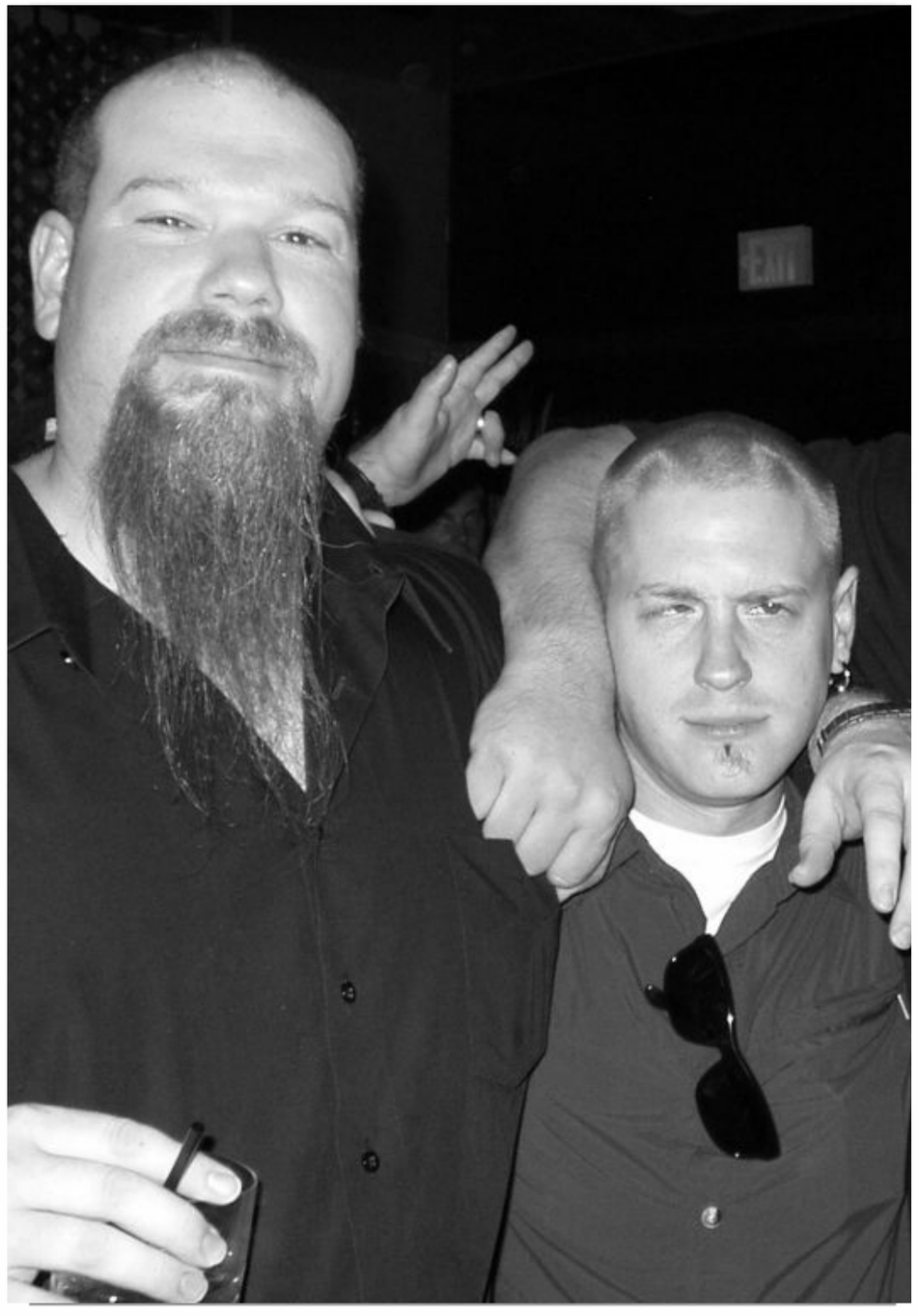
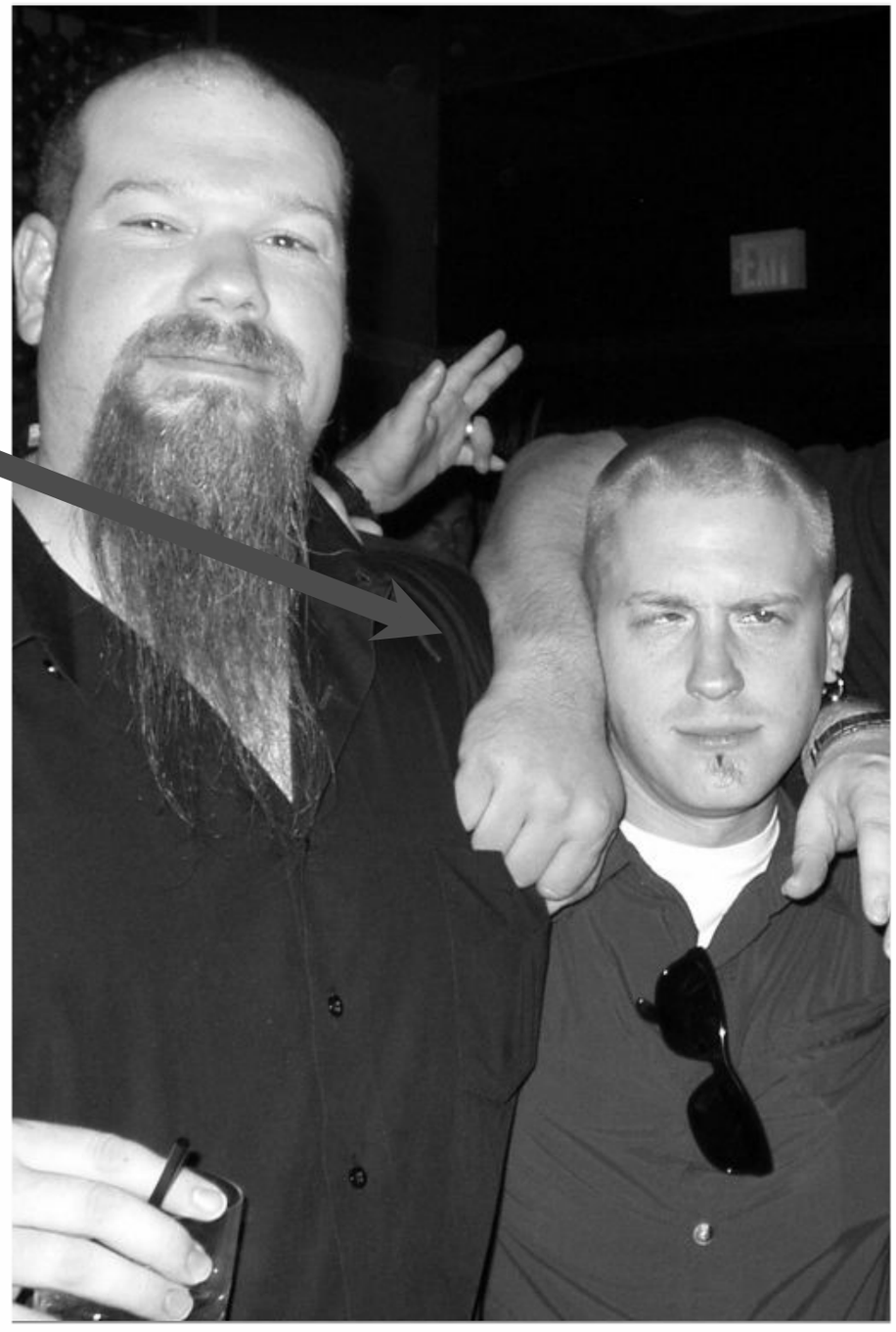
the pentest is dead, long live the pentest!

Taylor Banks
& Carric

carric



taylor



Overview

- 1 the pentest is dead
  - 1.1 history of the pentest
  - 1.2 pentesting goes mainstream
- 2 long live the pentest
  - 2.1 the value of the pentest
  - 2.2 evolution of the pentest
  - 2.3 a framework for repeatable testing
  - 2.4 pentesting in the 21st century and beyond
- conclusions

Taylor's [Don't Give Me Bad Reviews Because I Made Fun of You] Disclaimer:

I'm about to really rip on some folks, so I figure I might as well offer an explanation, (and some semblance of an apology) in advance.

Contrary to implications in later slides, there **ARE** actually a handful of really smart people doing pentests, writing books about pentests and teaching classes on pentesting, who despite their certifications (or lack thereof) actually know WTF they are doing.

Those are not the people I'm talking about.

This presentation picks on the other douchebags who *call themselves* pentesters. As such, I plan to talk about what you (and I) can do to take the industry back from the shameless charlatans who've *almost* been successful in giving the rest of us a bad name.

Yours very sincerely,
-Taylor



Part 1

the pentest is dead

the pentest is dead

- history of the pentest
- pentesting goes mainstream



1.1

history of the pentest

the timeline

- 1970 - 1979 — Captain Crunch, Vin Cerf, Blue Boxes, Catch-22
- 1980 - 1989 — CCC, 414s, WarGames, LoD, MoD, CoDC, 2600, Phrack, Morris worm, Mitnick v MIT/DEC, Poulsen, CERT
- 1990 - 1999 — Sundevil, EFF, LOD vs MOD, Poulsen, Sneakers, **DEF CON**, AOHell, Mitnick, The Net, Hackers, MP3, RIAA, Back Orifice, L0pht, Melissa
- 2000 - 2008 — ILOVEYOU, Dmitry Sklyarov, DMCA, Code Red, Paris Hilton's Sidekick, XSS, Storm Worm, Web2.x, AJAX

on semantics

- we're talking about "classic" [network-based] penetration testing
- we're *not* talking about 0-day vulndev, on-the-fly reversing, etc
  - (if that's what you were looking for, you can skip out to the bar now)

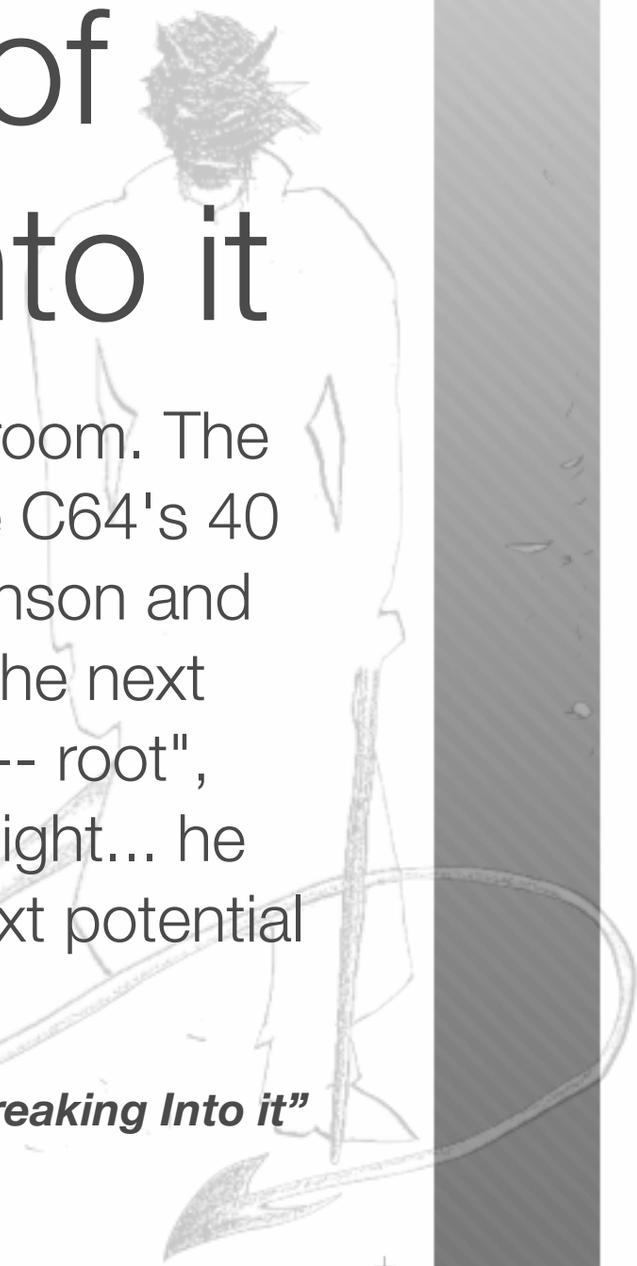
a brief history: the pentest

- early pentesting was a black art
- nobody saw the need; employees were trusted
- information security was poorly understood,
  except by the learned few
  - **The Hacker Manifesto**
    by The Mentor
  - **Improving the Security of Your Site by Breaking Into It**
    by Dan Farmer and Wietse Venema

the hacker manifesto

- Says *The Mentor*, “I am a hacker, enter my world…”
- Provides a voice that transforms a sub-culture:
- “Yes, I am a criminal. My crime is that of curiosity. My crime is that of judging people by what they say and think, not what they look like. My crime is that of outsmarting you, something that you will never forgive me for.”

improving the security of your site by breaking into it



"A young boy, with greasy blonde hair, sitting in a dark room. The room is illuminated only by the luminescense [sic] of the C64's 40 character screen. Taking another long drag from his Benson and Hedges cigarette, the weary system cracker telnets to the next faceless ".mil" site on his hit list. "guest -- guest", "root -- root", and "system -- manager" all fail. No matter. He has all night... he pencils the host off of his list, and tiredly types in the next potential victim…"

Courtesy of ***"Improving the Security of Your Site by Breaking Into it"***

more history

- Sterling's "The Cuckoo's Egg" documents the discovery, identification and eventual arrest of a cracker
- We begin to research and recognize "cracker activity"
- Bill Cheswick authors "An Evening with Berferd In Which a Cracker is Lured, Endured and Studied"
- While a student, Chris Klaus gives us Internet Scanner 1.x ;)
- Cheswick and Bellovin author "Firewalls and Internet Security"

enough history, i thought there were war stories?!

- once upon a time…
- pentest, circa 2000
- public school system with sql server, public i/f, sa no passwd
- thousands of vulns, top findings:
  - blank or weak passwords, poor architecture and perimeter defenses, unpatched systems, open file shares, no formal security program or awareness efforts
- what grade would you like today?

other fun shit... that *used* to work

- IIS Unicode
- Solaris TTYPROMPT
- froot
- blank passwords
  - ‘sa’
  - Administrator

whitehats by day…

- early on, true penetration testing skills were learned mostly in and amongst small, underground communities
- those who were good were often that way because their hat's weren't always white

early methodologies

- when i began performing penetration tests professionally, there was no semblance of a commonly-accepted methodology, so i wrote my own
- in fact, i wrote methodologies used successfully by three companies based entirely on my own early experiences
- in late 2000, pete herzog (ideahamster) released the first version of the open source security testing methodology manual (the OSSTMM - like awesome with a T in the middle)

osstmm v1.x

- the earliest editions of the osstmm were helpful, and showed promise, but had a *long* way to go before they would replace my own hand-written process/procedure documentation
- even still, the effort was laudable, as no other similar effort of any significance otherwise existed



a service in search of a methodology

- the real problem with a generally-accepted methodology, however, was rooted in ruthless competition
- in 2001 there was a lot of money in pentesting, and a lot of competition for the mid and large enterprise
- in other words, it was “job security through process obscurity”
- if you were good at what you did, as long as nobody else could produce as thorough results with as effective remediation recommendations, you won ;)

a stain on your practice

- unfortunately, “job security through process obscurity” ultimately hurt us all, as not only were no two pentests alike, but they were often so radically different that no one could feel confident or secure with only a single organization’s results
- and if it ain’t repeatable, it ain’t a pentest… it’s just a hack
- thus it was time to embrace the osstmm to help ensure a basic set of best practices, necessary processes, and general business ethics that anyone worth their salt should possess

progress?

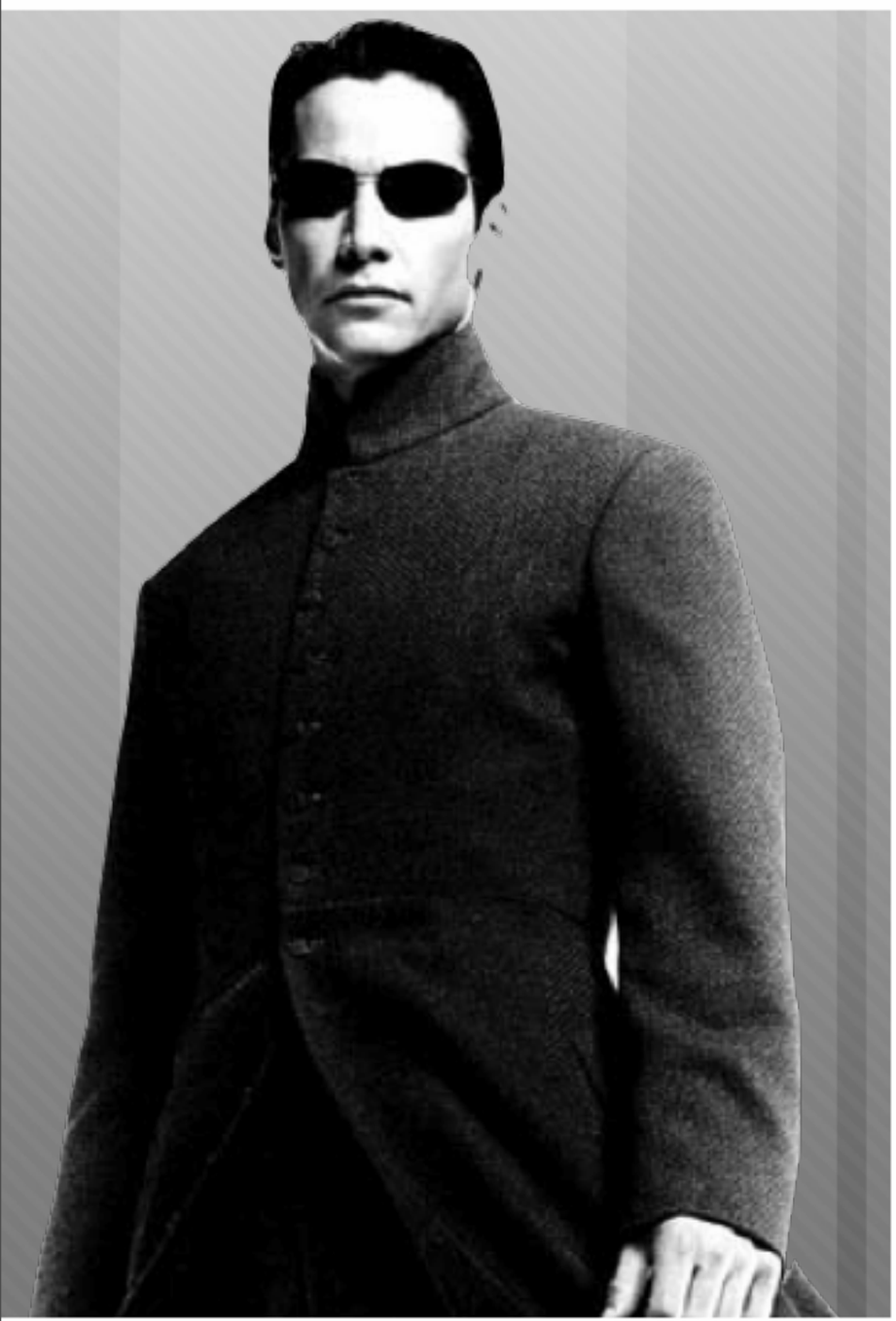
- ISACA
- ISECOM
- CHECK
- OWASP
- ISAAF
- NSA
- TIGERSCHEME

so where does pentesting fit?

we don’t know, but pentesting is cool!

(more on this later)





1.2

pentesting goes mainstream

pentesting goes mainstream

- by 2000, pentesting began to gain more widespread appeal
- assessment tools have come a long way since then (hell, even portscanners used to be a pain in the ass)
- their effectiveness, efficiency and ease of use have improved:
  - take nmap, superscanner, nessus, caine/abel, metasploit
- with easier and more readily available tools, more practitioners emerge, though most lack both experience and methodology

hacking in the movies

- WarGames
- Sneakers
- Hackers
- The Matrix
- Swordfish
- Antitrust
- Takedown



the lunatics have taken over the asylum!

- you better get used to it
  - in this segment of this industry, you'll likely compete with idiots
  - why? because there are thousands of people who mistakenly believe they're good hackers (this audience of course excluded ;)
  - unfortunately, although ego is often a by-product of a good hacker (or maybe even a factor of?), i can guarantee that ***ego alone does not a good hacker make***

so how did i become a pentester then?

- With Internet texts and a series of good mentors :)
  - The Rainbow Series, always a good place to start
  - “Smashing the Stack for Fun and Profit” by Aleph One
  - “How to Become a Hacker” by ESR
  - IRC and underground websites
    (just take everything with a grain of salt)
  - understanding the *process* of an attack; not just the tools and the vulns… but the actual mindset one must achieve to circumvent

hacking training: the good, the bad, the ugly

- Early on (pre-2000), your choices were few, but the education was generally good
- *Good*, but **not great**
  - For the most part, we were teaching tools with a basic [prescribed] formula for using those tools to explore common network security deficiencies
  - But we weren't teaching a methodology, because:
    - It was difficult to teach someone to "think like a hacker" in only 5 days
    - *A good (and commonly accepted) methodology didn't yet exist*

hacking training continued

- Unfortunately, nowadays, there are a zillion companies who will teach you “applied hacking,” “penetration testing,” “ethical hacking,” and other such crap
- Few of them actually know what they’re doing
  - Most are “certified” but lack real experience.
  - They’ll teach you nmap and offer you 80 hours of “bootcamp-style” rhetoric, but they can’t teach you to be a good pentester.
    - (In fact, of the **dozen** or so “C|EH instructors” I’ve met, only **3** had ever actually performed a penetration test for hire. OMGWTF?)

hacking books

- Hacking Exposed. Good book, set the bar pretty high.
- Nonetheless, a million other “hacking books” followed, and as with “hacking training,” many (most) of them sucked.
- I have at least a dozen crappy books that are basically re-worked re-writes of each other… teaching the same old tools in the same old way, with the same old screenshots.
- A few notable exceptions: shellcoders handbook, hacking: the art of exploitation, grayhat hacking, google hacking for pentesters

hacking certifications

- No, seriously, are you really proud of that?
  - All certifications, given time, become worthless due to brain dumps and study guides.
- Assuming they weren't worthless to begin with.
  Does a tool-based class & tool-based cert really prove your skill-set?
- I posit that "certified hacker" is *almost* as good as a note from your mom (but not *quite*). Who exactly is really qualified to certify a hacker?
  - I've never seen a test, multiple choice or otherwise, that could even hope to identify a good hacker. Especially one with an 80% pass-rate at the conclusion of a 5-day class. Get real.

apologies

- Yeah, yeah, ok.
- I’m sorry to those of you who do actually know what you’re doing. You are the notable few, you’re smarter than your peers, you’re a dying breed, blah blah blah. (remember my disclaimer?)
- The rest of you know who you are. If your face turned beet red during that last slide, you’re probably one of the people who thinks that a “hacking instructor certification” makes you an expert. Do you seriously believe that crap?

on regurgitation

- i've heard "war stories" about pentests that ***i performed*** told by more than a handful of other "hacking instructors" (many of whom attended my classes) across the course of the past several years
- if i ever catch you using one of *my* stories, i can assure you that i will make every effort to ridicule and humiliate you, publicly ;)

“scan now” pentests

- from the “scan now” button in internet scanner
  - clients get a report with thousands of vulnerabilities with subjective risk ratings
  - does not account for the environment, network architecture or asset value
  - little guidance, no strategy, limited value
- ***many*** of the “pentests” currently being delivered are little more than “scan now” tests; they are ultimately in-depth vulnerability scans that produce thousands of pages of worthless results

bottom line:

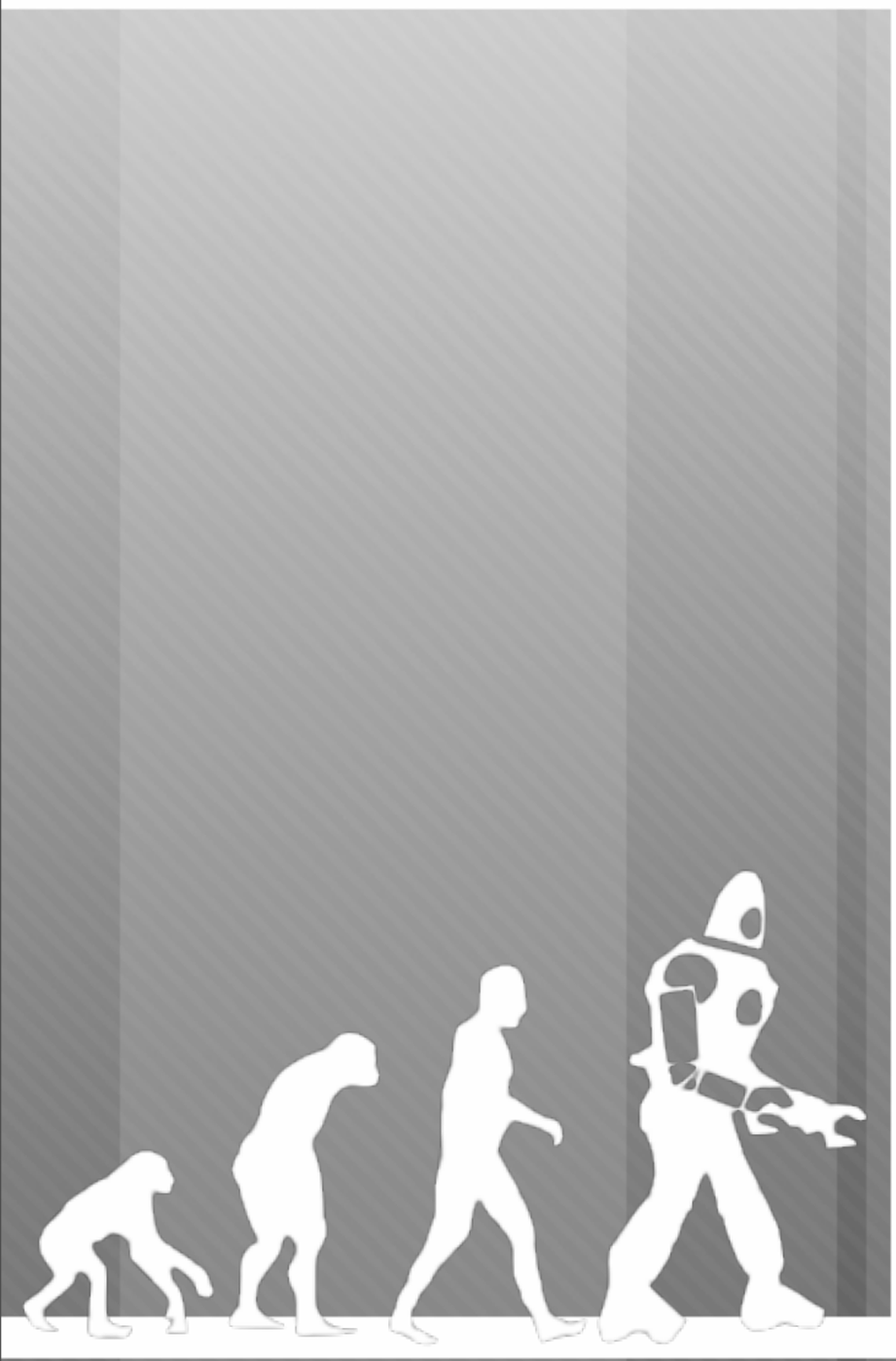
- it’s not about the tools!

another story. goody.

- pentest for a banker's bank (that's a bank that provides services only to other banks)
- external pentest was helpful, but not revelatory
- onsite pentest, however, revealed:
  - several oddly named accounts on an internal webserver; after two hours of password cracking, only a non-admin password was revealed. heartbroken, i continued on.
  - 20 minutes and about three guesses later, variations on my non-admin password gave me admin access to:
    - domain controllers, dns servers, core routers, and firewalls. game over

conclusion: why yesterday's pentest is worthless

- security is a process, not a project
  - lacking a methodology
  - no two tests are alike
  - early pentests were very adhoc
- pentesting goes mainstream
  - hacking in the movies
  - books, classes and certifications



Part 2

long live the pentest!

long live the pentest!

- the value of the pentest
- evolution of the pentest
- a framework for repeatable testing

2.1

the value of the pentest

where does pentesting fit?

- “penetration testing is a dead-end service line, as more and more tasks can be automated”
  - but is a pentest really just a series of tasks?
- “secure coding eliminates the need for pentesting”
  - pie in the sky?
  - if everyone were honest, there’d be no more crime
  - of course, this also overlooks many other more fundamental problems in the information security world

so pentesting isn't quite dead yet

- we say: "no, not yet"
  - current level of automation amounts to little more than automated vulnerability scanning
- as we said before, a pentest is much more than just a vulnscan!

remember that time…

- Client with AS400 and Windows

assessing the value of a modern-day pentest

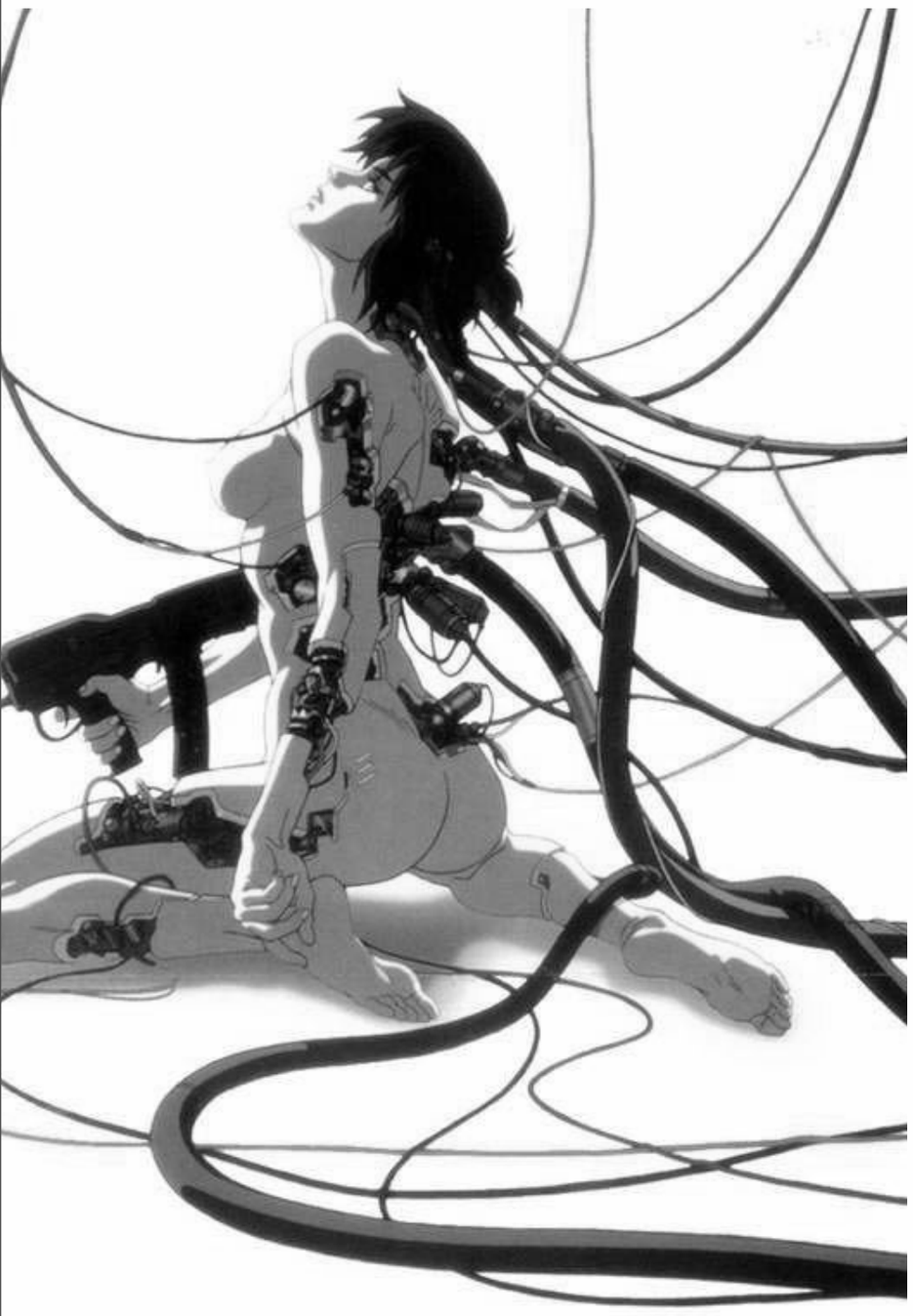
- is secure coding a realistic future?
  - the state of software flaws
  - the value of third-party review
    - oracle / litchfield paradigm
    - challenge issued, accepted and met
- not the only example - “pwn to own”

“we aren’t conducting a penetration test, we’re…”

- “...creating compelling events,” says marty sells (iss)
- it makes for a nice pop-quiz to see if current hacker tools and techniques can bypass deployed countermeasures
  - ofir arkin’s paper on bypassing NAC or using VLAN hopping to monitor “isolated” segments
  - recent research by Brad Antoniewicz and Josh Wright in wireless security expose problems in common implementations of WPA Enterprise
  - the point being, smart people can find unexpected/unforeseen issues that may not be common knowledge, so they would not be accounted for in any security initiatives
- pentesting might even improve awareness!

getting funding for infosec initiatives

- database tables for a slot machine operation
- doctors doing the Heisman pose



2.2

evolution of the pentest

what kind of things do we find today?

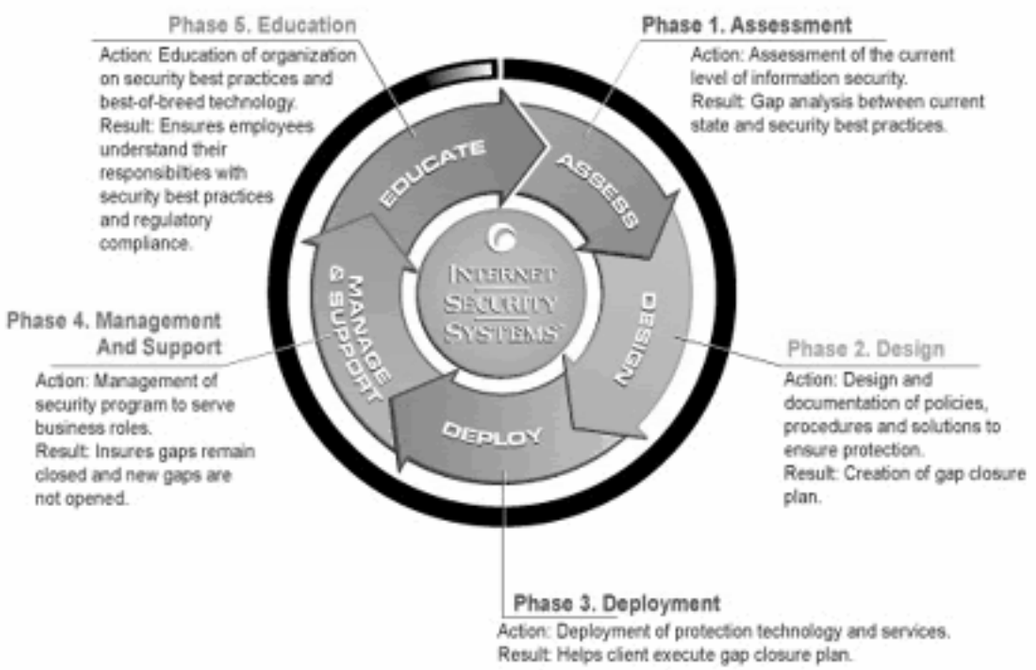
- weak passwords
- poor architecture
- missing patches
- system defaults
- poorly configured vendor devices
  - yep, we're talking about that printer/scanner/fax!

the funny thing is

- these are **the same damn things** we were finding 10+ years ago!
- **so have we really learned?**
  - is software measurably more secure?
  - is network architecture that much better?
  - has anybody listened to anything we've been saying?
- *(not a damn thing, apparently!)*

an ongoing process

- remember the iss addme model?
  - assess
  - design
  - deploy
  - manage
  - educate
  - (rinse and repeat)



a repeatable process!

- pentests of lore were often quite ad-hoc
  - unfortunately, with no continuity between tests, it's difficult if not impossible to effectively determine if things are improving
- believe it or not, ***process*** and (thank god there are no shmooballs at this con) ***metrics*** are actually quite important here

a systematic approach to security management

- ok, so let's compare:
  - yesterday's pentest:
    - "here's your 1300 page report from internet scanner^H^H^H^H^H, errr… that we custom generated, just for you!"
    - "risk profile? what do you mean?"

a systematic approach to security management

- current pentest
  - action plan matrix to deal with highest impact / lowest cost first
    - (still no accepted standard for determining risk profile improvements)
- systems that just count vulns don't take into account the # of vulns announced last week, last month, etc.
- we need an ever better system of metrics here

the metrics reloaded

- optimally, a good metric would account for
  - number of vulns discovered, over time
  - number of vulns by platform, over time
  - mean time for remediation
- and follow-up testing would ensure
  - follow-up pentest
  - ***assessment of effectiveness of deployed countermeasures***

invariably variable

- a pentest is still always influenced by the individual pentester's experience and background
- again, this reinforces the understanding that simple vuln counting is ineffective
  - for new findings across a systematic rescan
    - were these actual new findings? were they missed previously?
    - did the tools improve? was there a new team? did the team improve?

hammer time.

- 2006 pentest with “partial control”
- 2007 follow-up
- how complex are the metrics required to explain this situation?

upgrades to the toolbox

- nmap still reigns king (go see fyodor's talk!)
- superscanner
- john the ripper
- rainbow tables
- cain and abel
- metasploit, holy shit

upgrades to the toolbox

- vulnerability scan^H^H^H^H management
  - nessus
  - foundstone
  - iss
  - ncircle
  - tenable

upgrades to the toolbox

- wireless
  - high-powered pcmcia and usb cards (alfa!)
  - aircrack-ng
  - kismet, kismac
  - asleap
  - cowpatty (omgwtf, saw bregenzer's talk?)

upgrades to the toolbox

- live distros and other misc
  - backtrack (one pentest distro to rule them all)
  - damn vulnerable linux
  - winpe (haha, no just kidding, omg)

2.3

a framework for repeatable testing

improved methodologies

- isecom's **osstmm** now at v2.2, with 3.0 eminent (and available to paying subscribers)
- the open information systems security group is now proffering the **issaf**, the information systems security assessment framework
- kevin orrey (vulnerabilityassessment.co.uk) offers his **penetration testing framework v0.5**
- nist special publication **800-42** provides guidelines on network security testing
- wirelessdefence.org offers a **wireless penetration testing framework**, now part of kevin orrey's full pentesting framework, above

…forest for the trees

- early pentests were little more than exhaustive enumerations of all [known] vulnerabilities, occasionally with documentation on the process by which to most effectively exploit them
- with time, networks grew geometrically more complex, rendering mere vulnerability enumeration all but useless
- we now have to focus on architectural flaws and systemic issues in addition to vulnerability enumeration
- methodologies can be very helpful, but don't obviate the need for original thought. in other words, neither a cert nor a methodology can make you a good pentester if you don't already think like a hacker.

tactical vs strategic

- the [old] tactical approach
  - identify all vulnerabilities [known by your automated scanner], rate their risk as high, medium or low, then dump them into a client's lap and haul ass
- the [new] strategic approach
  - identify all known vulnerabilities, including architectural and conceptual, correlate them within the context of the company's risk (subject to available risk tolerance data) then assist in creating an action plan to calculate risk vs effort required to remediate

embrace the strategic

- strategic penetration testing therefore requires
  - a skilled individual or team with sufficient background (and a hacker-like *mindset*, not just a *certification*), capable of creatively interpreting and implementing a framework or methodology
- a scoring system that factors in things like
  - system criticality
  - complexity and/or likelihood of attack
  - complexity and/or effort involved in remediation
- **effective metrics!**

how providers are chosen

- i'll choose these guys if it's compliance and i don't want anything found,
- or… these other guys if i actually want to know what the hell is going on and don't want to get pwned later
- many companies also now have internal "tiger teams" for pentesting
  - while a good idea, third party validation is both important and necessary; remember our comments on different backgrounds and experience?



Part 2.4

pentesting in the 21st century…
and beyond

why we need an organic [open] methodology

- working with what we have
  - no point trying to reinvent the wheel
  - already have a methodology of your own? map, correlate and contribute it!
  - improvement of standardized methodologies only happens through contributions
- osstmm and issaf stand out as most complete
  - osstmm has been around longer, but both have wide body of contributors
  - moderate overlap, so review of both recommended

contributing to open methodologies

- osstmm and issaf will continue to improve
  - fueled by contributions
  - need continuous review
- difficult to measure the effectiveness of any one framework, but they can be evaluated against each other in terms of thoroughness and accuracy
- bottom line: *not* using a framework or methodology (at least in part) will almost certainly place you at a disadvantage

adapting to new technologies

- so how does one keep up with the ever changing threat / vulnerability landscape? what about wpa, nac, web2.0 and beyond? (which way did he go, george?)
- simple answer -- ***be dan kaminsky or billy hoffman***, or:
  - new technology does not necessarily imply old threats, vulnerabilities, attacks and solutions won't still work
  - want to pentest a new technology, but not sure where to begin, which tools to use?
  - do what smart developers do, threat/attack models!
    (see bruce scneier, windows snyder, adam shostack, et. al.)

can you test without a baseline?

- absolutely! (though you might have a hard time quantifying and/or measuring risks associated with discovered flaws)
  - then identify data flows, data stores, processes, interactors and trust boundaries
  - in other words, find the data, determine how the data is modified and by what/whom, figure out how and where the data extends and attack as many pieces of this puzzle as your existing beachhead allows!
  - if it's a piece of software running on a computer, it's ultimately vulnerable… somewhere

threat/attack modeling

- several different approaches, but all focus on the same basic set of tasks and objectives
  - msft says: identify security objectives, survey application, decompose application, identify, understand and categorize threats, identify vulnerabilities, [identify mitigation strategies, test]
  - wikipedia: identify [business objectives, user roles, data, use cases]; model [components, service roles, dependencies]; identify threats to cia; assign risk values; determine countermeasures
- although threat models are useful for securing software, at a more abstract level, they are also extremely useful for compromising new and/or untested technologies

quality assurance

- so can we define qa and/or qc in the context of penetration testing?
  - sure, it's basically an elaboration on our previously mentioned set of necessary / desired metrics
- # of vulns discovered over time, # discovered by platform, mean time for remediation and potential for mitigation by means of available countermeasures. further, apply richard bejtlich's five components used to judge a threat: existence, capability, history, intentions, and targeting
  - these metrics are then mapped back to assets against which individual vulnerabilities were identified and you have a quantifiable and quantitative analysis of a penetration test

hacker insurance?

- often dubbed “network risk insurance”
  - $5k - $30k/ year for $1m coverage
  - is it worth it? should you be recommending it?
    - well, that’s quite subjective. how good was your pentest? ;)
    - depends on the organization, the nature of the information they purvey, their potential for loss, etc. in general, i say absolutely!
    - providers include aig, lloyd’s of london / hiscox, chubb, zurich north america, insuretrust, arden financial, marsh, st. paul, tennant
    - unless you can “guarantee” your pentest by offering your client a money-back guarantee, suggesting hacker insurance might be a wise idea

Conclusions

- 1 the pentest is dead
- 2 long live the pentest
  - 2.3 a framework for repeatable testing
  - 2.4 pentesting in the 21st century and beyond

- Until next time...



End.

everything we said might be a lie

thanks for hearing us out,

-taylor and carric