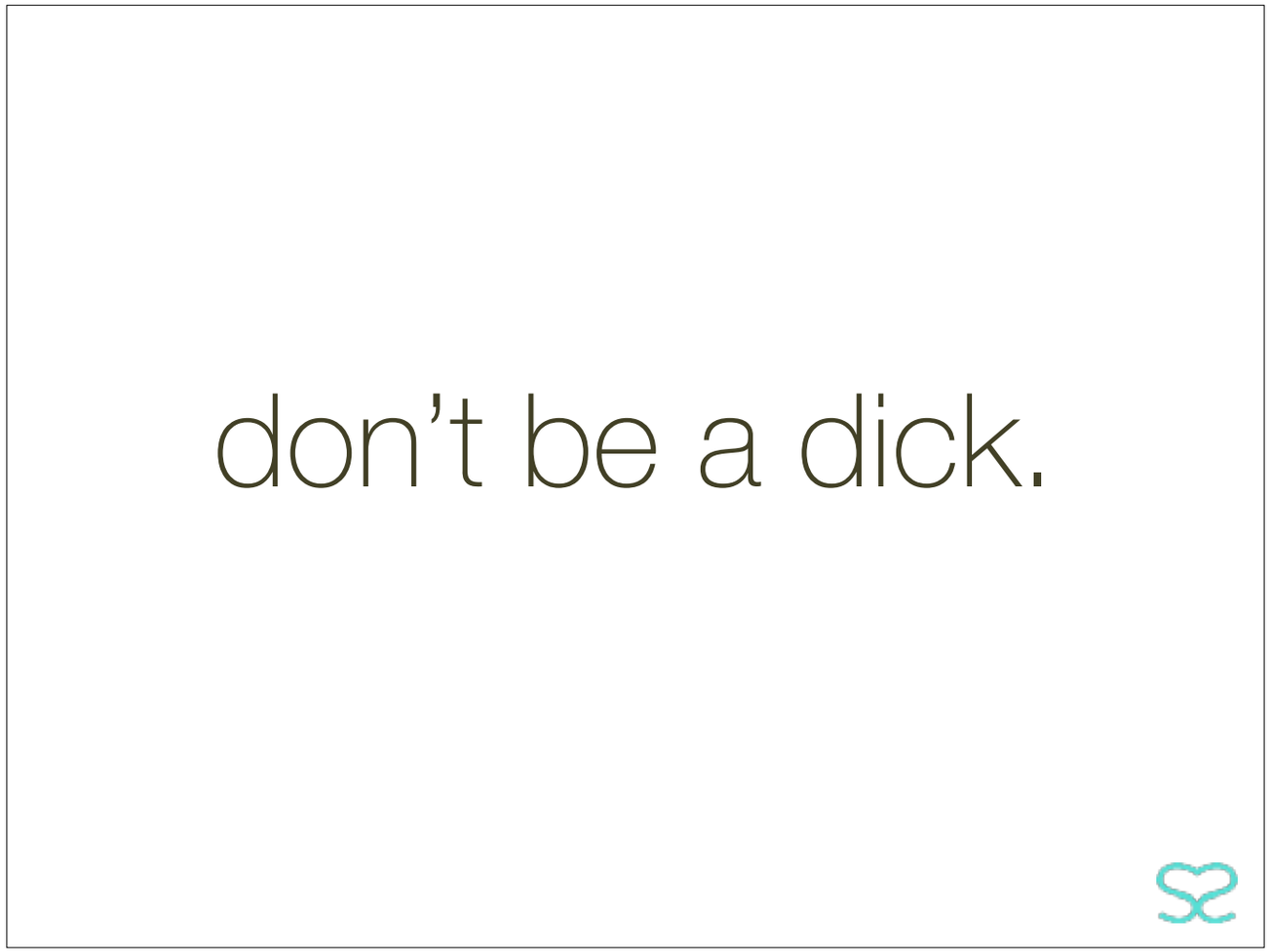
alright, so, i’m whitney champion.
i think it’s safe to say that a lot of us are here because we have an affinity for the security field, in some form or another
so this will be… kind of my spin on it

before we get started, i want to establish some ground rules
this is my first talk at a con
i don’t want to throw up in public

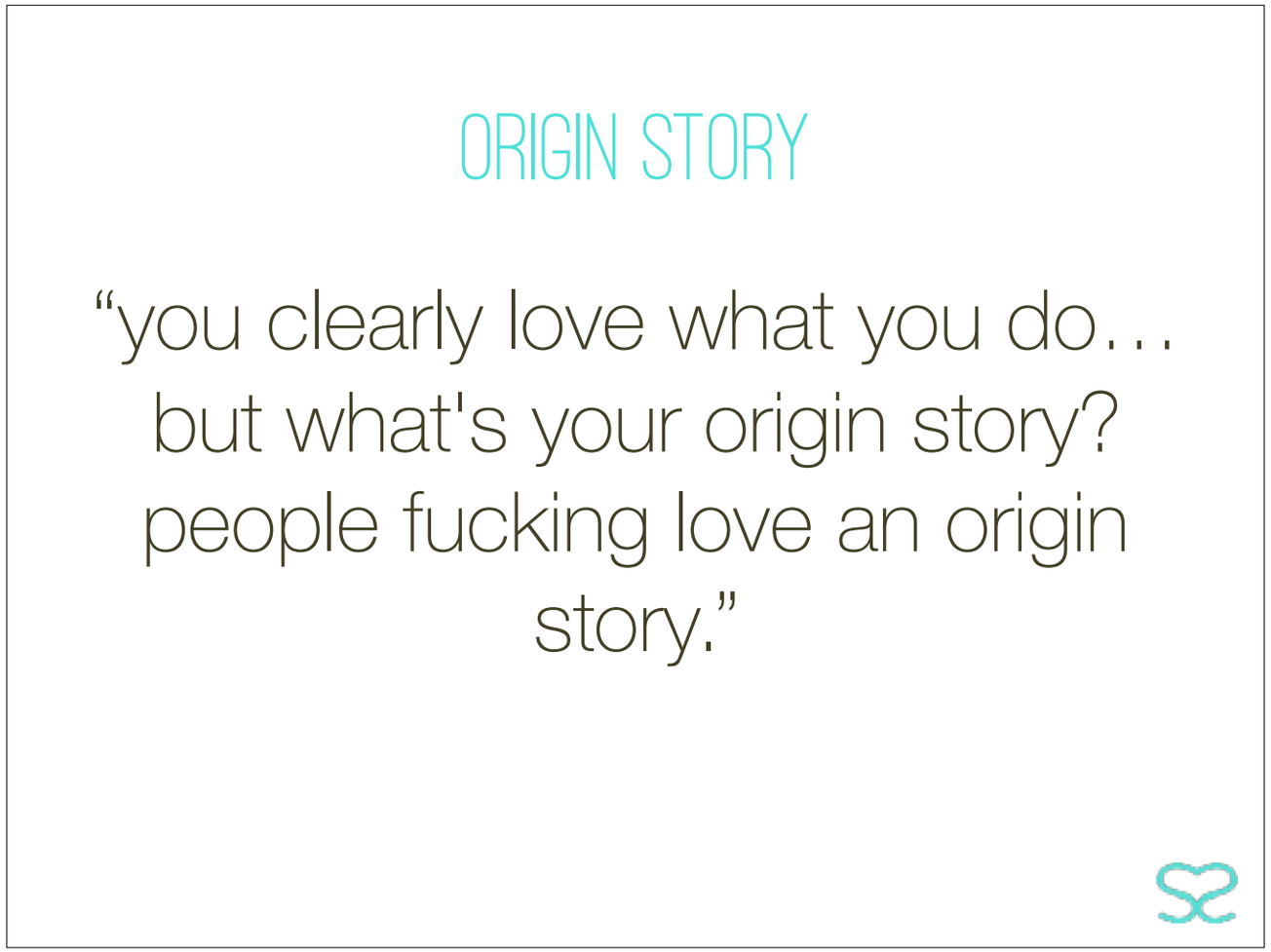


don’t be a dick

that’s all i got

drinks will be passed around

you can take a shot every time i drop the f bomb if you want, it’ll be fun until we run out of whiskey

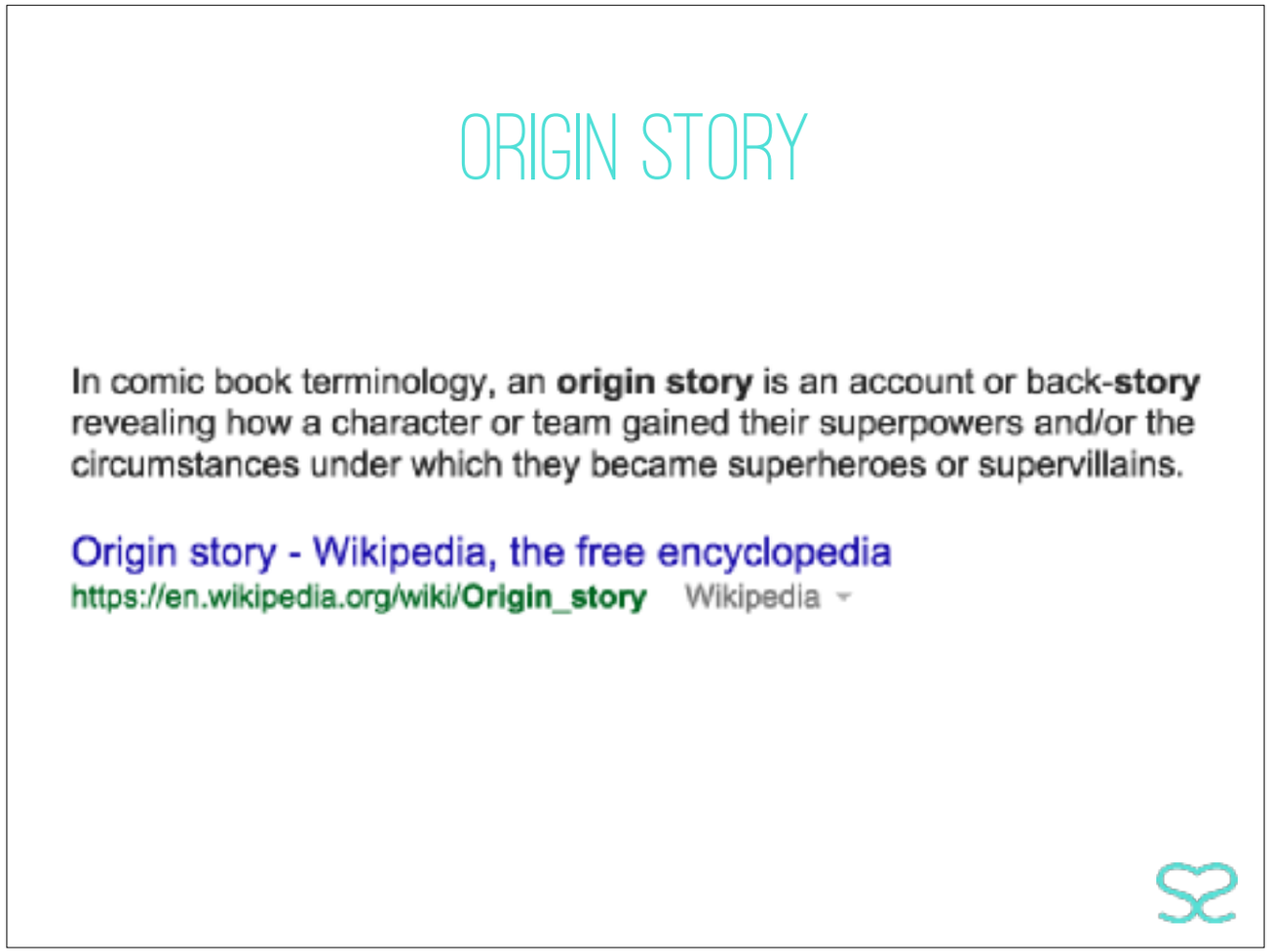


so, i started planning this thing back in… november
and was so nervous about submitting, because it involves talking in front of people, and basically that’s something i just don’t do
like… i have a minor panic attack in my daily standups almost every morning

so this kind of talk is terrifying, to me. i had friends review my slides, multiple times

and one of them told me… “you clearly love what you do... but what's your origin story?” … “people fucking love an origin story.”

and that sounds easy, right?

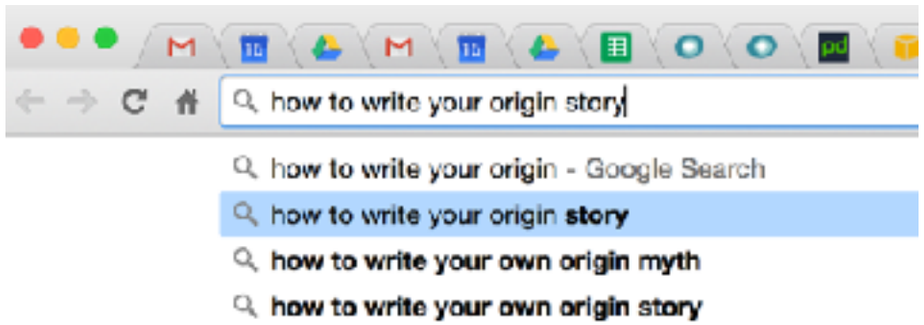


WRONG

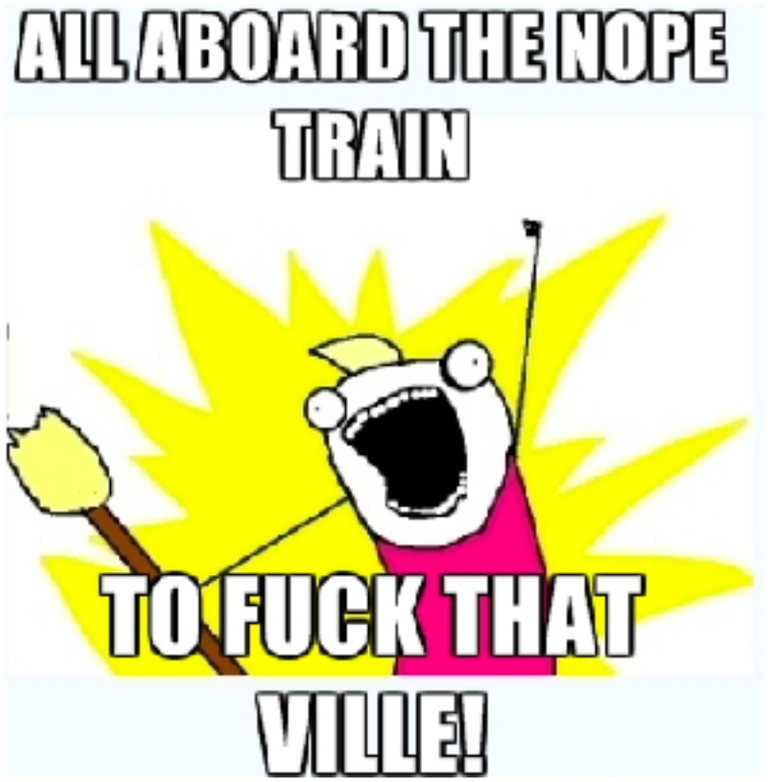
so i googled it… because i have no idea what my origin story is and that's what i do when i am fucking clueless

and that didn't help, because iii am not a superhero

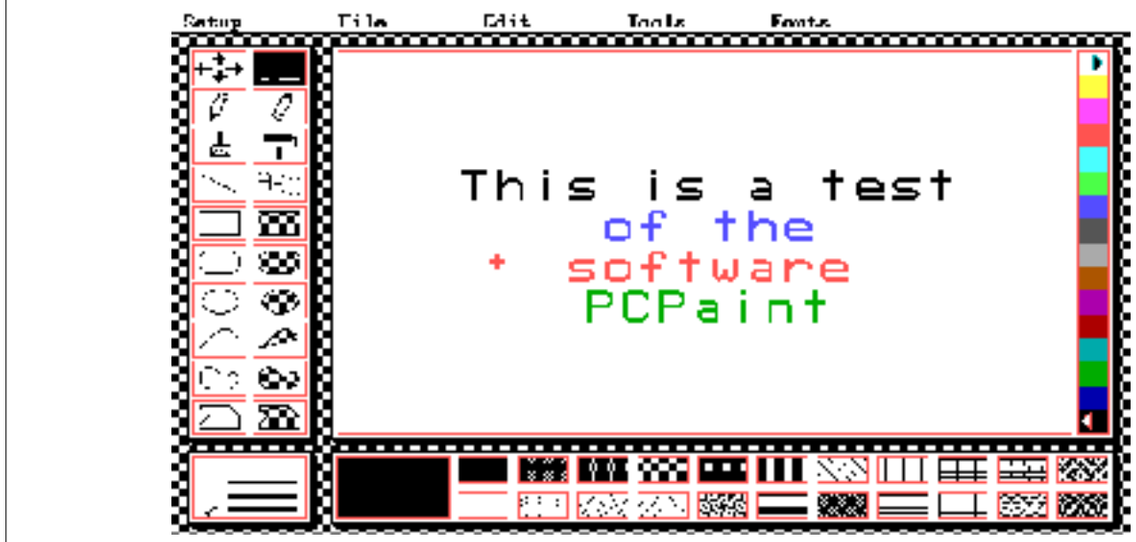
ORIGIN STORY



so then i googled that, because i still have no idea how to write an origin story



i wrote like 12 different paragraphs, they were all stupid
turns out, i suck at this



i ended up with this. this is my origin story, all the way back in 1990

TL;DR - I NEED OUTDOOR HOBBIES

- **1990** - dad brought home a PC
- 1995 - dad brought home a laptop, i decided that whatever i do, it's going to be on one of these doohickies
- 1996 - powerpoint and wordart freaking NINJA
- 1998 - hello, AIM! learned to type like a BOSS, also, 12 year olds shouldn't be allowed in chatrooms
- 2000 - expage.com, anyone? started learning how to build websites
- 2001 - found nerd friends, commence the LAN parties, became high school webmaster, dad bought me short-stack.com, started learning PHP
- 2002 - got job as flash developer, AND worked at pizza hut. ballin! pascal classes, sweet 16 LAN party
- 2003 - built my first PC, started taking cisco classes, C++ classes, learned how to host my own web server
- 2004 - java classes, graduated high school, started working at appstate tech support, installed slackware
- 2005 - introduced to fedora and red hat, first time using wifi on a laptop, NETBOOK holy shit
- 2006 - got involved in gaming club and appstate LUG, more nerd friends, so many LAN parties
- 2007 - got involved in appstate AITP, "wasted" entire spring break installing nvidia and broadcom drivers
- 2008 - started freelancing, taking security classes, went to campus job fair, got myself a job lined up
- 2009 - graduated, started working as gov contractor for honeywell, lived in a lab with nothing but bare metal, went to first CES and DEF CON
- 2010 - got CISSP/CEH/security+/linux+, left honeywell, started working for SPAWAR
- 2011 - learned android, got RHCE, left SPAWAR, went to SPARC, started using AWS and "the cloud"
- 2012 - started making the android app for DEF CON
- 2013 - joined the SPARC mobile team
- 2014 - red hat enterprise virtualization certified
- **2015** - openstack certified, DEF CON makes app official, cofounded DryStax

and then all this happened
i am not actually going to read this
nor do i expect you to be able to because that is ridiculous
TL;DR: i need more outdoor hobbies

so, a little bit about me, these are things i like

whiskey
weapons
crossfit
hermit crabs
jersey boys
unicorn paraphernalia
sinatra
seinfeld
wearing sweatpants

THINGS I DON'T LIKE

Jon Welborn
to Matt, Vic, David, Curbob, Nick, Iock

We just need to make sure we have canned air and a volunteer on hand for her talk.

evan booth <evanbooth@gmail.com>
to Jon, Matt, Vic, David, Curbob, Nick, Iock

This.

this is jon and evan talking shit about me in an email chain after i submitted this talk
SERIOUSLY
neither of you assholes were even there when that happened

i went to bsides charleston a few months ago, and when getting ready to give his talk, a friend of mine said you should always do this, or no one is going to give a damn what you say. establish credibility

and he’s right. because, let’s be honest, most of you probably wouldn’t listen to me otherwise.

i might look like i’m 21 and sound like a bimbo. but i’m not, i promise. and i’ve done a few things.

WHO AM I?

linux / devops / infosec / android at @SPARCedge

back in 2009, i was a government contractor at honeywell
then i went to SPAWAR, became a government employee, learned that this is where happiness goes to DIE
i was on some badass projects, miserable everything else
also, fun fact, if you hop on IRC from a SPAWAR address, and go in a defcon or 2600 or infosec channel, people freak the FUCK out and it's kind of awesome
anyway, i wanted out of there, i found SPARC
i was an engineer on the VBMS prod ops team for over 2 years, supporting a few hundred servers that host the veterans benefits management system
now i do the same on my team now for our USPS applications
was the lead android dev on a project called stream for about a year, kind of like periscope and meerkat
i also support several other commercial projects at SPARC

WHO AM I?

co-founder / infrastructure at @DryStax

i’m a co-founder of a company called drystax, we formed sometime in the middle of last year
it’s a SaaS platform for marinas that manages their operations and boat launches
we scored our first international client in australia back in november, it’s been fun so far

WHO AM I?

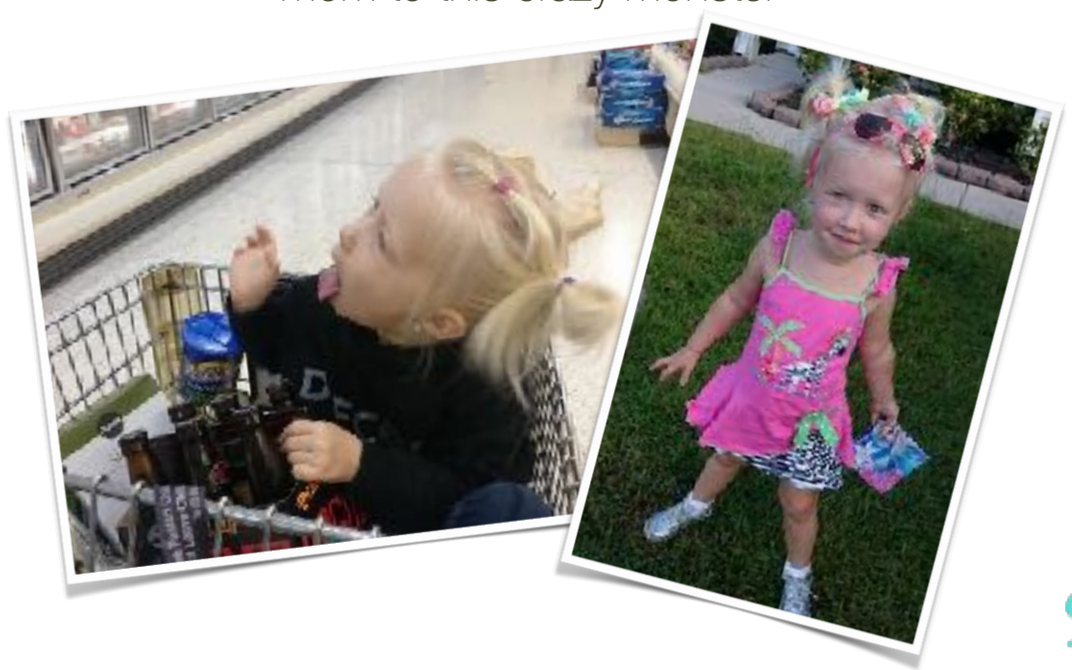
consulting / development

SHORTSTACK, LLC

i’ve had my own freelance business for about 8 years now
it comes and goes, depending on how much i want to work in the evenings
90% of which is done while drinking and watching friends

WHO AM I?

mom to this crazy monster



aaand last, but not least, i’m a mom
she’s 3 and she’s terrifying and i love it
one day she came home and had painted her own hair blue, and when i asked her what she had for lunch, she said “babies”. so, you know. parenthood is fun.

onward!

dad always told me...

"whitney, always have a resume."

"whitney, always have something to fall back on."

"whitney, always have a good knife."

when i was growing up, my dad told me 3 things
always have a resume…
always have something to fall back on…
always have a good knife…
dad is a smart guy, and a badass
he is the engineer i have ALWAYS aspired to be
he's unbelievably smart, comfy job, awesome toys
also he kind of looks like christopher walken

i listened...

so… naturally, i listened
since i was 15, i've made sure i
always had an up to date resume and portfolio
and had multiple career paths

i’ve tried to do all the things
i don’t sleep enough
definition of burning at both ends

but what it boils down to is this
whether it’s at home… or at work…

@PLAY

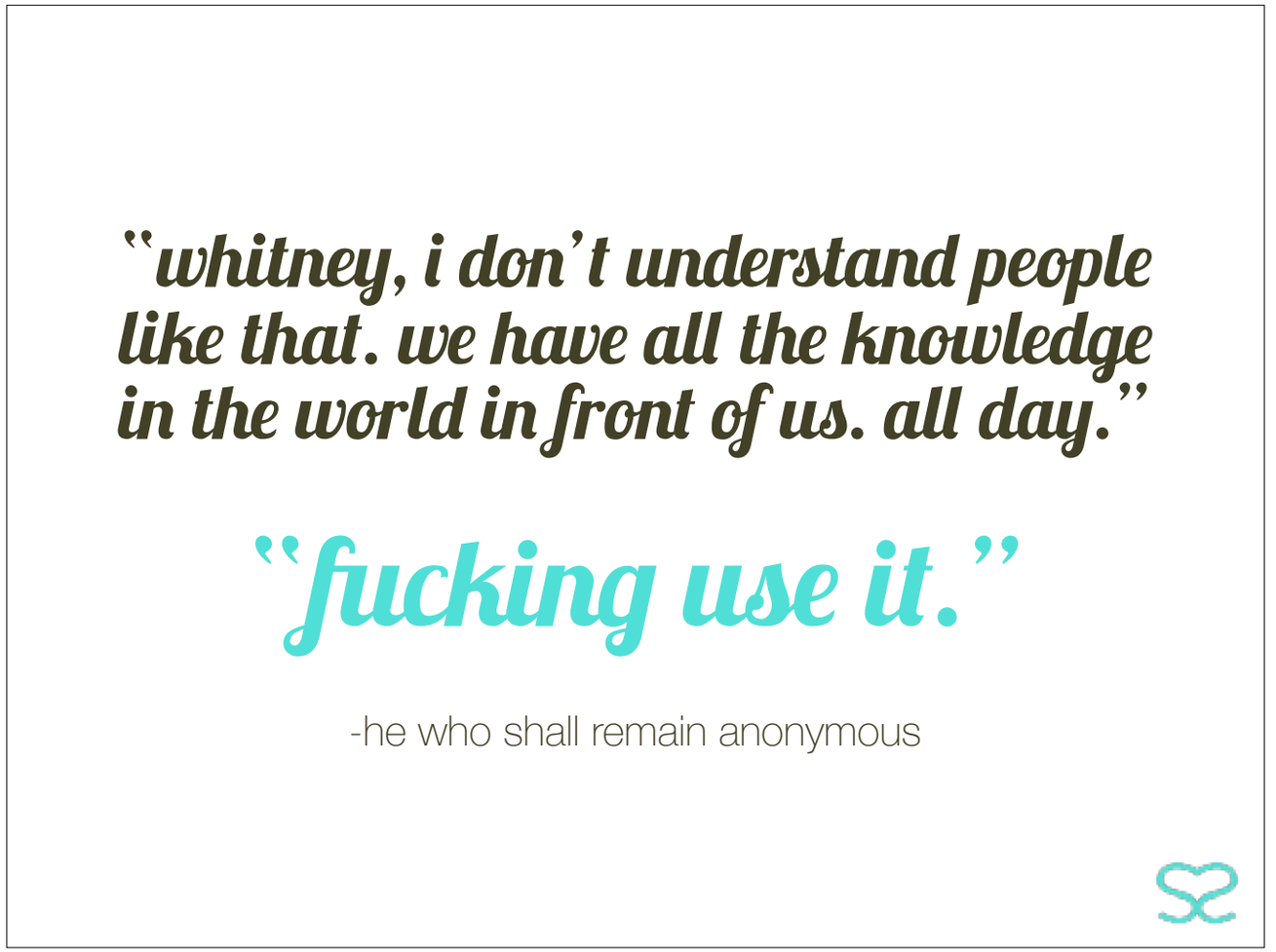
hackathons
LUG groups
LAN parties
building computers
welding
graphic design
web development
android
podcasts

@WORK

linux
infosec
android
certifications
competitions
training
meetups
conferences

college didn’t teach me shit in regards to anything technical in my career
all of this did
branch out, get involved

i hear friends and coworkers say this all the time
and don’t get me wrong, i used to be guilty of it, too
until a good friend of mine gave me this lightbulb moment



he was on a rant about another coworker saying how bored he was
his response was… "we have all the knowledge in the world in front of us. all day. fucking use it"
…boom

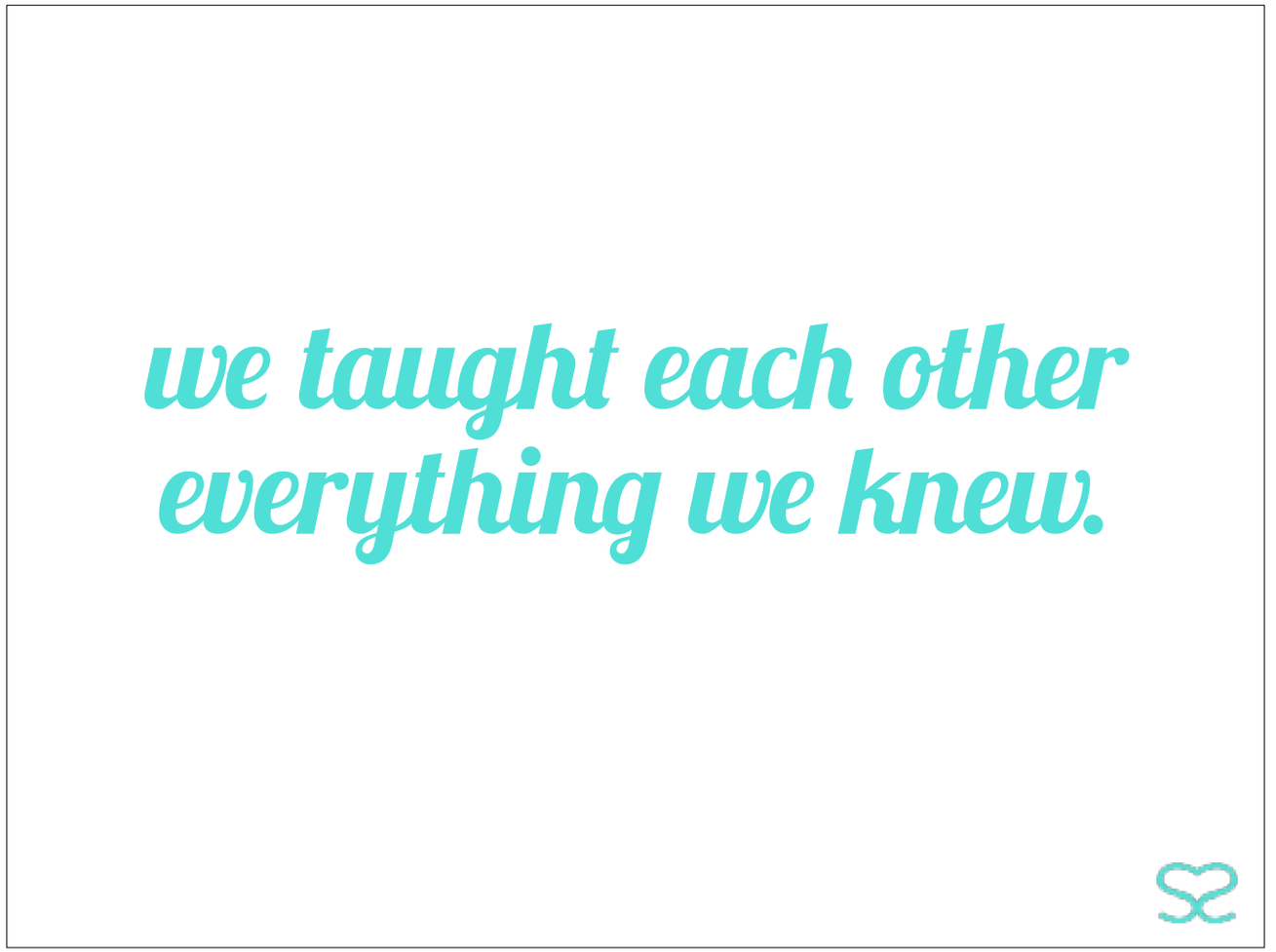

"LIGHT BULB"

it's impossible to be bored.

unless the internet is down…

my outlook has been THIS ever since
it's impossible to be bored

in college, people were motivated
we were drunk, but motivated
granted, we had more free time, but the excitement was there
we had meetups and LAN parties and linux user groups and game nights



we taught each other everything we knew
whether it was learning python or teaching someone how to set up a game server
or figuring out how to hack quake 3 so we didn’t have to buy it at walmart
$10 was a lot of money in college!
we were hungry for it
then i graduated and started my first big girl job… talk about a wake up call

in the real world?

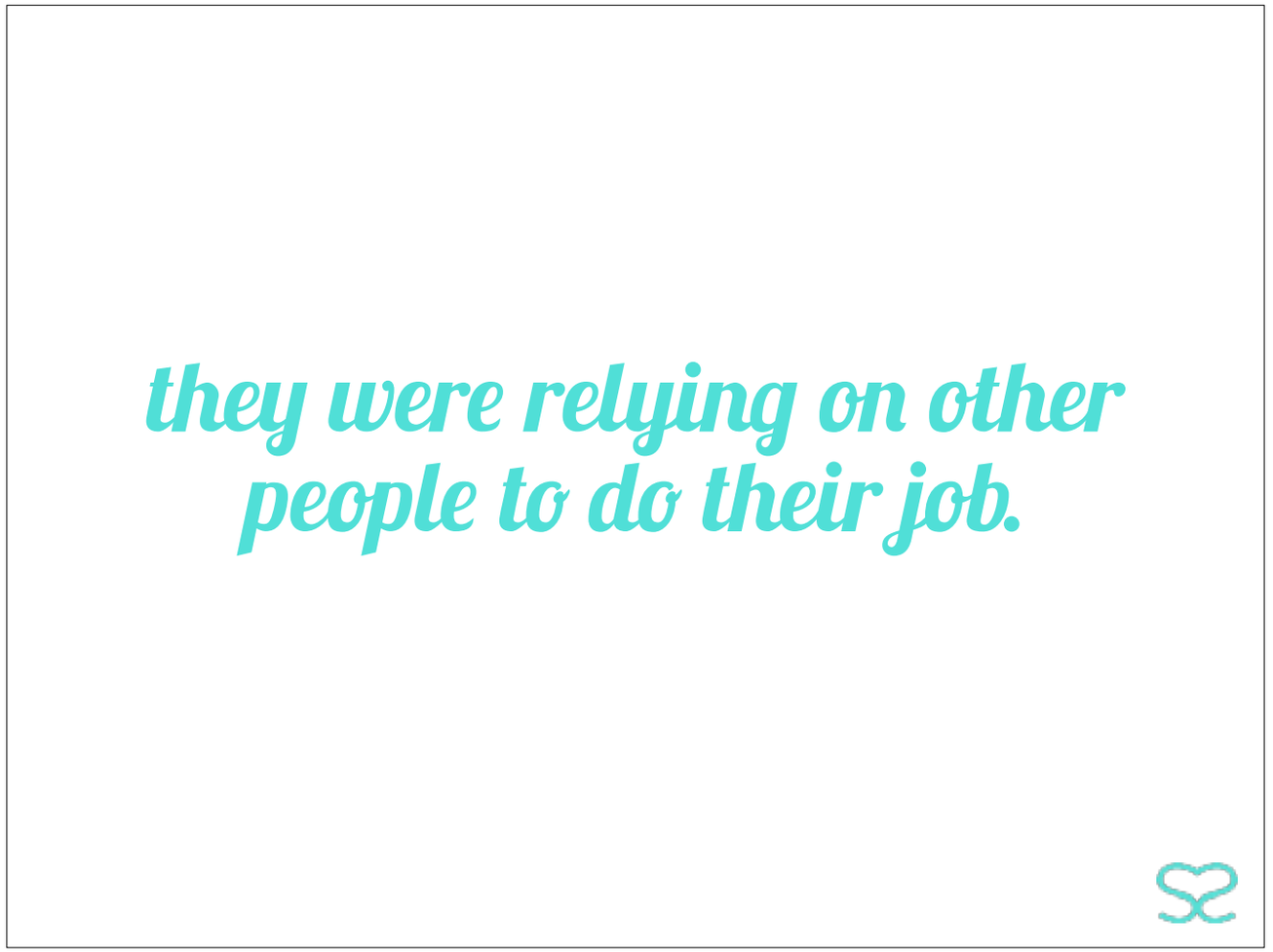
not so much...

in the real world? this was not usually the case
no one was hungry for it

COMPLACENCY.

this. this became the biggest obstacle
the general mindset was complacency
“i have a steady job, don’t need to learn anything new”

there were a lot of “guys”. like “oh, he’s our STIG guy”, and “this guy runs gold disk” and the “solaris guy” and the “windows guy” and the “network guy”, “retina guy”, “documentation guy”
people got really good at 1 THING and that’s just all they did
there were still people who shared the itch to keep learning
but overall… you had to work a hell of a lot harder to get others involved a lot of people became pigeonholed
often by their own doing, intentional or not
they would get stuck in their role because that’s what they were really good at
that’s all they knew, they didn’t care to branch out



and that just made them have to rely on other people to do their job
they couldn’t do anything outside their bubble of knowledge
coworkers were only worried about the task at hand

developers didn't understand the platforms they were developing for, or on.

they just wrote the code, and handed it off.

(or they'd spin up an ec2 instance, not change anything, and label it "production DO NOT TOUCH!!!" *cringe*)

(i hear "i hate developers" a LOT!)

now, in this case, i'd actually prefer the developer writing the code and handing it off to someone, versus the alternative
but that goes against the purpose of this talk which is… to be multidirectional and learn OTHER things that relate to your work
i've come into a lot of projects late in the game, when developers have already stood up multiple environments. one of which is always labeled "production". gotta give them kudos for trying… BUT
nothing is documented. it's typically living on ubuntu, not hardened whatsoever, no iptables, root login enabled, password login enabled, no key-based auth (thank you amazon for making this standard), port 22 wide open. no load balancing. no high availability set up.
no failover. and almost ALWAYS… no backups in place. zero monitoring. zero alerts. but… DON'T YOU DARE TOUCH IT. because it's production.
i hear security folks say "i hate developers" a lot, for this reason. because security is almost always an afterthought
if you're gonna do something, do it right. this should be baked in from the beginning

security folks didn't understand the platforms they were securing.

they just told the sysadmins to make the changes.

(or someone would give them access and they'd attempt to do it themselves… *more cringing*)

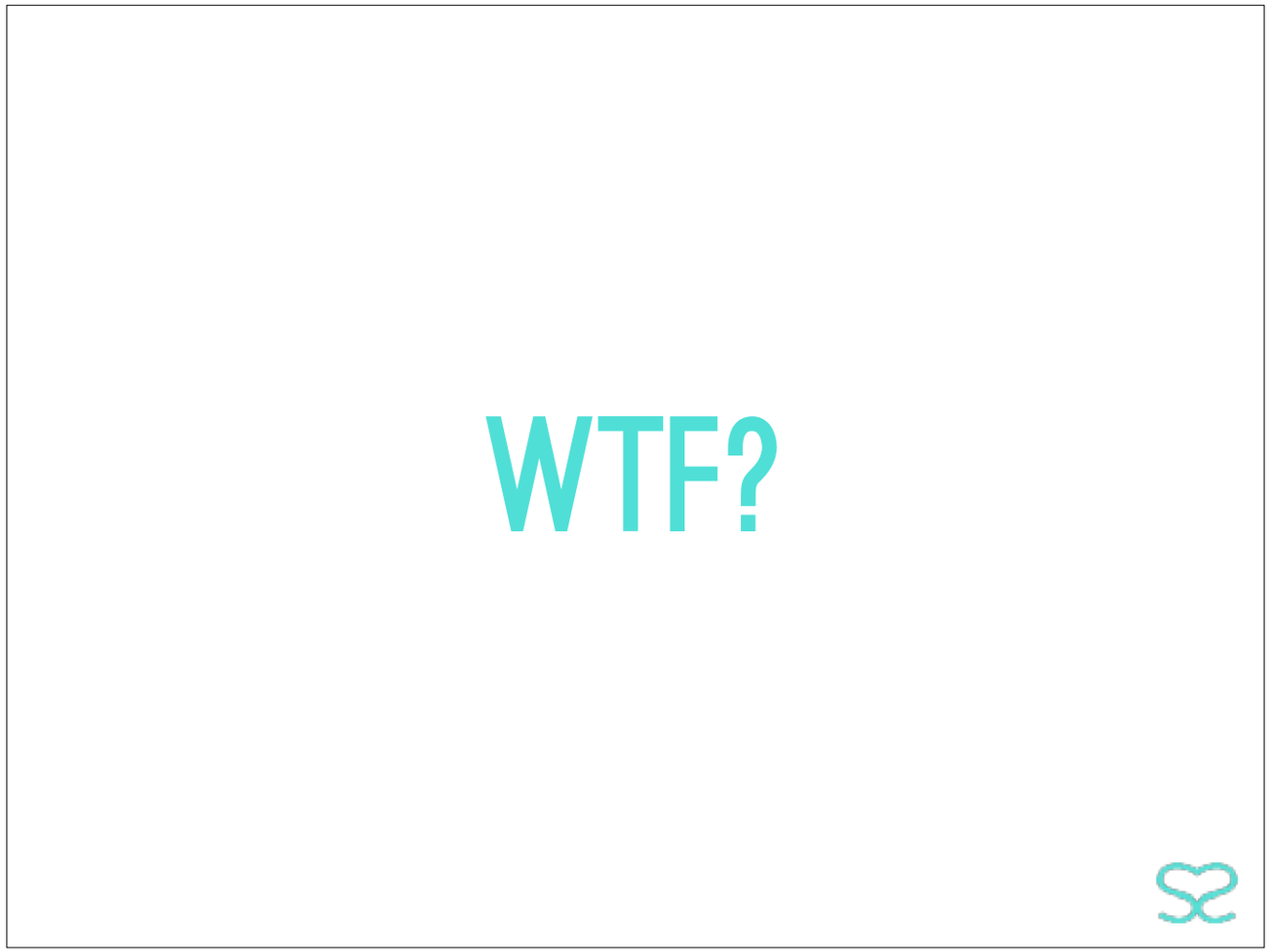
this is where most of my beef lives
this just blows my mind
i don't understand the people who work in security, with no development or programming or systems background
it's like they woke up one day and said, "hey, i'm gonna go try to do information security"
how do you get into security when you don't know what you're securing? again, kudos for taking the leap, but do your homework
i worked with one security engineer after another… mostly linux projects… zero knowledge of linux. or anything technical it seemed
you find one worth keeping and you hold on for dear life
so my initial reaction was almost always, ok, let's teach this person
if it were me, i'd appreciate the gesture
but this rarely went over as expected. as it turns out, you can't teach people who don't want to learn, or just don't give a shit
circling back to the whole complacency thing

IA didn't understand the policies they were enforcing, or even know what they did.

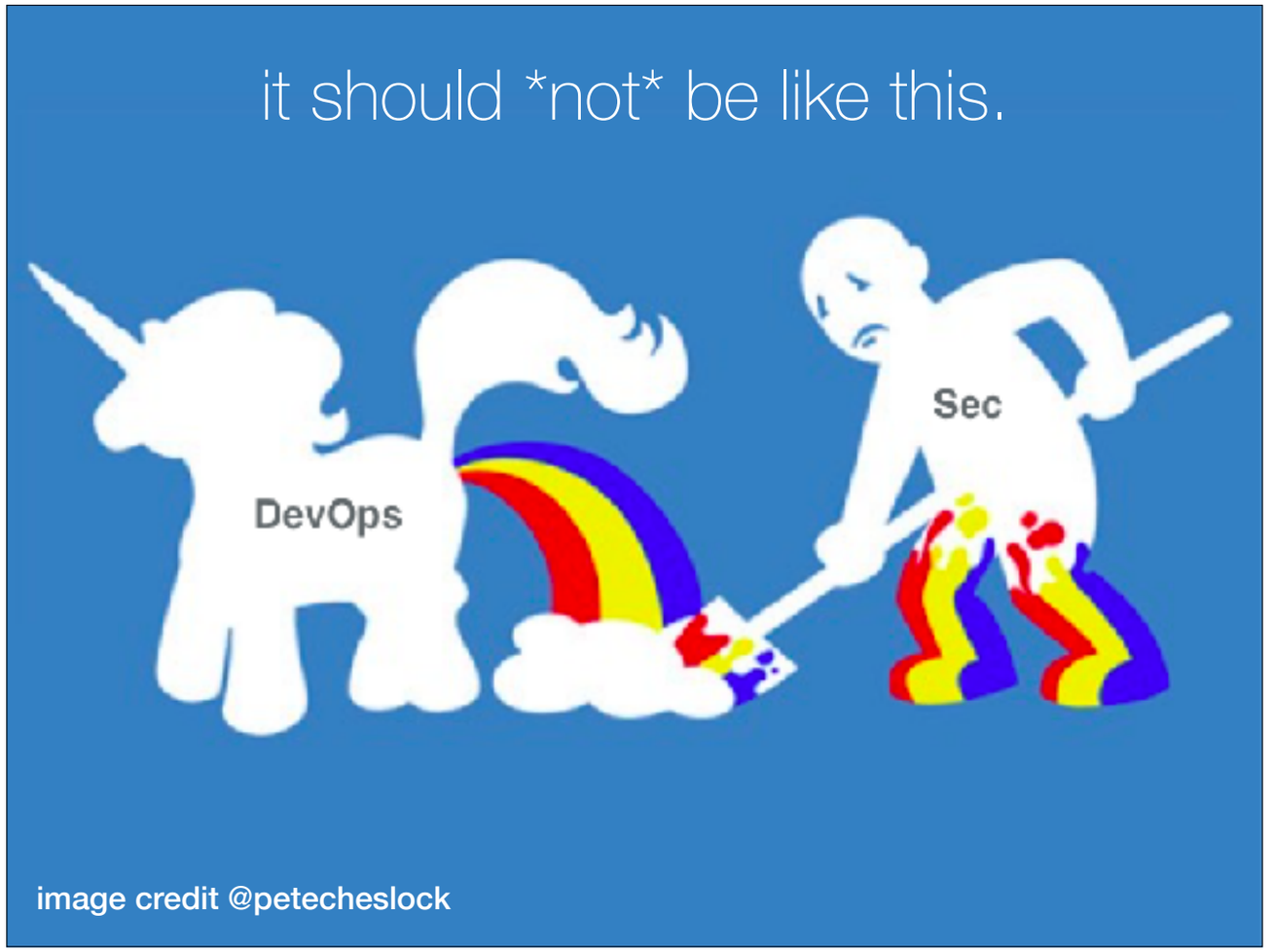
they just told the teams what access controls weren't met because…
that's what the scan results said.

facepalm

this kind of ties into the last slide
but… if you're IA, and your purpose on a project is to perform audits and help make recommendations in regards to implementing security policies, wouldn't it make sense to understand what those policies actually mean?
additionally, if you're running scans on servers to determine compliance with said security policies, wouldn't it make sense to be able to translate those scan results into actionable items?
one would think…

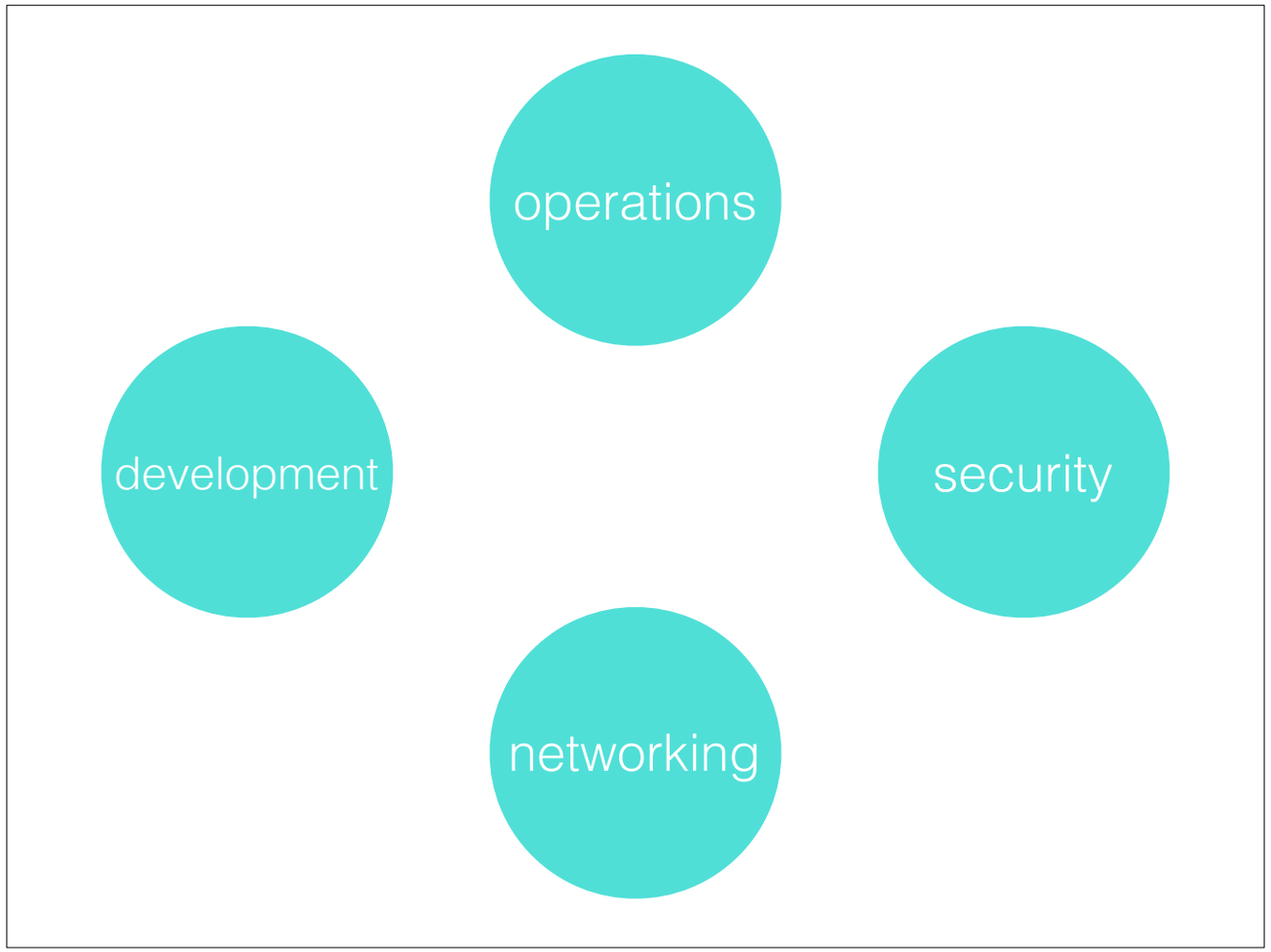


seriously



it should not be like this

don’t be that guy



this is where the job security comes in

if you can connect some of the dots, or even bring them a little bit closer together
learn just a little bit about what enables you to do your job
or what the person next to you does
or what the operations team does
or what your security person does
or what your network person does

if you can fill more than one role in any capacity,… it's a step in the right direction.

I GET ASKED THIS ALL THE TIME

how do i get into [subject]?

and then i cringe, because
A) i know they won't take my response seriously, and
B) i know my response gets bitchier each year

2002

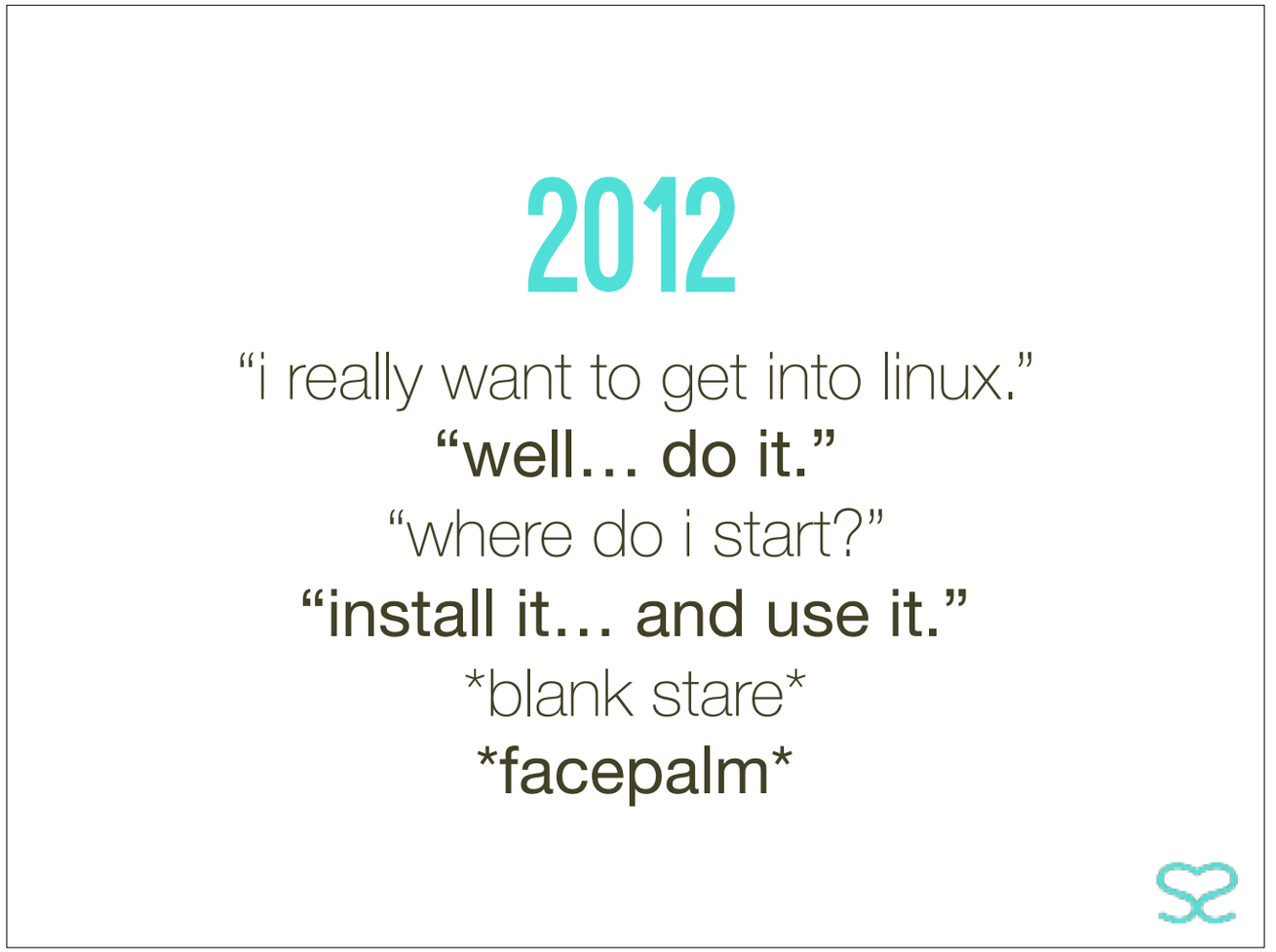
"let's meet after school once a week. i can teach you."

2002. highschool. this was my initial response, and i tried this. twice… it seemed like a good idea at the time. never again
in one case, it failed miserably. no ambition. no focus. it was exhausting.
in another case, it failed miserably. i later learned that the guy was trying to date me, and actually knew more than i did about c++ at the time. so that was embarrassing.
on the bright side, i got a lot of free cookies out of it since we met at starbucks
either way…
none of it was worth the frustration

2007

“there are tons of training materials online, but most of what i know i learned just tinkering and working on the side. maybe you need a side project. have you thought about that?”

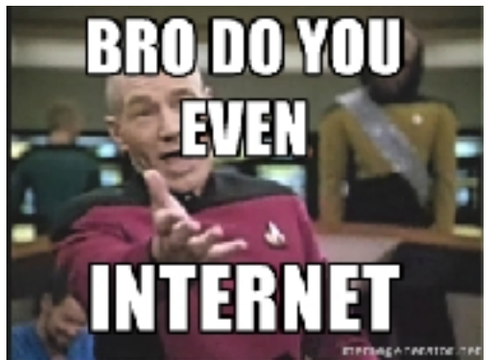
fast forward to 2007. college.
at this point, i had 3 years of tech support under my belt
i never wanted to teach anyone, or explain ANYTHING, ever again if i didn’t have to
especially if i’m not getting paid for it
so i’d say something along these lines
not bitchy yet, just… a helpful suggestion



2012
i’ve been out of college and working for 3 years
which means, most of my friends/colleagues are in the same boat, and older
this one always kills me
rarely does anything actually happen

2015

"do you even internet, bro?"



seriously
EVERYTHING IS THERE
READ SOMETHING

lots of this
my husband used to get REALLY mad when during the first year we dated
he was trying to get into linux and security, and kept asking for help
and this was almost always my response
he did not find it as amusing as i did

YOU’RE DOING IT WRONG

“how do i get into security? how do i get into android development? how do i get into hacking? how do i get into linux?”

NO

you’re asking the wrong questions

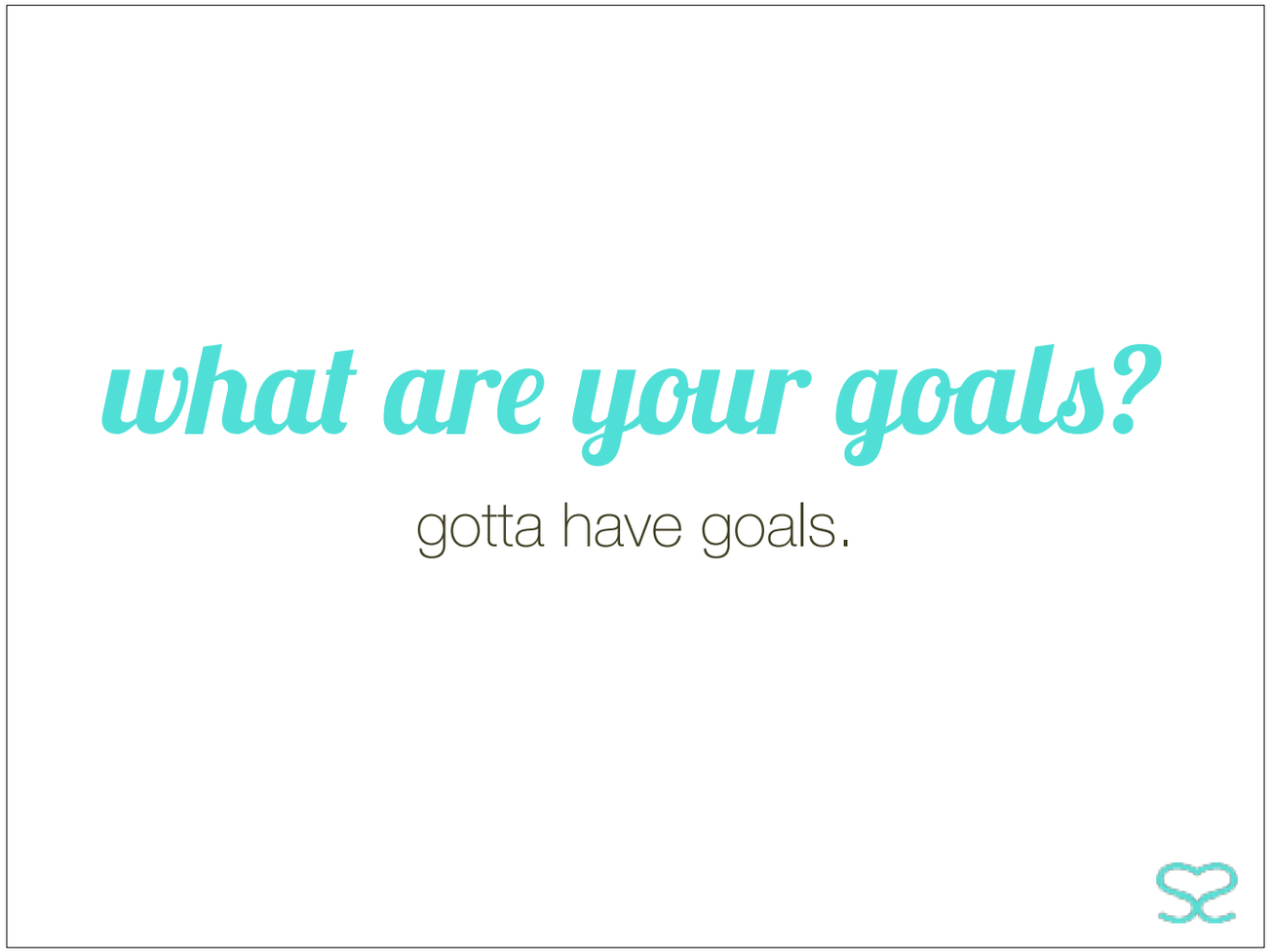
be curious, be hungry, be passionate

jump in and figure this shit out

put some TIME into researching things before you go asking for help

people will be much more willing to help when you show you’re willing to put forth some effort and have already done some digging on your own

and chances are they won’t just send you to a let-me-google-that-for-you address



so… figure out your goals
i ask myself this often. probably 2-3 times a year, at least
i never have a solid answer. i still have NO idea what i want to do when i grow up
but i'm really glad that i've written down my goals a few times over the years
and i can go back and revisit them to see if i checked anything off the list

THEN

my goals after college…

- get security certs (CISSP, security+, CEH)
- get RHCE
- don't stay more than ~1 year at first job
- make X dollars before age 30
- **make myself irreplaceable wherever i land**

get security certs
	this is DoD land after all, and these are pretty much prerequisites
	added job security
get RHCE
	because i'm obsessed
don't stay more than 1 year at my first job
	get experience, jump around on projects,
	each hop is a new challenge and a pay bump
make X dollars before 30
	i don't want to rely on anyone
and most importantly
it's not a long list, but bottom line: it requires a continued effort on my part to keep learning

NOW

my goals now…

- get RHCA
- get AWS professional certs
- get more red team experience
- keep moving up, but don't sacrifice technical/ hands on work
- **make myself irreplaceable wherever i land**

i'm still working towards my RHCA, it's a long road
i want to get an AWS professional cert, or certs… not because it's another certification, but because it MAKES me learn it better
    and the fucked up side of me thinks it's fun
i want to more red team experience
    infosec is something i love doing, and it's also something i don't get to do a lot of at work. at least not this side of it
    i sign up for any CTF or hackathon i come across
keep climbing the ladder, without sacrificing skill sets —
    this is huge for me, i don't like people, it's why i'm in this industry, just… let me stick with computers
and, once again, last but not least
i don't know where i want to be 5 years from now, but i do know where i don't want to be
and that is obsolete

what does this list look like for you?

so… what does this list look like for you?

i don’t care if you’ve been doing this shit your whole life.

there is ALWAYS something you can do better, always something new to learn

ASK YOURSELF

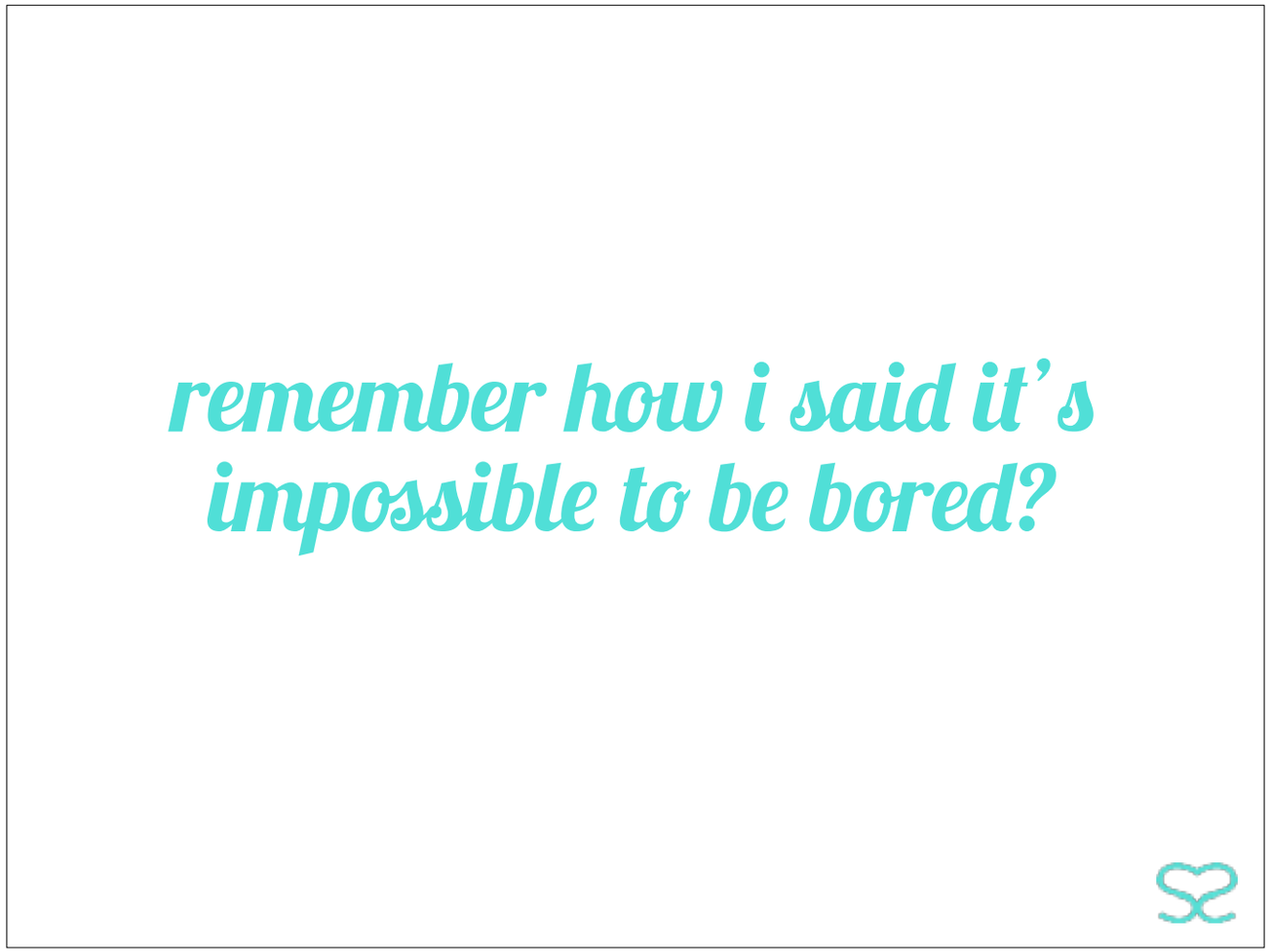
what do you want to do?
what does your company do?
what do other people at your company do?
what stacks do they use?
what APIs do they use?
what languages?
what are you securing? apps? servers? websites?
do they do any penetration testing?
what operating systems? windows? linux?
where do they live? on site? azure? AWS? google? heroku?
what do the networks look like?
what kind of compliance and security requirements are in place?

that gives you like… at least 10 things to start with right there
so now… take those questions/answers and make it more specific

MAKE YOUR TO-DO LIST

how do i code in language X?
how do i deploy an application in language X on server Y?
how do i set up and secure the database?
how do i set up and secure the network?
how do i secure the server(s) it's all running on?
how do i scale it?
how do i automate it?
how do i monitor it to make sure it's always up?
how do i implement high availability? failover? backups?
how do i monitor performance?
how do i test how secure it is when it's all said and done?
how do i perform an audit when people ask me how secure it is?

even a simple hello world app in just about any language can allow you to learn all of the above, to some degree



this is why impossible to be bored

even if your current job doesn't fit this model,

tailor your list to the career path
you'd *like* to have.

what would you rather be doing? where do you want to end up?
take matters into your own hands.

this should go without saying, but there will be failures

S/FAILURE/EXPERIENCE

- restarting (not reloading) iptables in production… in the middle of the work day
- rebooting the wrong server… in the middle of the work day
- pushing puppet changes to the wrong environment
- taking the wrong server off the load balancer
- not putting SSL certificate expiration dates on the calendar (FML)
- that one time i didn't use visudo
- not using screen when i really, REALLY should have been
- restarting apache after a "harmless" change… only to find out i made a typo and shit's BROKE! (apachectl configtest FTW)
- locking out local accounts accidentally, not realizing it was fubared until i ran an ansible playbook and NOPE
- taking a 1 week course and 6 hour red hat cert exam at 7 months pregnant <— actually worse than labor
- mistaking a female client for a male… OH MY GOD

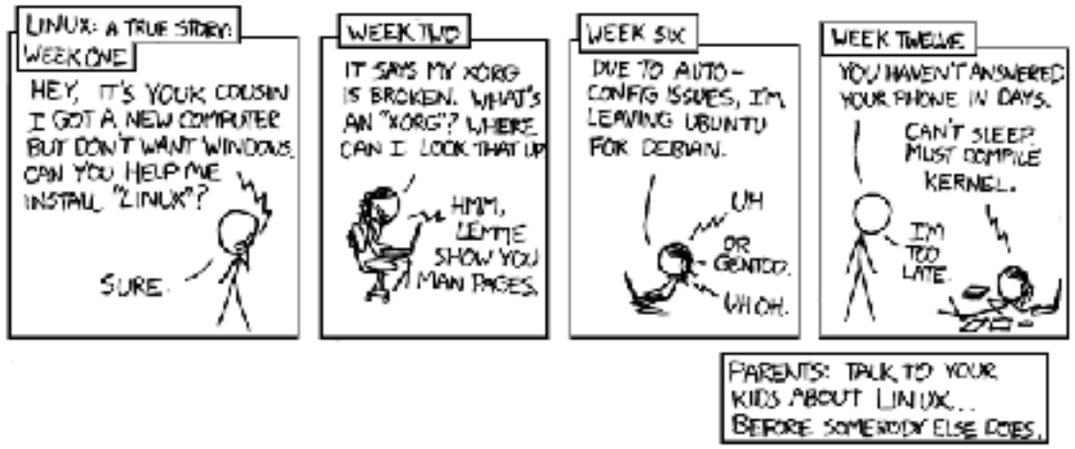
i have fucked up… many times in my career
see how small the font is? this doesn't fit on one page. there's plenty i've left out
but… these are the things you never forget, and [hopefully] never repeat
because you remember the very real fear that hit you the moment it happened
fix it, learn from it, move on
just call it "experience"

“i don’t have time.”

neither do i.

another thing i hear all too often…
we can all throw out the “i don’t have time” excuse. i say it, too
there are truly, not enough hours in the day
but if you want to do something bad enough, you’ll make it happen
sleep is overrated

LEARN ALL THE THINGS.



https://xkcd.com/456/ - cautionary

this is one of my favorite xkcd's, and is actually a fairly accurate depiction of me and a buddy of mine in 04

so now you have a plan, and a list, and hopefully… some motivation. time to learn all the things
the best part is, there is no rush or timeline on any of it
i have so many unfinished projects i have in my dropbox
things-to-try bookmarked in chrome and pocket
todo lists in evernote
i listen to podcasts in the car.
i stay up late and break shit
i try to participate in every hackathon and CTF i can
i go to code camps
i beg for training
i try to go to every conference i can get to within reason
i stay up late most nights working on SOMETHING
whether it's freelance work or trying out some tutorial or playing with new toys in AWS or working on an app

STORYTIME! some of you may have heard this. but it has a happy ending and a point

the whole reason i even learned android was because my husband bought me a tablet for christmas back in 2010. i wanted to play angry birds on it, instead of my phone. i didn't want to have to replay it to get all my gold stars back. i was stupid obsessed. at the time, all the apps in the market were just to transfer your scores to your SD card, and it's a pain in the ass to swap cards all the time. so instead i relearned java (because that makes sense). it had been roughly 7 years since i last used it.

taught myself how to android, built 4 apps and a PHP API
users could log in, upload their scores from one device, and download them onto the other

i learned a shitload, and even made money (thank you, rovio, for the free marketing)
that november, my dog got diagnosed with cancer
the money made from the apps was enough to cover more than half of my dog's surgery, chemotherapy, radiation, and medication.

i was too lazy to swap out SD cards, but spent god knows how many hours drawing stupid birds in illustrator and writing these apps.
in the end, i learned how to make android apps
got a little bit of API development under my belt
because of that, i was able to switch it up a bit, show off new skill sets
i became more of an asset to the company

if i hadn't taken that step, i would never have joined the mobile team at SPARC
i never would have worked on stre.am, and i would still be saying "i've never been to an NFL game"
i never would have made the app for DEF CON
i wouldn't know a LOT of the people i know now
life would be very different

so… opportunities are everywhere. find a problem, or something that you can get EXCITED about
and learn something

you never know what doors will open up.

DOCUMENT ALL THE THINGS.

i can only think of one thing worse than doing all that work and then forgetting all the little pieces of how you got there
and that is being hit up with questions by all the people who have to use what you built

i have actively tried to get my hands in just about every project to come through SPARC's doors in the past 4 years, in some way or another.
most of it is building and securing the infrastructure, which… is all over the place because the projects are all over the place
some are hosted in amazon, google cloud platform, digital ocean
some use the play framework, tomcat, weblogic, nginx, .NET apps, LAMP stacks, MEAN stacks
mongo, mysql, oracle
i've built and deployed a lot of shit, and all of it has to be used by someone that isn't me

sharing your knowledge
does not reduce your job
security. it amplifies it.

i quickly learned to write everything down. EVERYTHING
for personal stuff, it's all in evernote
for clients, it's either a google doc or a google spreadsheet and it's shared with everyone on the project
this is a basic rule of CYA — if i'm gone, someone who's never used it needs to be able to figure it out

TEACH ALL THE THINGS.

so… now you have all this documentation.
and the nice thing about having all of this documentation, is now you can teach people and a lot of the work is already done for you

share what you know with your coworkers. your friends. your teammates
present something at a meetup
do a lunch and learn
explaining it will only make you learn it better
not only that, but it builds a reputation
it gets you out of your comfort zone

DO NOT let your efforts go unnoticed

after my first year or so doing infrastructure/ops on several projects, i put together an almost 2 hour talk about AWS -> LAMP -> wordpress -> bootstrap. turns out, that's a shitload to cover in that amount of time with varied skillsets

and even though i didn't get to finish, the talk was good, and bottom line…. PEOPLE LEARNED. and LISTENED. and got interested in something new

RINSE & REPEAT.

as the saying goes, practice makes perfect
and if you’re not keeping up, you’re falling behind
the technology industry is a blessing and a curse

to quote HD moore, “if you don’t think you are a newb, you’re not trying hard enough.”

"i feel dumb everyday, and i'm proud of it."

-@SHORTXSTACK

TAKEAWAYS

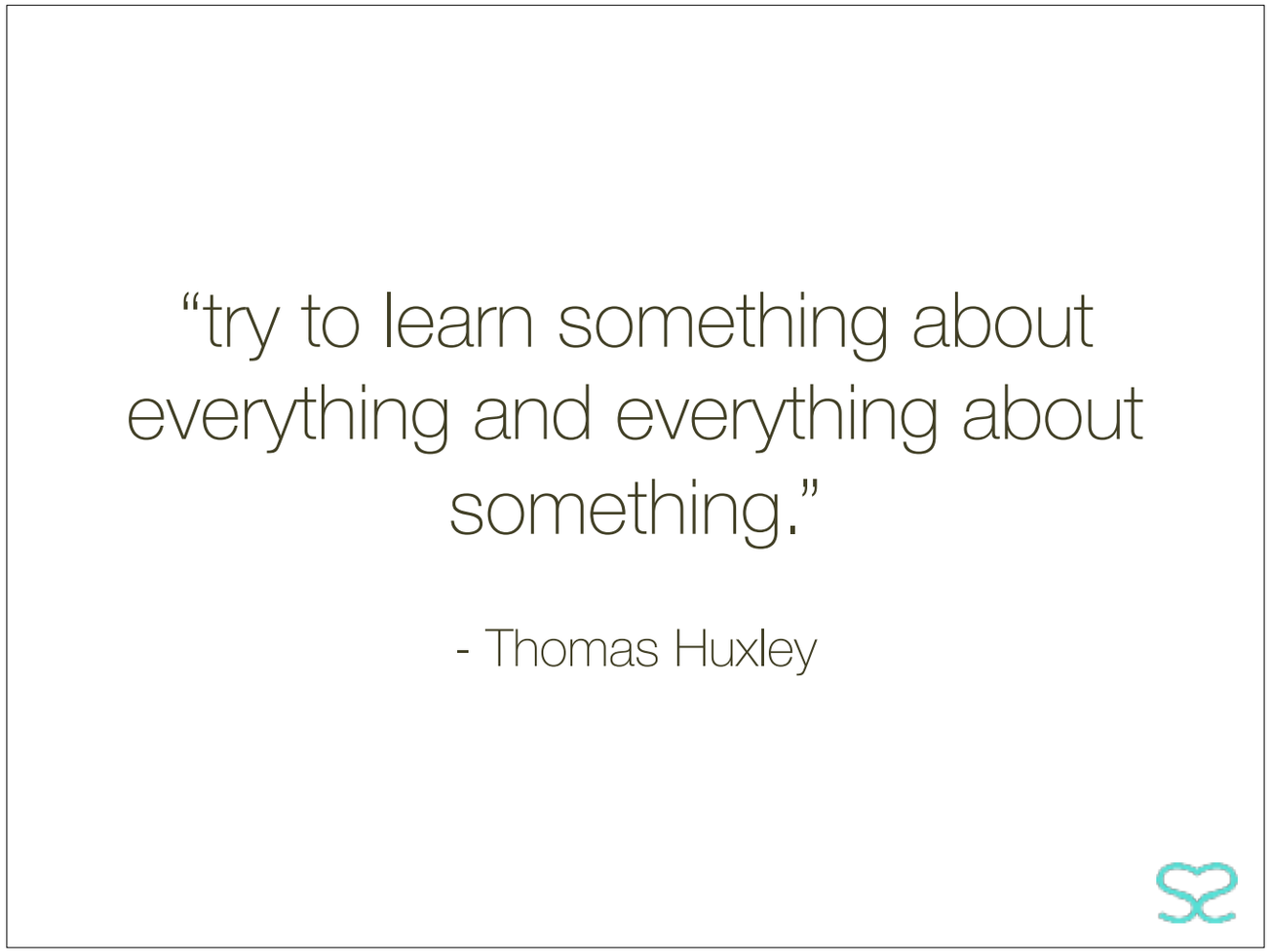
be willing to put in the extra effort and hours.

be willing to learn the things that other people don't want to learn.

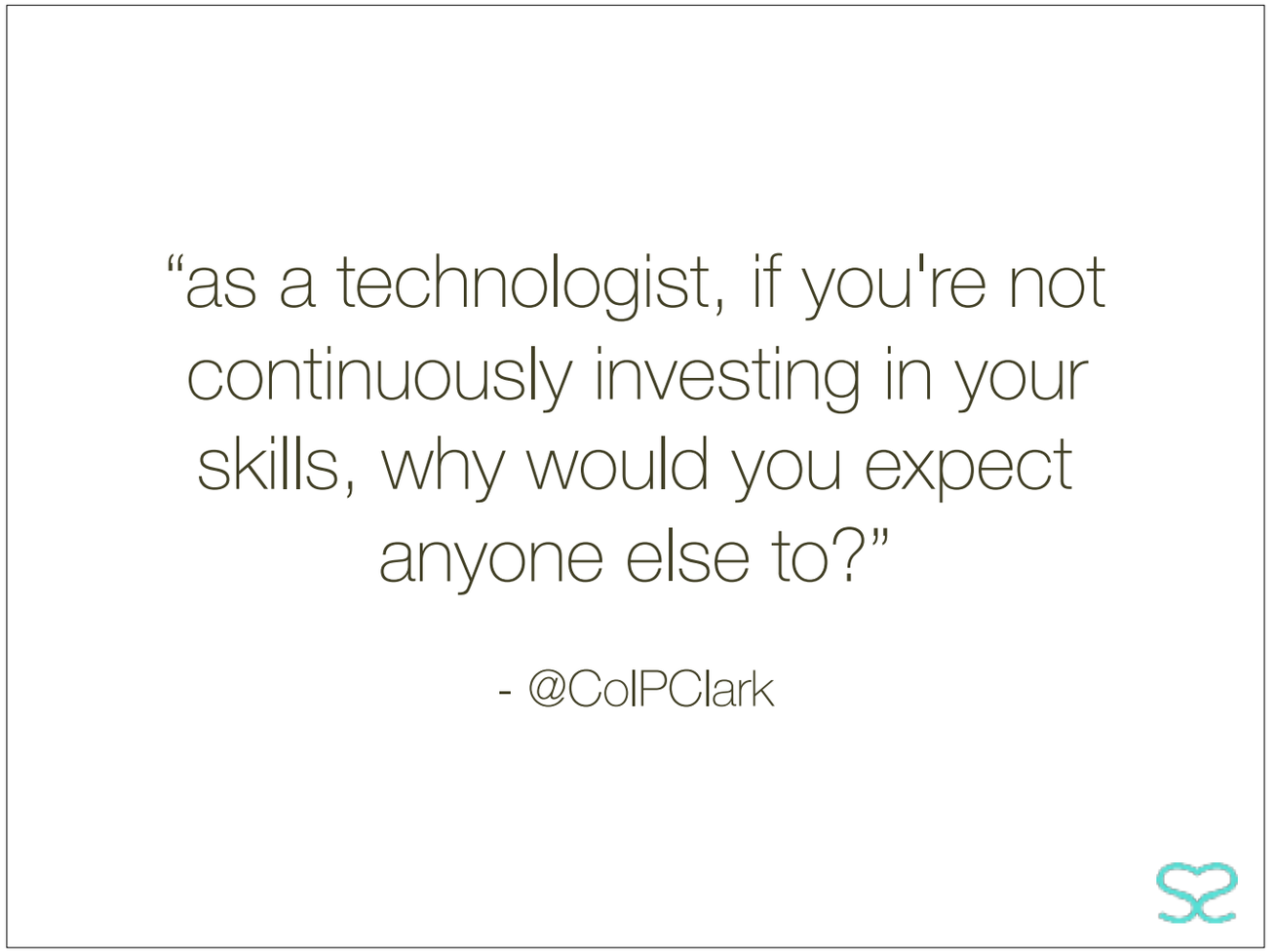
find the gaps, and fill them in.

make it fun.

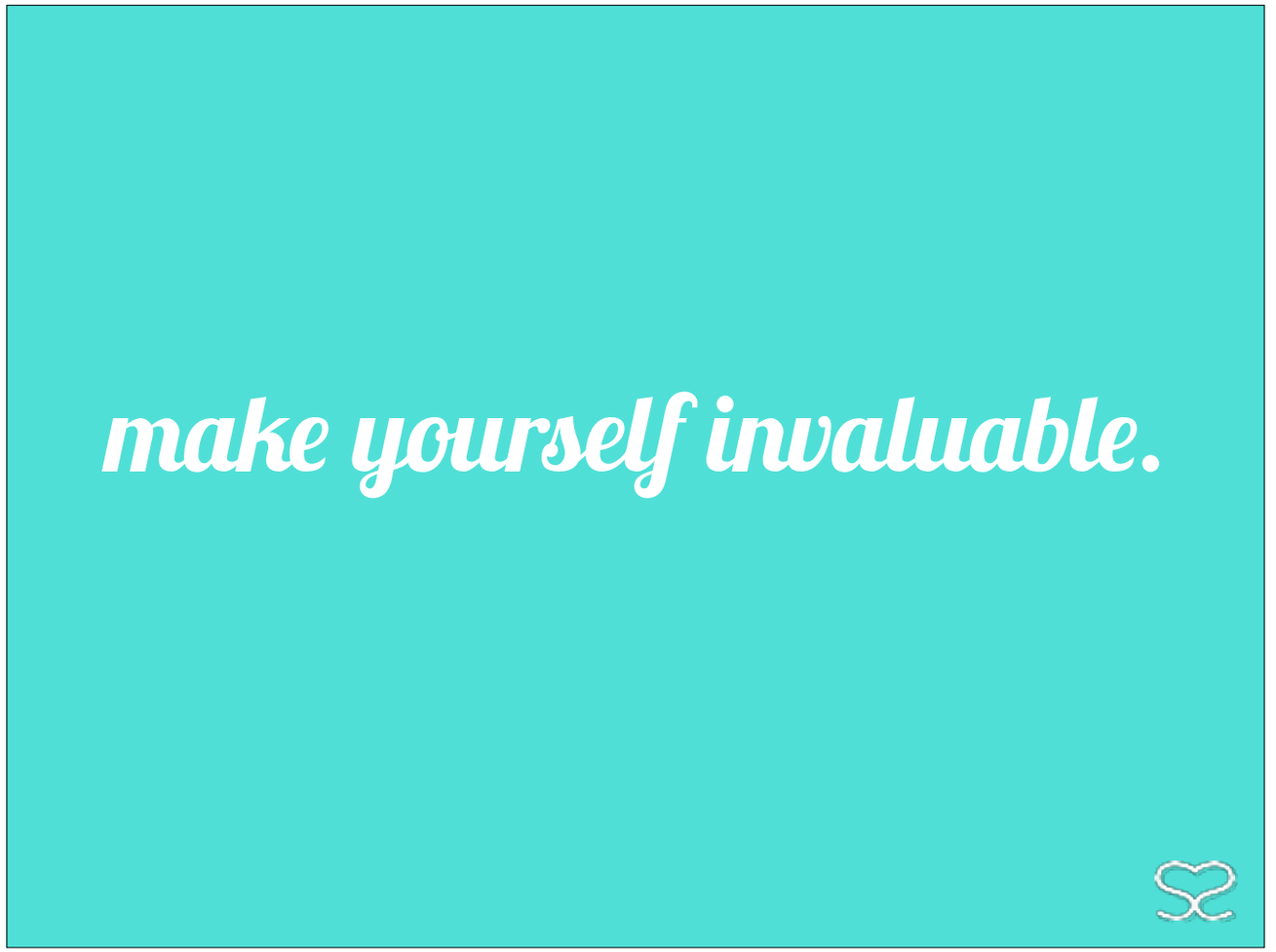
so… a few takeaways
be willing to put in the effort and extra hours
be willing to learn the things that other people don't want to learn
find the gaps, and fill them in



i’ve always liked this quote
just food for thought



and this one… i love this one. saw this and retweeted it a while back
and it’s so true. more doors will open, people will be more willing to help you…
if they know you will take full advantage of the opportunities they put in front of you



my message over the last 60 slides is this:

be motivated
push yourself
keep learning
and put that knowledge back into your company
make yourself invaluable

THANK YOU :)

WHITNEY CHAMPION // @SHORTXSTACK