretching Yourself <ul style="list-style-type: none"> • Key aspect of hacking is learning new things <ul style="list-style-type: none"> • As a professional, at least early on, need a certain amount of skills • As an experience hacker, keeps skills sharp • Sheds new light on the skills you already have • Keeps the mind flexible • Just perception of challenge can stall trying <ul style="list-style-type: none"> • Reality is rarely so, most programming is more similar than not • Find the courage to try, don't worry about failing • Compiling Linux kernel <ul style="list-style-type: none"> • My first time was a bit nerve wracking • No substitute for doing, learning firsthand it is very simple • Advanced math of crypto <ul style="list-style-type: none"> • Composed of simpler steps • Using trusted skills, functional decomposition helps ease learning • Code is forgiving, can always try again • Back up critical data, code • Take care not to try on a time critical project • Don't give in to temptation to coast <ul style="list-style-type: none"> • Balance can be difficult, don't want to burn out trying too much • But need to recognize when you've been standing still long enough • This applies equally to many different endeavors, especially the creative ones • Never know what you can or cannot accomplish unless you try • Finding, making time to learn <ul style="list-style-type: none"> • Personal projects, can always try to find opportunity • May have to invent a project if a new skill doesn't exactly fit <ul style="list-style-type: none"> • Try to make such projects as realistic as possible • Otherwise, like training muscles asymmetrically

Offset

Topic

- Don't skimp, build a full UI, use a full configuration, read from a real file, database
- On large professional projects, may be more opportunities in the small
- Again, when learning in a professional environment be sensitive to time
- If there is a tried and true approach, technology, set a drop dead date
- Give enough time to execute plan B
- Of course, some skills have no substitute, unavoidable risk
- Projects may have to bend to learning
- Not too different from building completely new, untried feature
- Research, read, and do
 - Original sources, like specifications can be a bit dense, inscrutable
 - Write ups from other hackers
 - It is rare you will be the first tackling a new skill
 - Best notes include sources demonstrating key aspects
 - Check your sources and don't use just one
 - Can run a risk of following a bad example if you don't consider more than one
 - In the worst case, someone else wrote up something they didn't really understand
 - Thrashed until they got something to work
 - May still post sources but this can obscure a lack of understanding
 - Open source provides lots of opportunities for reading
 - If you find some code that does what to you need, increases odds it will help
 - If you can interactively debug the code, that is even more helpful for understanding
 - If the research you've found seems thin, don't be shy, write, add your own
 - At a certain point, need to stop reading and try
 - When you're going a little cross eyed, that's the time to get your hands on code
- Take notes
 - Check your own assumptions as you go
 - Help build on what you understand along the way
 - Can make questions you need to ask, answer obvious
 - Good reference after the fact as you are practicing to achieve mastery
 - If you use the skill intermittently, helps to refresh as needed
- Careful about over using new tools, skills
 - Similar to golden hammer
 - Newness of skill may encourage overuse

Offset

Topic

- Try to err on the side of being conservative
- No matter how much you like the new skill, old one is still easier to quantify
- For professional work, this is critical to keep cost of maintenance, risk down
- Try to evaluate what the new skill, tech does better than existing
- This needs to outweigh the risk of using something new, its own inherent risks
- Seen this over and over with XML, in particular
 - XML is human and machine readable
 - Very verbose, though, not useful in memory, space constraint applications
 - Is further removed from compiled, behavioral code, may add complexity
 - In particular, some of the deployment config uses I have seen violate locality of reference
 - Code annotations, by comparison, local to code
 - Same language, same domain
 - No absolutely correct answer, just need to do best to weigh trade offs
- Key to building an attractive resume
 - The skills themselves have value
 - Being able to explain in the resume the speed, thoroughness of learning even better
 - Technology rarely stays still
 - Not necessarily realistic to expect the skills your start with will always be enough
 - Some of this is artificial
 - Shiny new thing syndrome
 - New must be better, sometimes vendors capitalize on this
 - Proof is in the pudding, try to use as objective arguments as possible to dissuade clueless managers
 - There have indeed been genuine innovations that beg learning
 - Hard to tell at the time
 - Trust your instincts and be skeptical
 - If you can get by just as well with what you have, then isn't a truly new thing, need
- Learning can be its own reward
 - Going from zero knowledge to a working implementation
 - Immensely satisfying
 - A very common characteristic of hackers
 - Fun to share your knowledge, helps cement understanding
 - Enjoyable to be an expert, be the answer guy
 - Don't be shy to share, though

<u>Offset</u>	<u>Topic</u>
32:34	<ul style="list-style-type: none">• Those who can teach are more valuable than those that can just learn• Outro<ul style="list-style-type: none">• Contact me<ul style="list-style-type: none">• Email to feedback@thecommandline.net• Web site at http://thecommandline.net/• IM to command.line@skype• Listener comment line is 240-949-2638• del.icio.us tag is "for:cmdln"• http://twitter.com/cmdln• I'd like to thank libsyn.com for AAC hosting and Wouter de Bie for MP3 hosting• These notes and the show audio and music are covered by a Creative Commons license<ul style="list-style-type: none">• http://creativecommons.org/licenses/by-nc-sa/3.0/us/• Attribution, non-commercial, share alike