Entrepreneurship for Hackers

“A thing or two I learnt while building PentesterLab”

Louis Nyffenegger <louis@pentesterlab.com>
@PentesterLab / @snyff

About me

Louis Nyffenegger

Security Engineer

Diploma in Computer Architecture from a French Engineering school (+ master degree in security) in 2006

Moved to Australia in 2009

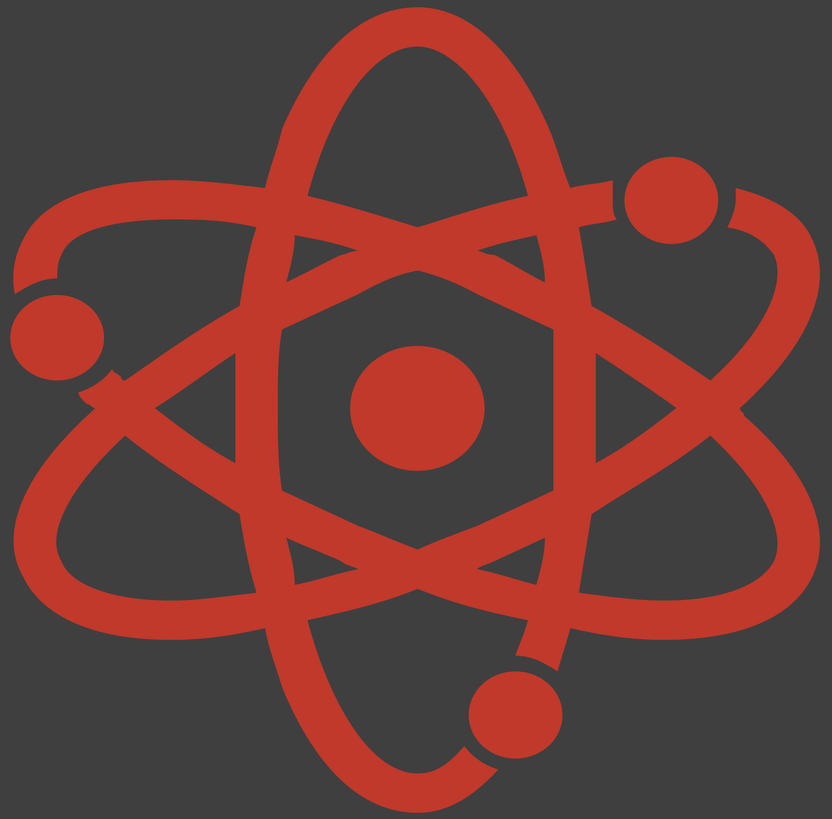
Security Consultant (2006)

-> Pentester (2009)

-> Code Reviewer (2012)

-> AppSec/DevSecOps (2014)

The security industry!



The Market for Lemons

Buyers are not necessarily knowledgeable (but it’s changing)

Hackers think they can rewrite your product in a week-end of hacking (or maybe 2…)

Funding for anything with Cyber in the name (double that if AI)…

Timing!

Very special time

Compute is cheap

Most products don’t cloudify well

Most products don’t dockerize well

Most products don’t CI/CD well

People who used to be (very) technical are now in leadership roles

Why this talk?

The media focuses on the wrong aspects:

- Funding rounds
- Exit from founders
- Revolutionary products

No one cares about a small founder that had a good year…

No one cares about products that just do solve an actually problem.

PentesterLab

December 2011

First version of the site
One exercise as ISO and course as PDF for sale $9.99

~ July 2012

Everything for free !
Around 6 exercises as ISO and course

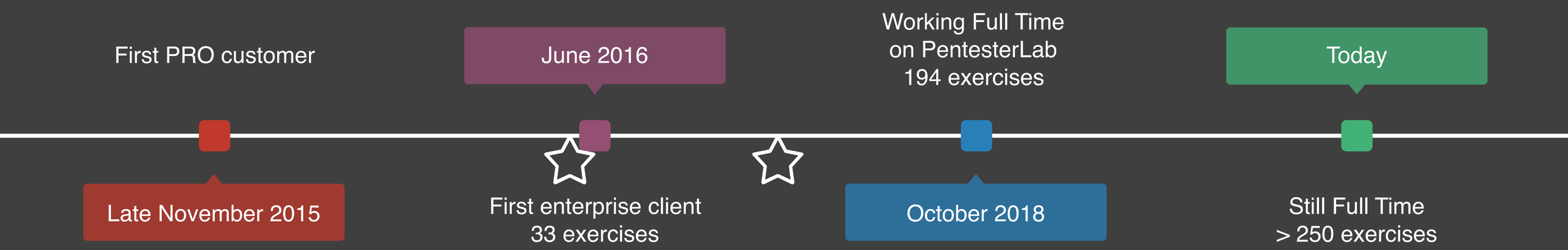
December 2014

Trying to build a PRO version…
and failing!

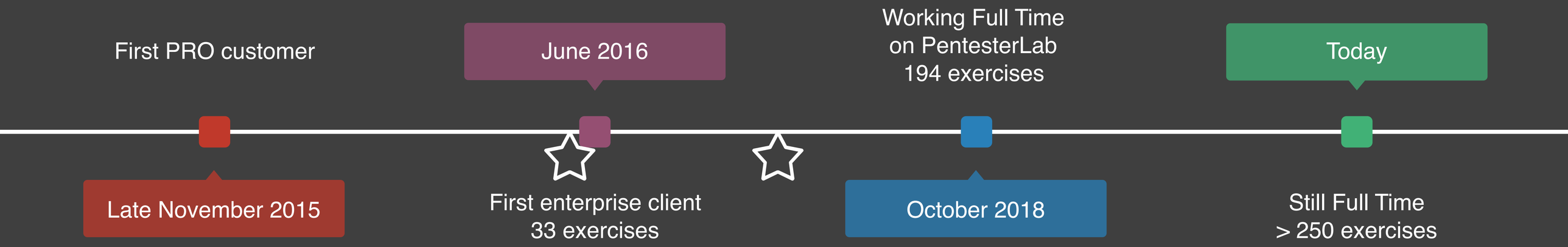
Novembre 2015

First release of PentesterLab PRO
$19.99 per month
5 PRO exercises
Soft launch

PentesterLab



PentesterLab



AN OVERNIGHT SUCCESS 🤔

The IDEA

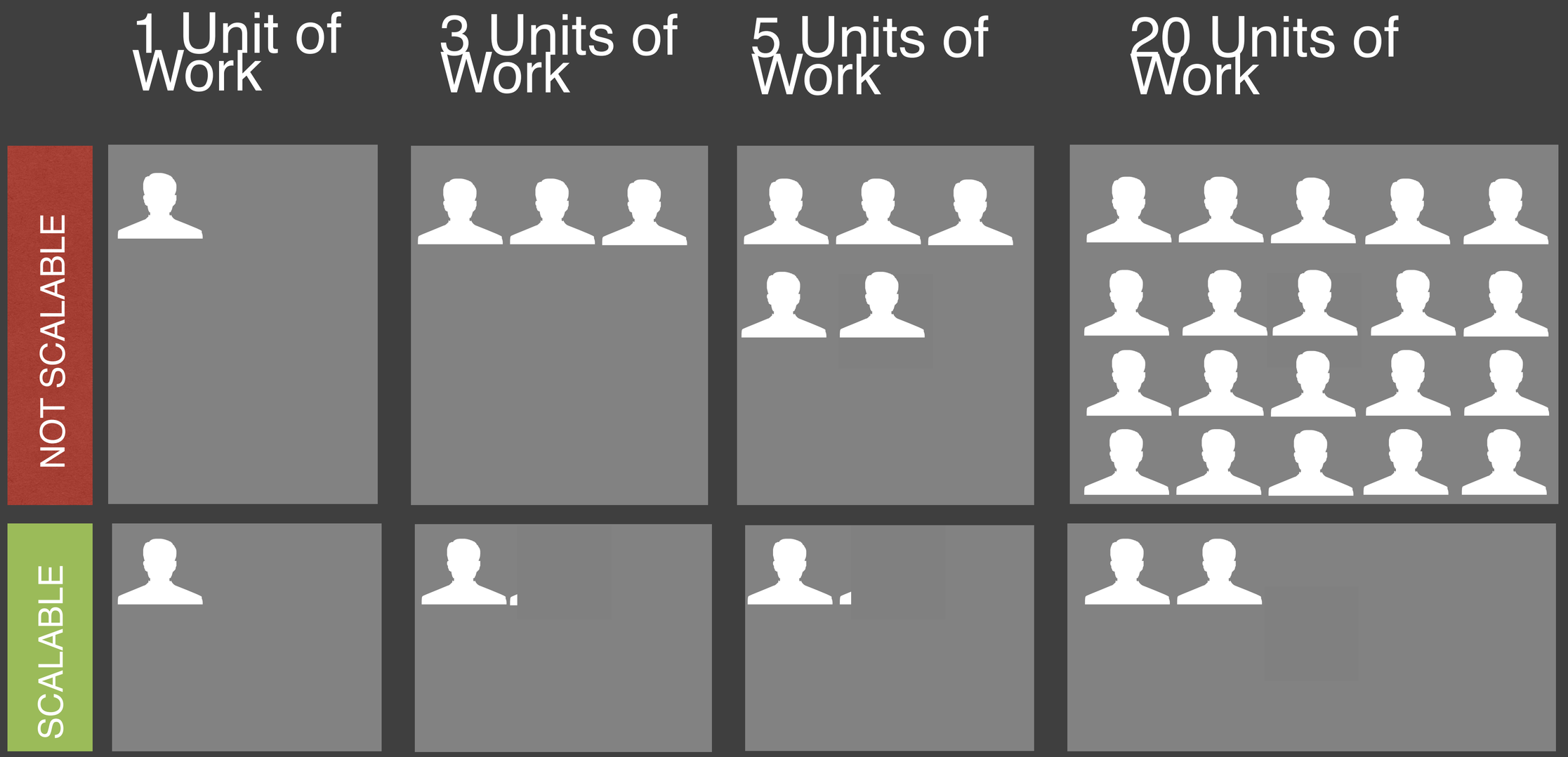
Something you love

Something that scales and compounds

Something you are good at

Something people are willing to pay for

Something scalable…



Something compoundable…

	After 1 month	After 3 months	After 1 year	After 3 years
! COMPOUNDABLE				
COUMPOUNDABLE				

The IDEA

- Don’t try to be everything
- Don’t try to be for everyone

- Solve a problem you have
- Solve a problem people have

- Try to leverage your unfair advantage/strengths

- Something ethical

- Something non critical (your customers’ SLA shouldn't rely on you)

- Pivot!

The IDEA: common misconceptions

Needs to be original

- Most likely someone thought about it before you did
- If no one thought about it, it is maybe because no one cares (aside from you)

Needs to be protected

- If you rely on the fact that your idea is not public, you will most likely have a big surprise once it is
- If you protect your idea, you don't get any feedback
- Anti-Reversing, Anti-X, … and spending more time protecting the idea than building it
- Software patent == EVIL

Needs to be huge

- You don't need to be the next Facebook/RSA/ Twitter
- You don't need a huge part of a huge cake
- It just needs to be sustainable

Needs to be cool

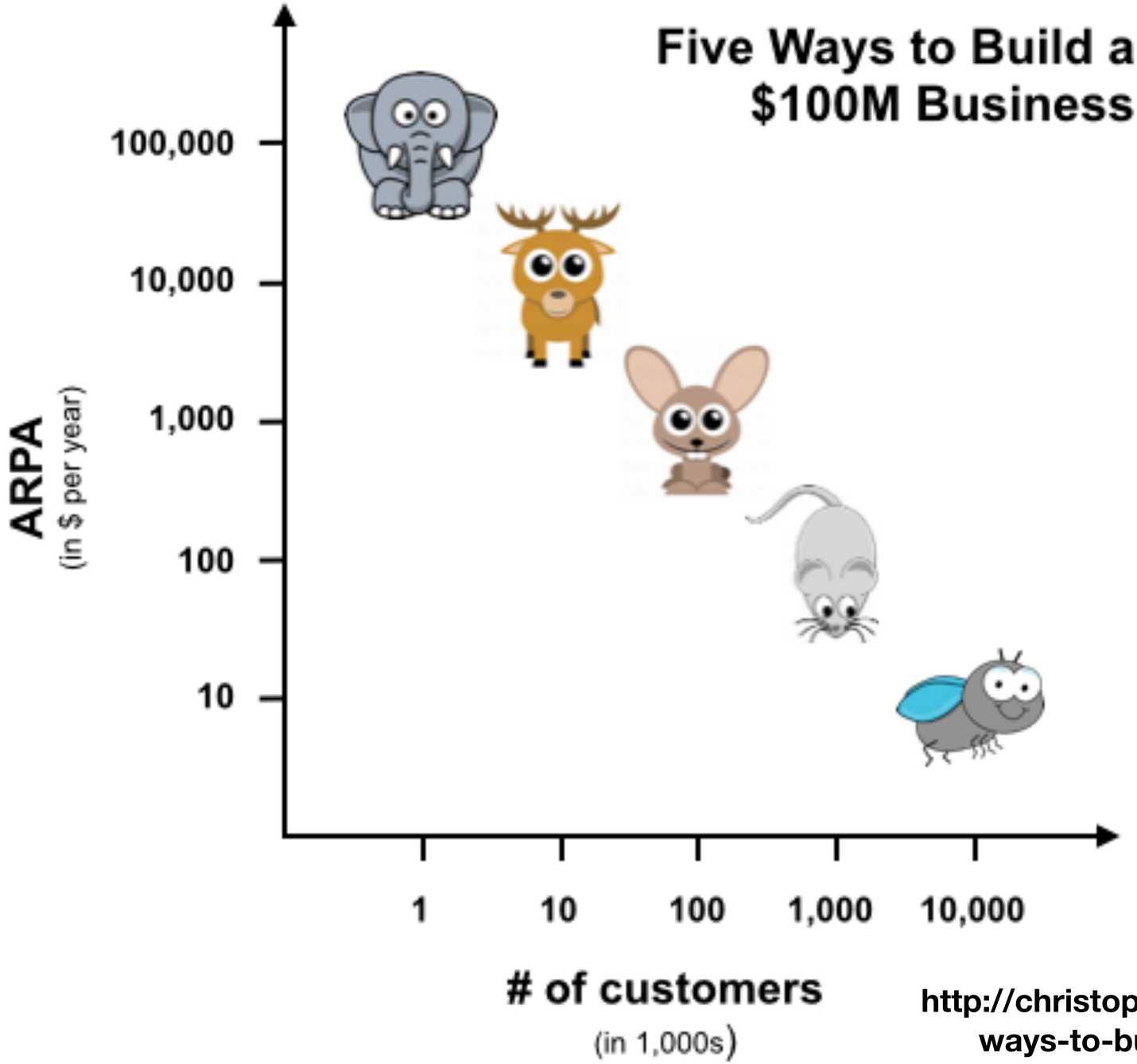
- Sometimes boring products solve real problems
- You can add the "cool" later on!

Bad Ideas

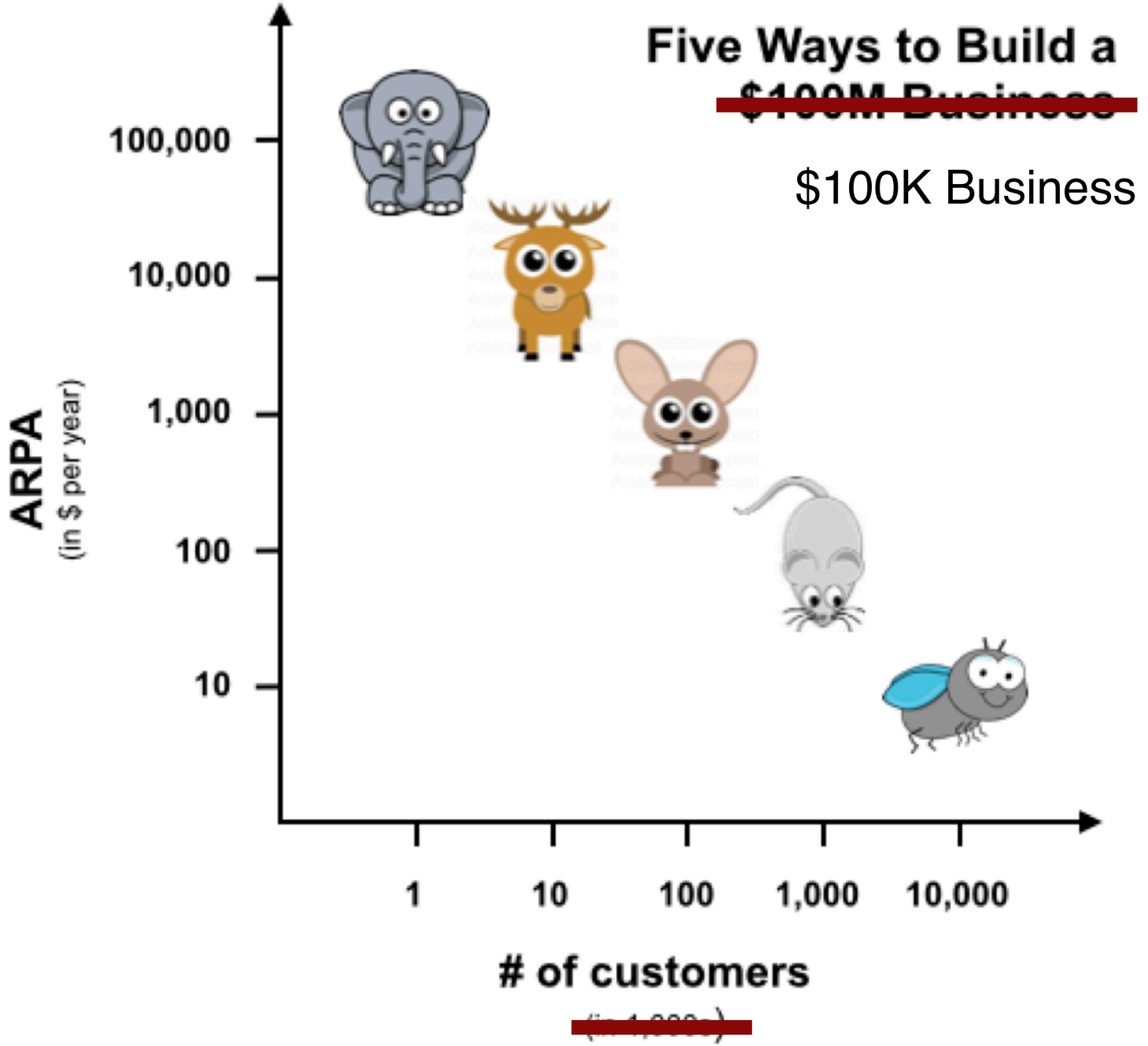
- AI for Blockchain Security
- Web Application Firewall based on AI
- Cloud based Vulnerability Detection
- Code review tool

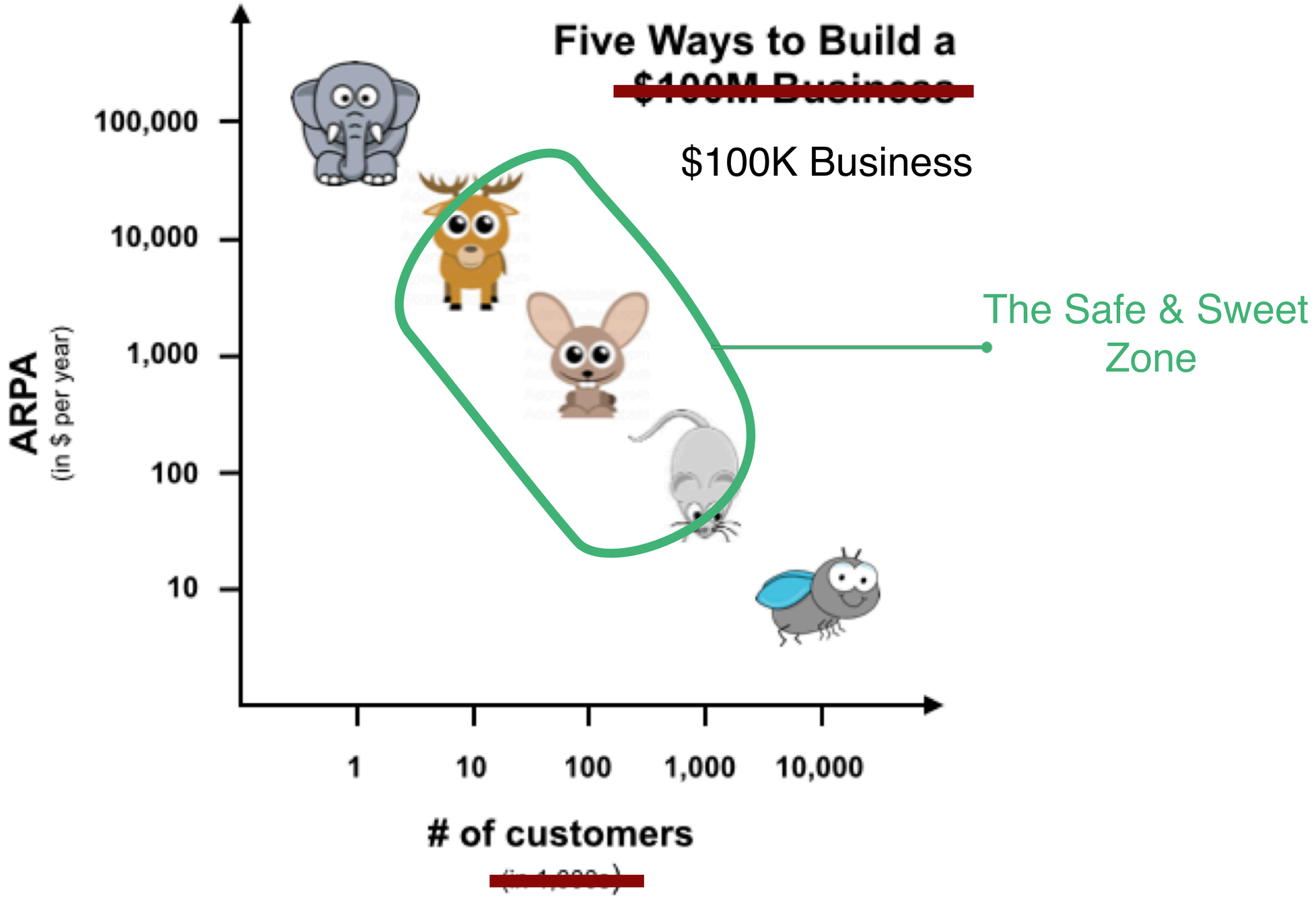
Better Ideas

- Fuzzers Integrated in CI/CD for firmware developers
- Vulnerability Assessment for docker images
- Web Scanner for QA testers
- Code review tool integrating in CI/CD for appsec teams
- Something that actually helps increase security



http://christophjanz.blogspot.com/2014/10/five-ways-to-build-100-million-business.html

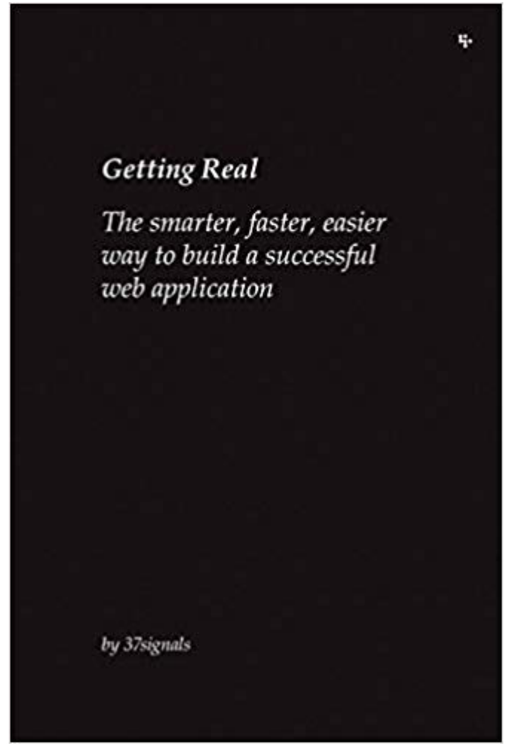

Five Ways to Build a
$100M Business
$100K Business
ARPA
(in $ per year)
100,000
10,000
1,000
100
10
1
10
100
1,000
10,000
# of customers


Five Ways to Build a
$100M Business
$100K Business
The Safe & Sweet Zone
ARPA
(in $ per year)
100,000
10,000
1,000
100
10
1
10
100
1,000
10,000
# of customers

Awful idea = -1
Weak idea = 1
So-so idea = 5
Good idea = 10
Great idea = 15
Brilliant idea = 20

No execution = $1
Weak execution = $1000
So-so execution = $10,000
Good execution = $100,000
Great execution = $1,000,000
Brilliant execution = $10,000,000

To make a business, you need to multiply the two.



Funding

Try to avoid external funding

Try to wait for as long as possible

If you do it, read carefully the terms (money is great, keeping control of your company is greater)

People funding business are not here for a nice&profitable business they want multipliers (10x, 100x)

Disclaimer: from my limited experience with funding

Avoid making a business with a free product

People are going to love you at first!

It’s all fine until you need to make money

Not sustainable

Ads…

Selling user’s data

It’s hard to get back to a paid model

Pricing

Features	Personal	Student	Corporate	Business	Platinum
	$ 10	$ 20	$ 35	$ 50	$ 75
Feature One	✓	✓	✓	✓	✓
Feature Two	✓	✓	✓	✓	✓
Feature Three	✗	✓	✓	✓	✓
Feature Four	✗	✗	✗	✓	✓
Feature Five	✗	✗	✓	✓	✓
Feature Six	✗	✗	✗	✗	✓

Disclaimer: I did terrible at this

Base your prices on the value you are bringing to your customers

Don’t base your prices on the time it takes you

Don’t base your prices on how much it costs you (time or $ amount)

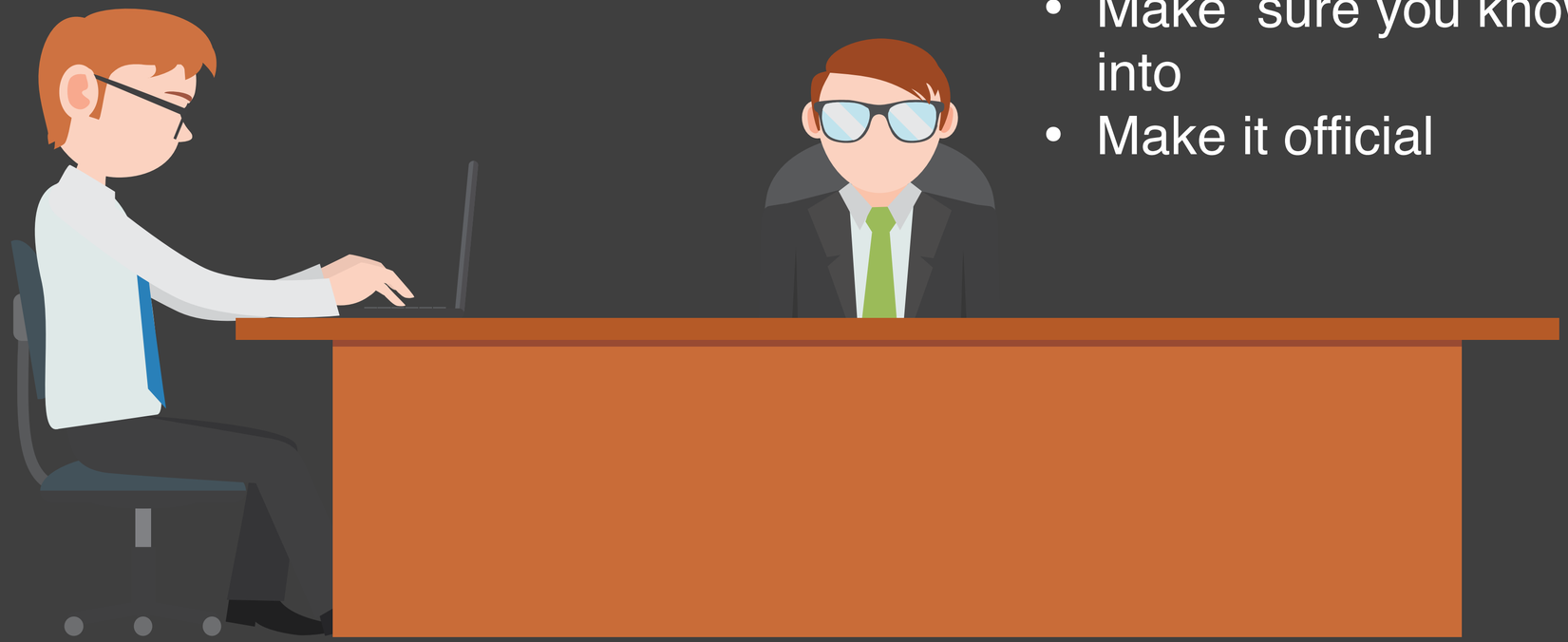
KISS: Keep It Simple Stupid!

The impact of low pricing…

Co-Founders?

Try to find one… it's hard

Try to get someone who is the opposite of you in term of background (avoid image below)



Try to find one… it gets lonely

Try to get someone who is the opposite of you in term of skills

It's kind of like dating/getting married:

- Make sure you know what you are getting into
- Make it official

Employees

Avoid hiring until you cannot anymore:

- Try to automate first (or avoid doing)
- Try to use freelancers
- Every person you hire will impact the ability of your company to survive

Try to get people who are the opposite of you in term of background (avoid image below)

Minimum Viable Product

Wikipedia:

“A minimum viable product (MVP) is a product with just enough features to satisfy early customers and provide feedback for future product development”

- Just enough features
- Satisfy early customers
- Provide feedback for future product development

Exercises available

CVE-2012-1823: PHP CGI

From SQL injection to shell

PHP Include And Post Exploitation

CVE-2012-2661: ActiveRecord SQL Injection

Introduction to Linux Host Review

Rack Cookies and Commands Injection

The Exercises

Our exercises are based on common vulnerabilities found in different systems. The issues are not emulated. We provide you real systems with real vulnerabilities.

Download the ISO and the PDF. Boot the ISO using any virtualisation software and **start learning!**

The number of ★ under the exercise indicates its difficulty.

★

★

★

★

★

★

★★

★★★

★

★★

★

★

Our exercises

BitTorrent Sync

Play XML Entities

Difficulty:

This exercise covers the exploitation of a XML entities in the Play framework.

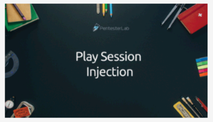
- Offline
- ISO (294MB)
- Java/Play

CVE-2014-6271/Shellshock

Difficulty:

This exercise covers the exploitation of a Bash vulnerability through a CGI.

- Offline
- ISO (19MB)
- CGI/Apache/Bash



Play Session Injection

Difficulty:

This exercise covers the exploitation of a session injection in the Play framework. This issue can be used to tamper with the content of the session while bypassing the signing mechanism

- Offline
- ISO (99MB)
- Java/Play

CVE-2007-1860: mod_jk double-decoding

Difficulty:

This exercise covers the exploitation of CVE-2007-1860. This vulnerability allows an attacker to gain access to unaccessible pages using crafted requests. This is a common trick that a lot of testers miss.

- Offline
- ISO (191MB)
- Tomcat/Apache

XSS and MySQL FILE

Difficulty:

This exercise explains how you can use a Cross-Site Scripting vulnerability to get access to an administrator's cookies. Then how you can use his/her session to gain access to the administration to find a SQL injection and gain code execution using it.

- Offline
- ISO (178MB)
- PHP/Apache/Mysql

Electronic Code Book

Difficulty:

This exercise explains how you can tamper with an encrypted cookies to access another user's account.

- Offline
- ISO (169MB)
- PHP/Apache

Our exercises

BitTorrent Sync

Cross-Origin Resource Sharing

PRO

Difficulty:

This exercise covers Cross-Origin Resource Sharing and how it can be used to bypass CSRF protection if misconfigured.

Offline
PHP/Apache/Mysql

API to Shell

PRO

Difficulty:

This exercise covers the exploitation of PHP type confusion to bypass a signature and the exploitation of unserialize.

Offline
PHP/Apache/Mysql

Pickle Code execution

PRO

Difficulty:

This exercise covers the exploitation of Python's pickle when used to deserialized untrusted data

Online & Offline
Python

Play XML Entities

Difficulty:

This exercise covers the exploitation of a XML entities in the Play framework.

Offline
ISO (294MB)
Java/Play

CVE-2014-6271/Shellshock

Difficulty:

This exercise covers the exploitation of a Bash vulnerability through a CGI.

Online & Offline
ISO (19MB)
CGI/Apache/Bash

Play Session Injection

Difficulty:

This exercise covers the exploitation of a session injection in the Play framework. This issue can be used to tamper with the content of the session while bypassing the signing mechanism

Offline
ISO (99MB)
Java/Play

Marketing

- Try to avoid ads
- Do things that don't scale
- Think of things that your customers like
- Do something unique
- Don't spend money on big events (RSA/…)

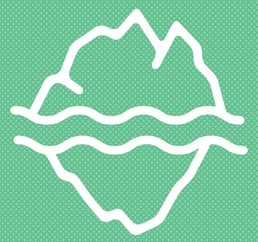
Services you can use

Things you will need

Sysadmin 101

Programming 101

Business 101

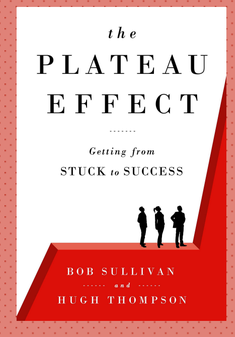


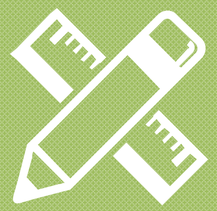
Final advice…

People on the Internet can be Jerks

BUT

“Don’t let the rare liar control your policy in every interaction and hijack good business decisions with paranoia”





Integrate a feedback loop in your product as soon as possible

Don’t underestimate your skills or how hard something may be for someone else

Don't worry about billing and issuing invoices… it's pretty easy (for hackers) and if you get it wrong people will tell/ help you

Stop making excuses and just do it!

Thank you!

Thanks for your time!

@snyff / @pentesterLab
louis@pentesterlab.com