Security Testing

An Overview

whoami

- ✪ Current
  - ✭ Penetration Tester
  - ✭ Team Lead
- ✪ Experience
  - ✭ >8 years Linux System Engineer
  - ✭ 1½ years Information Security Management
- ✪ Hobbies
  - ✭ Bouldering & hacking

Agenda

1. Security Assessment
2. Vulnerability Assessment
3. Penetration Test

Security Asssessment

Goal

Improve Security Posture

How and what?

Methodology

- Paper exercise

Scope

- Processes and People
- Systems, Organizations

How long, how often?

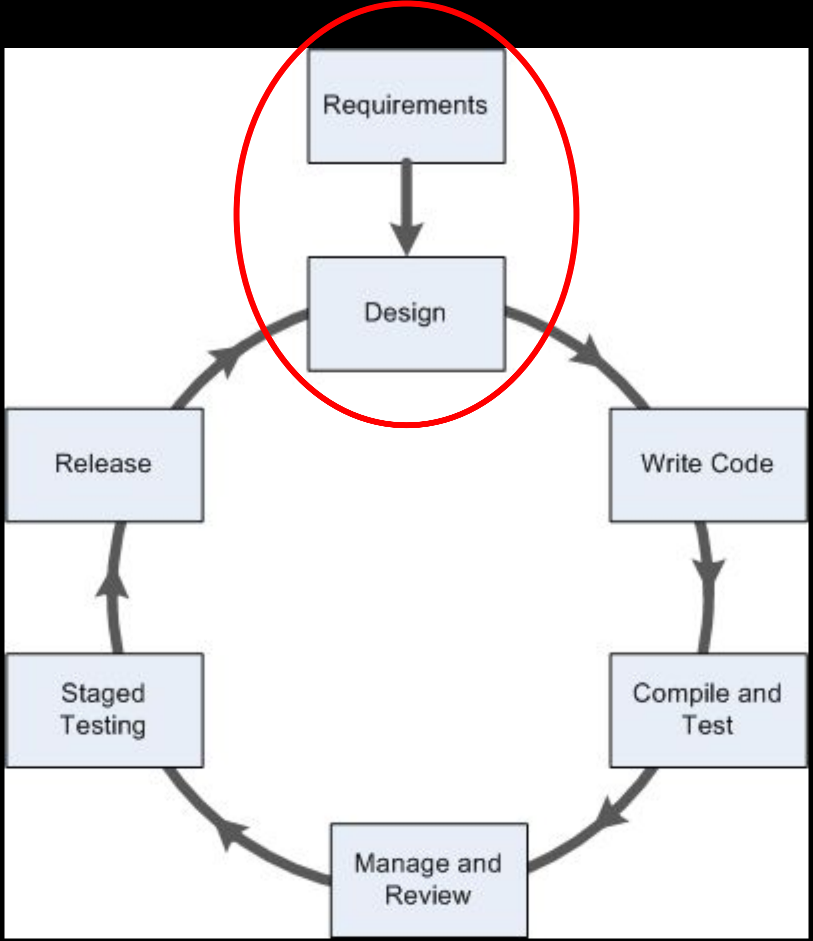
Duration

- Hours to days

Repetition

- Yearly or before major changes

SDLC



Requirements
Design
Write Code
Compile and Test
Manage and Review
Staged Testing
Release

Difference Audit – Assessment

Audit

- Singular event
- Always third parties
- Every few years
- Compliance w/ standards and best practices

Vulnerability Assessment

Goal

Identify and classify vulnerabilities

How and what?

Methodology

- Automated scanning

Scope

- Technology
- Applications, systems, organizations

How long how often?

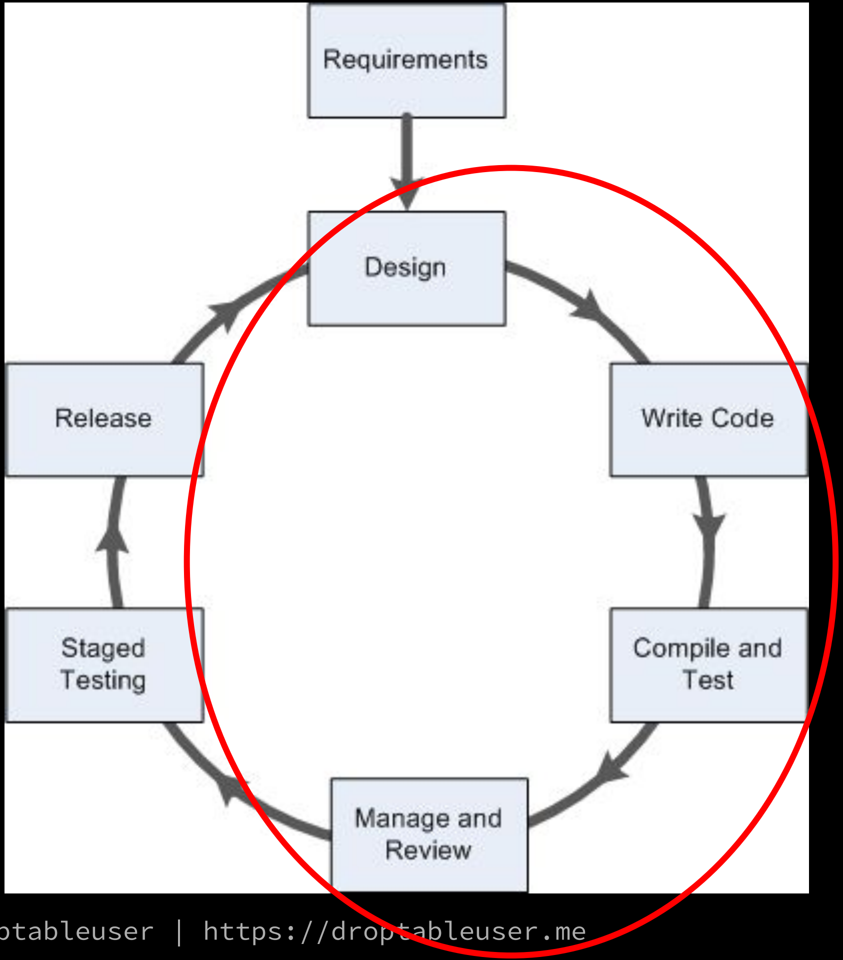
Duration

- Hours to days

Repetition

- Quarterly or after major changes

SDLC


Requirements
Design
Write Code
Compile and Test
Manage and Review
Staged Testing
Release

Tools

Semi automated scanners

- Network
- Application
- Source Code

Network Scanners

- Nmap (https://nmap.org)
- OpenVAS (http://www.openvas.org/)
- Nessus (https://www.tenable.com/downloads/nessus)

Application Scanners

- OWASP Zap (https://github.com/zaproxy/zaproxy)
- SQLmap (http://sqlmap.org/)
- BurpSuite (https://portswigger.net/burp)

Source Code Scanners

- Myriad of tools
  - Static
    - Style
    - Conventions
    - Standards
  - Dynamic
    - Logic bugs

Static - Benefits

- Output understandable for developers
- Scales well
- Integrated in IDE

Dynamic - Benefits

- Temporal information
- Runtime checks

Static - Drawbacks

- Can’t find configuration issues
- False-positives
- Hard to proof

Dynamic - Drawbacks

- Coverage difficult

Penetration Testing

Goal

Identify and exploit vulnerabilities while evading counter measures

How and what?

Methodology

- Automated scanning & manual exploitation

Scope

- Technology
- Applications, systems, organizations

How long, how often?

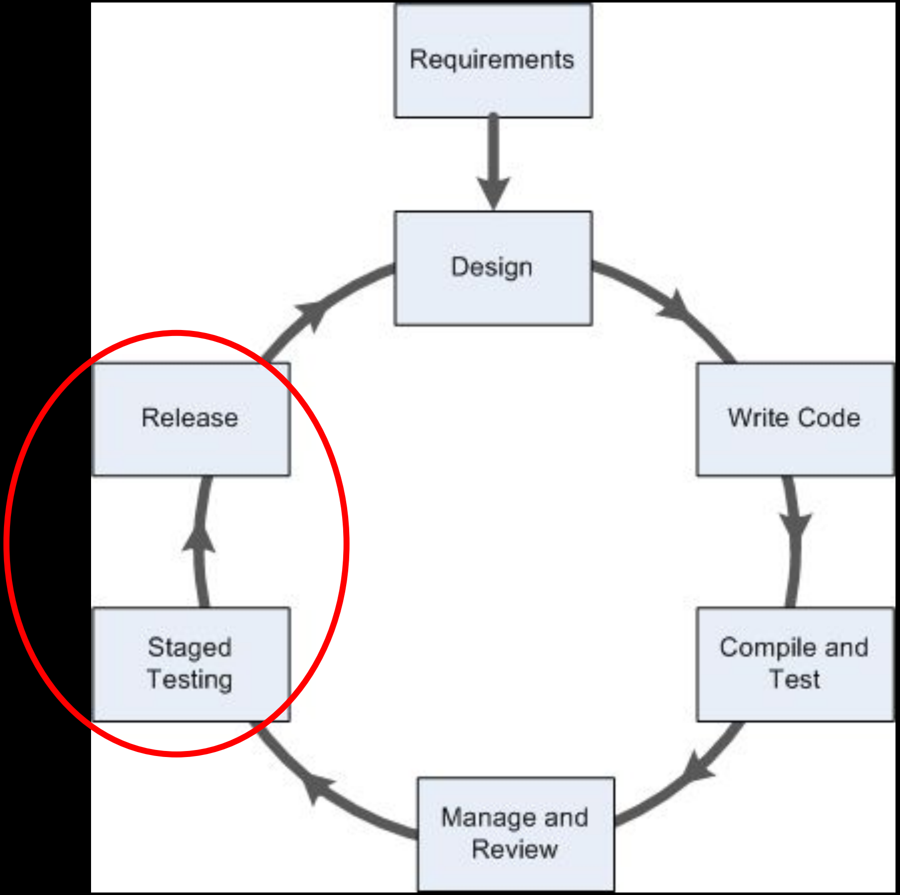
Duration

- Days to weeks

Repetition

- Yearly or after major changes

SDLC


Requirements
Design
Write Code
Compile and Test
Manage and Review
Staged Testing
Release

Phases of a Pentest

1. Pre-engagement
2. Intelligence Gathering
3. Threat Modeling
4. Vulnerability Analysis
5. Exploitation
6. Post Exploitation
7. Reporting

Pre-Engagement

- Permission to Attack
- Rules of Engagement
- Communication
- Contract
- Type of Penetration Test
- 3rd Parties

Tools

Word. Microsoft Word.

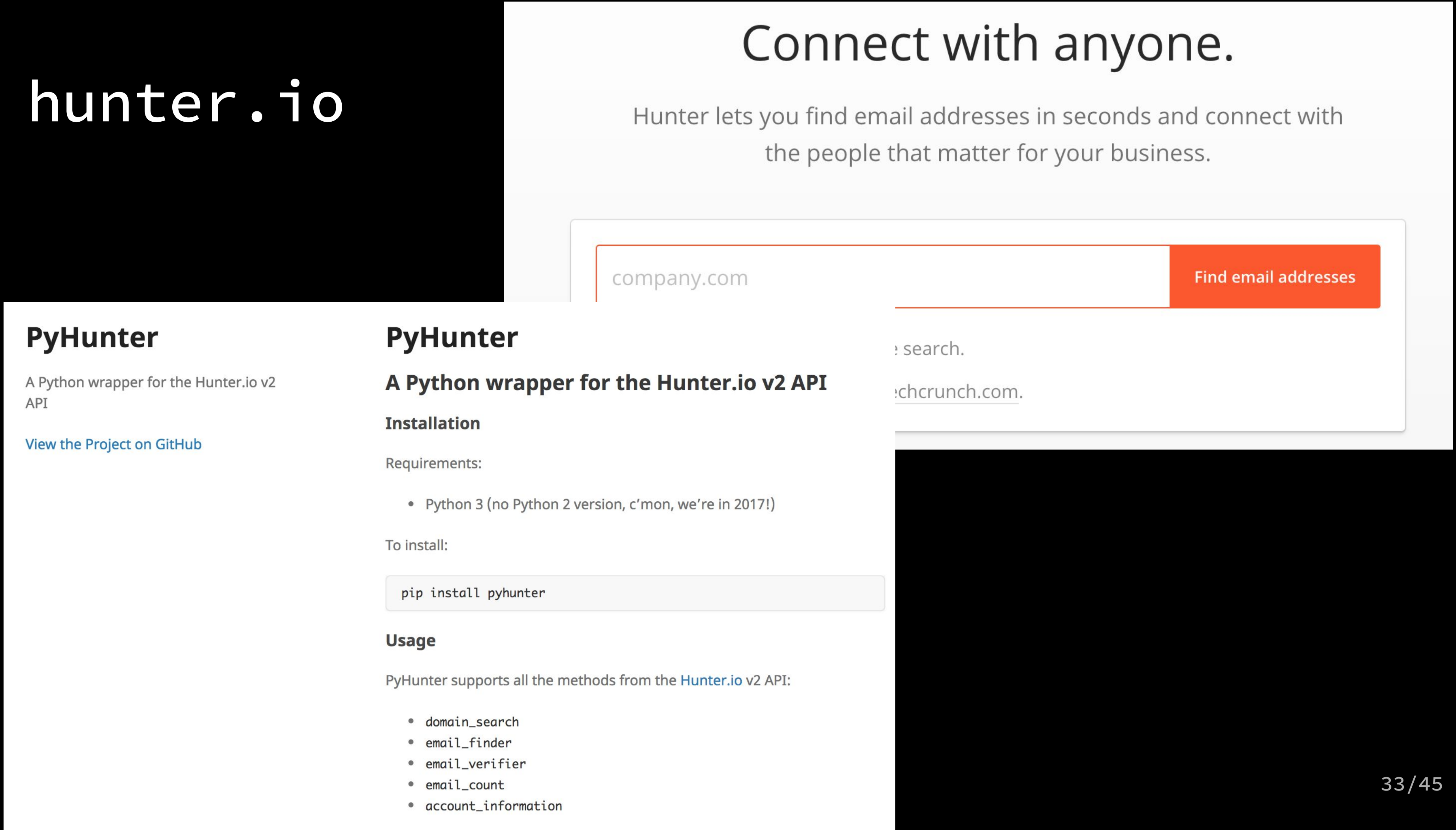
Intelligence Gathering

- OSINT
- Footprinting
- HUMINT

Tools

- https://github.com/digininja/CloudStorageFinder
- https://punk.sh/#/
- https://github.com/smicallef/spiderfoot

hunter.io



Connect with anyone.

Hunter lets you find email addresses in seconds and connect with the people that matter for your business.

company.com

: search.

:chcrunch.com.

PyHunter

A Python wrapper for the Hunter.io v2 API

View the Project on GitHub

PyHunter

A Python wrapper for the Hunter.io v2 API

Installation

Requirements:

- Python 3 (no Python 2 version, c'mon, we're in 2017!)

To install:

```
pip install pyhunter
```

Usage

PyHunter supports all the methods from the Hunter.io v2 API:

- domain_search
- email_finder
- email_verifier
- email_count
- account_information

Recon-ng

flickr
SHODAN
censys
Security driven by data
Google
Bing
GitHub

Threat Modeling

- Examine relevant data
- Identify assets
- Map assets/threats

Vulnerability Analysis

- Network Scanners
- General Vulnerability Scanners
- Traffic Monitoring
- Metadata Analysis

Tools

- Nmap scripts
  - `nmap --script smb-vuln*`
  - `ls /usr/share/nmap/scripts`
- Wireshark (https://www.wireshark.org/)
- OpenVAS
- Nikto (https://cirt.net/Nikto2)
- wp_scan (https://wpscan.org/)
- OWASP ZAP (prev. Dirbuster)
- Gobuster (https://github.com/OJ/gobuster)
- …

Exploitation

- Get initial foothold
- Circumvent security measure
- precision

Tools

- Metasploit
- DIY

Post-Exploitation

- Rules of Engagement
  - Protect the client
  - Protect yourself
- Infrastructure Analysis
- Pillaging
- Data Exfiltration
- Persistence
- Further Penetration
- Cleanup

Tools

- nmap
- Metasploit
- DIY

Reporting

- Objectives, Methods, Results
- CVSS3 Scores

This is what you buy!

Executive Summary

- Background
- Posture
- Risk Profile
- General Findings
- Recommendation/Roadmap

Technical Report

- Introduction
- Information gathered
- Vulnerabilities found
- Exploitations
- Risks
- Conclusion

Tools

- Dradis (https://dradisframework.com/ce/)
- Latex
- Most probably: Word. Again.

How to get started?

Bonus Slides

Books

- Penetration Testing - Georgia Weidman https://nostarch.com/pentesting
- The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws
- Black Hat Python - Justin Seitz https://nostarch.com/blackhatpython
- PoC||GTFO - Manul Laphroaig https://nostarch.com/gtfo
- …

Virtual Machines

https://github.com/Sliim/pentest-lab

https://github.com/bkimminich/juice-shop

More on:

https://www.abatchy.com/2017/02/oscp-like-vulnhub-vms

Wargames/Platforms

- http://OverTheWire.org
- http://hackthebox.eu
- https://www.wechall.net/active_sites

Writeups/Walkthroughs

- IPPSec’s Youtube Channel https://www.youtube.com/channel/UCa6eh7gCkpPo5XXUDfygQQA/playlists

How not to get started!

Wrong: An error means it didn’t work

Often an error is the result of a successful exploit.

Wrong: Spending too much time learning reversing/exploit writing instead of assessing systems, mobile and web

Though really, really awesome these spots are already filled usually. Mobile and web will get you the job.

Wrong: Reading a lot of security news without going in depth

Reproduce an exploit, or write one from the diff.

Wrong: Spending too much time building the perfect lab/laptop/…

Simply don’t.

Wrong: Not writing code/script

You should be able to code, to talk to software engineers as peers.