Red Teaming

Finding your flaws before someone else does

Disclaimer

What follows are *our* opinions based upon *our* experiences. The opinions expressed herein are not necessarily those of our employer. Nor are they necessarily reperesntative of ISSA Portland or any affiliated organization.

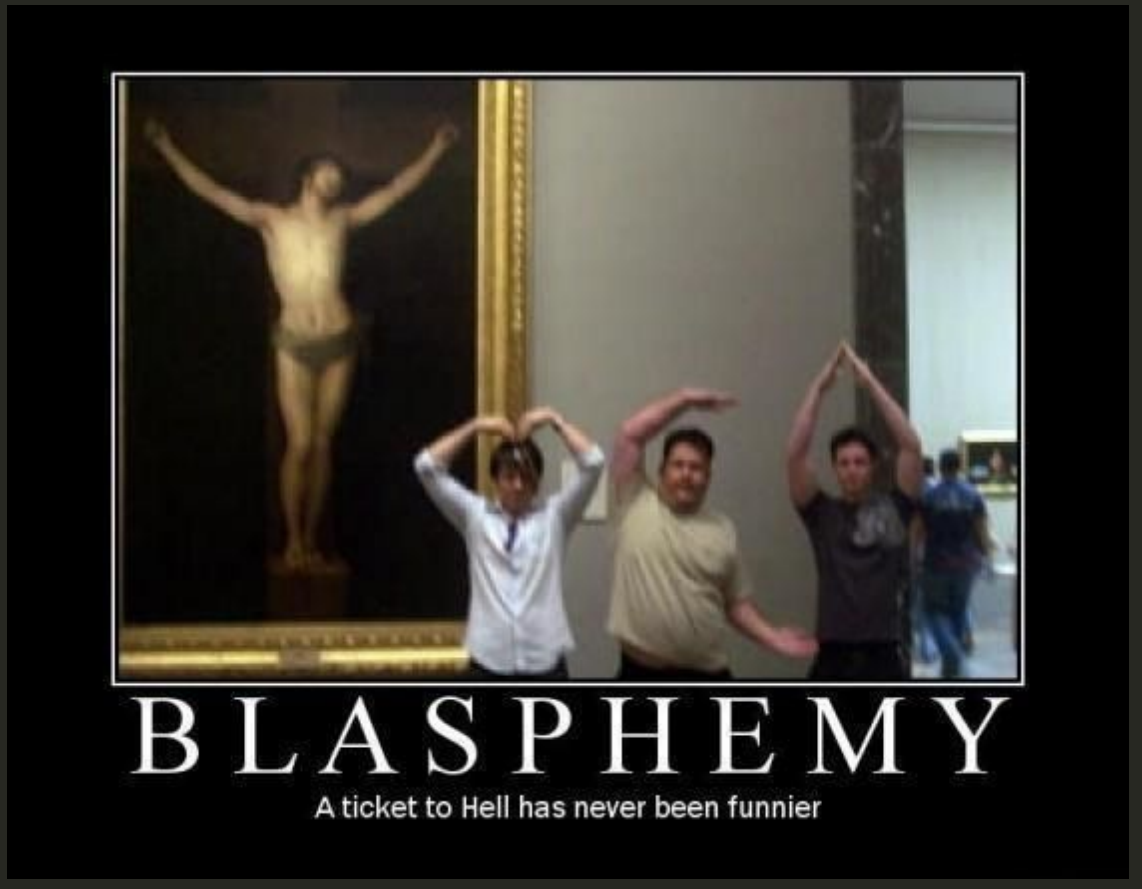
Trigger Warning

Cursing

Adult themes

Gratuitous use of gifs

Feel free to GTFO


BLASPHEMY
A ticket to Hell has never been funnier

Who are we?

Chris

- 5+ years in industry
- Focus on RE and Exploit Dev
- Certs: OSCP, OSCE, OSWP, OSEE (meta-cert: Offsec groupie)
- Bachelors of Sci in CS & Math, U of WA

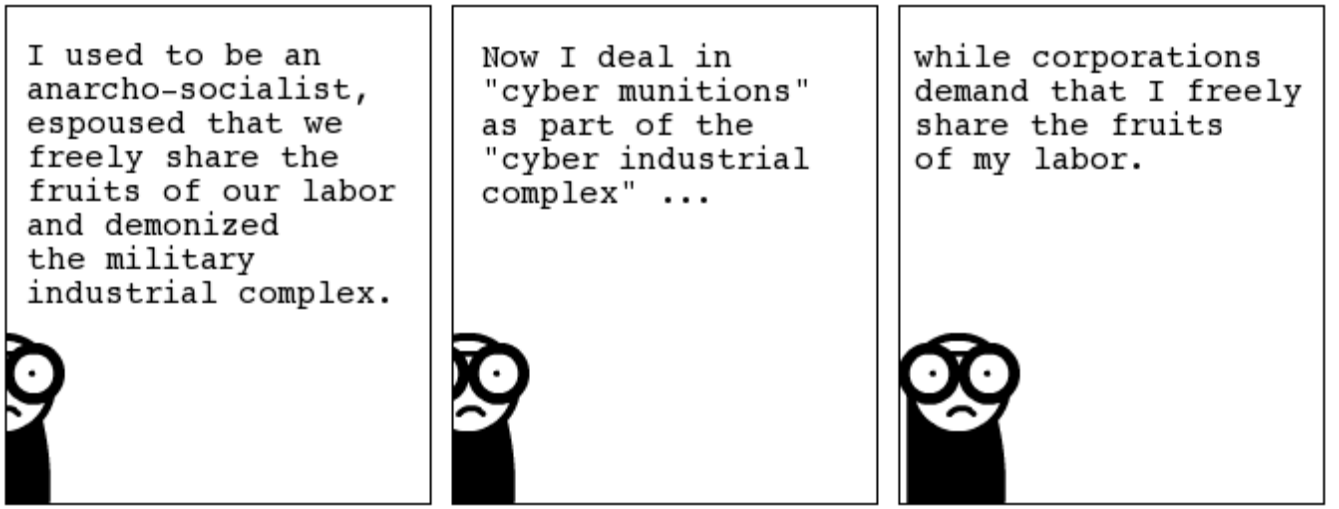
Robert

- 3+ years in industry
- Focus on Pen Testing
- Certs: OSCP
- Bachelors of Sci in CS, U of AK:FB

Who are you?

- Defense (blue team)?
- Background: IT / Eng / Other?

State of the industry



What is Red Teaming?

Broadly, any activities performed to test the security of an asset or the effectiveness of the security wrapped around that asset

- Vulnerability assessment: identifying known vulnerabilities in deployed infrastructure w/o exploitation (breadth emphasized)
- Penetration testing: identifying and exploiting known vulnerabilities, primary goal is penetration (depth emphasized)
- Vulnerability research: determining the exploitability of a vulnerability in custom code once found
- Exploit development: act of "weaponizing" a vulnerability
- To a much lesser degree static/dynamic code analysis: identifying vulnerabilities in custom code (src/binary must be accessible). This enters the realm of Prod Sec

Why have a Red Team?

Developers/SysAdmins are rather attached to their code/boxen

Red team has no such qualms

(actual video of Red Team at work)

Where does red teaming fit?

- Can collaborate with blue teams (to varrying degrees)
  - Poor-man's solution is "purple team": one team responsible for offense/defense
  - Slightly more expensive is two seperate teams
- Can be divorced from blue team
  - This makes sense in the context of a penetration test
  - Not so much the rest of the time
- Position wrt R&D vs. IT
  - Depends upon the culture of your company & goals
  - R&D: If you produce technology, red teaming needs to be engineering-oriented and working on abusing what your devs are producing.
  - IT: If you utilize someone else's technology, red teaming needs to understand how it is consumed and what it can do.

Product Security vs. Operational Security


COOL STORY BRO
TELL IT AGAIN
quickmeme.com

How to implement?

In-house or out-source

Depends on your resources, culture, & goals

In-house

- Need routine security work
- Have resources to maintain full-time team
- Your assets are *unique*, environment has a high-bar for access, or your IP is especially sensitive
- Conflict of interest: don't want to offend co-workers

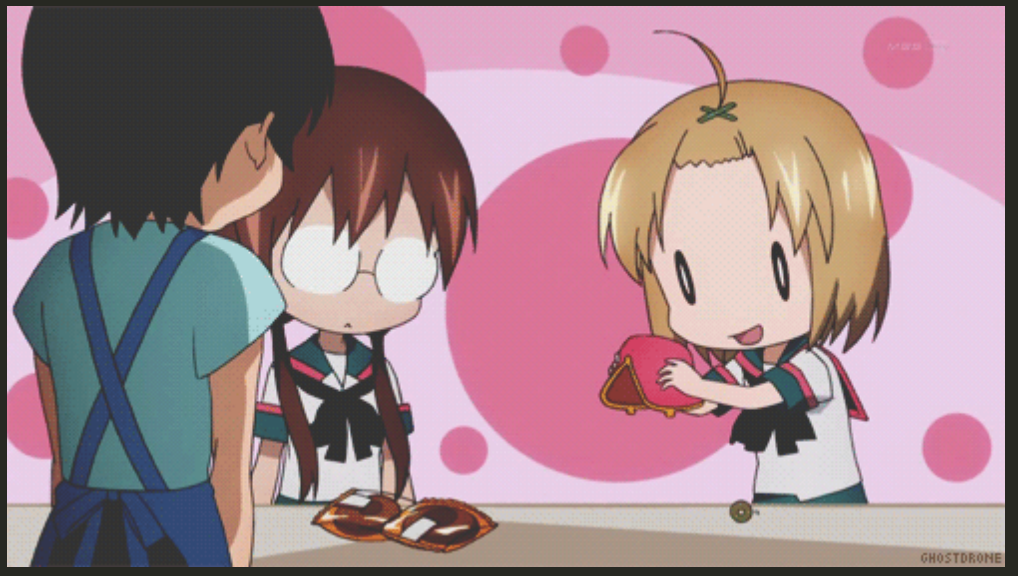
Out-source

- When you only need assessment rarely
- Have resources to burn
- Your assets are reltively routine
- Conflict of interest: they are your client and want to string you along as much as possible

Hooman/Tool Up

More pertinent to In-house team, but also relevant for outsourced

- Hiring
  - Here be dragons: insane amount of snake oil salesmen
  - If your role is technical: keeping your interviews technical will force frauds to sweat bullets
- Outsourcing
  - Scope the market, ask peers at other companies about their experiences
  - Options range from the crazy expensive to the dirt cheap; emulate something between Nation State and bored teenagers
- Tooling
  - In *almost* all cases, regardless of the type of tool, there will be commercial (read: expensive) and open-source solutions
  - In *almost* all cases, the open source solutions are more than adequate for most purposes
  - That being said, there are tools that will be worth paying for. I encourage you to consider these and know when it's worth dropping the money.



Security is expensive

Still cheaper than getting popped

Fancy new Red Team

Wat do?

Anarchy! Your job is to break rules

There are lots of areas

Network Security

Application Security

Mobile Security

Web Application Security

Incident Response

Checkbox Checking -- er, Compliance

Let's dive in!

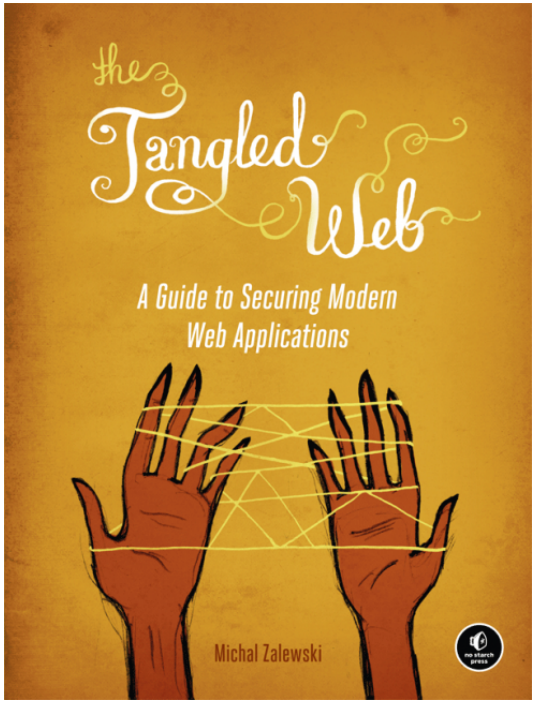
Web Application Security

I think this internet thing might just catch on...

- Relatively low barrier to entry
  - Straight-forward to do with open-source tools
  - Readily available research
  - Can often be done with *just* a browser
- Current market hotness
  - Paired with mobile security
  - Trend towards SaaS/subscription-model where valuables are server-side

Complexity Tradeoff

- Traditional AppSec
  - Focuses on Code Execution & Priv Esc
  - Requires deeper system knowledge
- Web Sec is a Tangled Web(c) of resources
  - Server resources:
    - Some sort of App stack (Java,Python,Ruby,PHP)
    - Database
    - Invariably other sources of compute
  - Client resources:
    - Cookies, Passwords, Compute



"The Tangled Web". Michal Zalewski (No Starch Press)

Attack Surface Everywhere

Demo time

Behold: our Demo App

- Simple CMS/Blog app
- Built on Flask(Python)
- Sqlite database

Step 1: Reconaissance

Step 2: ????

Step 2: ????

(Connect some dots)

Step 3: Profit! (Pwnage)

Bugs

- XSS via comments
- SQLi from admin panel
- Directory traversal via file upload

The directory traversal bug (using an SSH public key) can allow for shell access.