Kicking the Tires of Web Apps

Theory and Reality

whoami

- @alexkuhl → *Expert Human, Renaissance Man*
  - CBTS → Pen Testing and Security Assessment
  - Adjunct “Professor” (NKU) → Computer Science
  - Paper Traitors → Web development
  - alexkuhl.org → one day I'll redesign

Alex Kuhl: “horrid” and “troll-like”

Agenda

This agenda is purposefully left blank.

Warning

- Dense slides
- So fast your head will spin

Step 0

Scoping

Scoping

- **Goals?**
- Schedule?
- Cost?
- Systems?
- **Knowledge? → Black vs White vs Grey**
- **Style? → Automated vs Manual vs Hybrid**
- **What activities are allowed?**
- **Defense systems?**

Step 1

Recon

Recon

- Identify infrastructure of the web environment and company
- Important for both internal and external
  - Form strategy for attack(s)
- Commonly, and unwisely, skipped

Gather targets

- Client
  - Statement of work, documentation, interviews
  - Narrows scope, might limit natural attack vectors
  - Assumptions (ack!)
- Black box
  - Investigator uses minimal info
  - Slow, thus expensive
  - Might miss something (hey, I'm realistic)

Tools

- whois
- route server *fu*
- dig (nslookup)
- fierce
- dnsmap
- dnsrecon

Open Source Recon

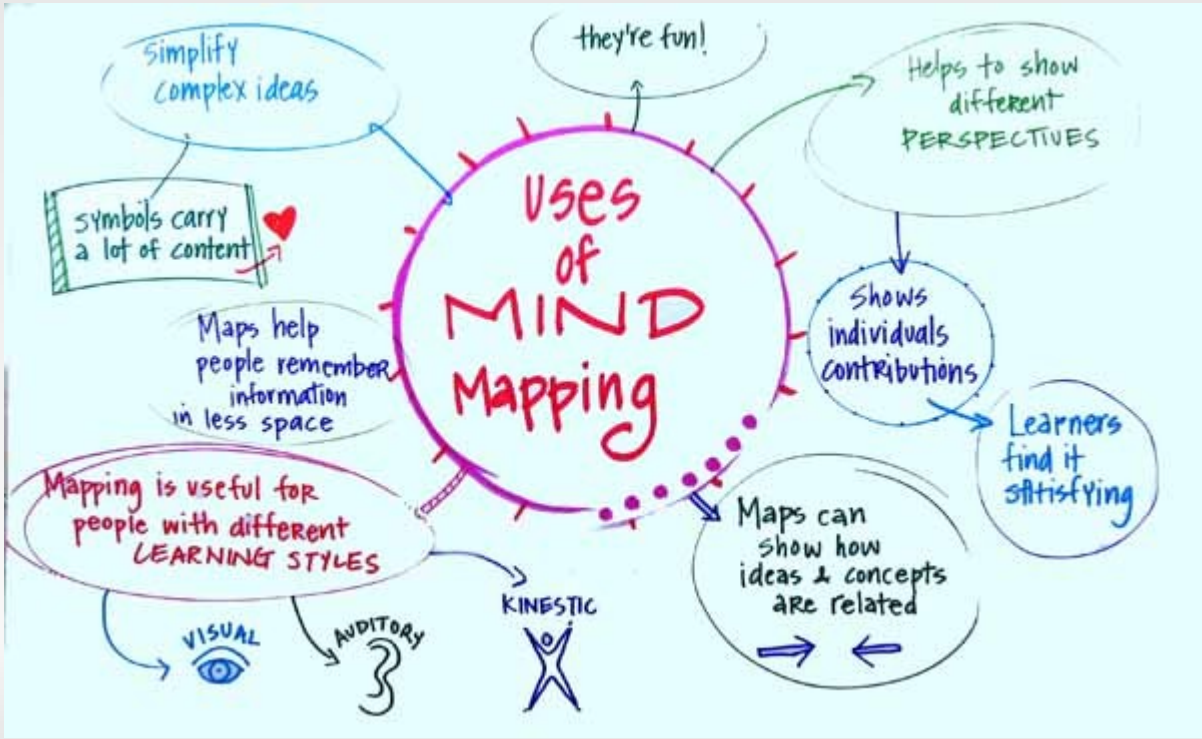
- Publicly-available information
  - Search engines
  - LinkedIn, Facebook, other social networks
  - Press releases and job posts
  - Newsgroups/bulletin boards, mailing lists, “help me” sites (StackOverflow family)
- Tools
  - Maltego
  - Google *fu* (or GHDB+illegal tools)
  - ***Manual***

Ideal

Loop until no changes:
    Talk to client about scope
    Reconnoiter
    Strategize

Step 2

Mapping
(Usually mixed with Recon)



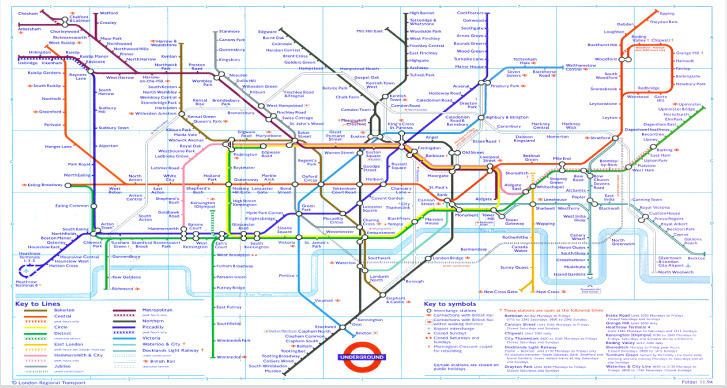
Mapping process

Iterate

Analyze

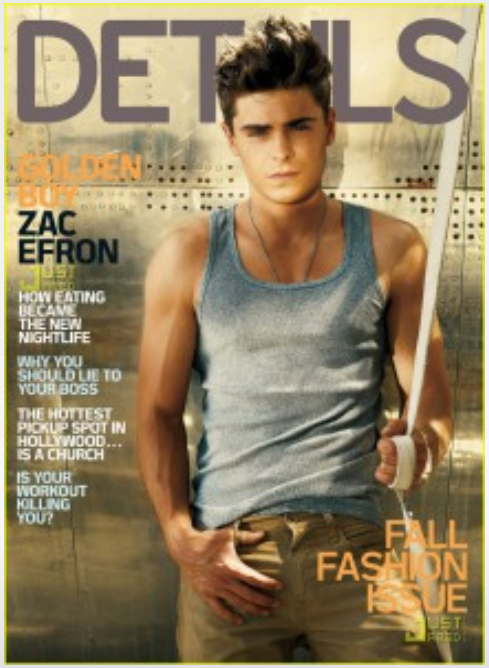
- Port Scan
- OS/Server Fingerprint & Version
- SSL
- Virtual Hosting & Load Balancer
- Software Configuration
- Spider

Full Analysis



Details & Tools

- Server Info – HTTP headers, `httprint`, `nmap`
- SSL
  - What is appropriate for the site? EV cert necessary?
  - `THC SSL Check` or `SSLDigger`
- Virtual Hosting
  - IP-based: Bing's 'ip:' modifier
  - Host-based: HTTP Host header
  - Alternative entry points (be careful)



Load Balancers

- Me no likey
- Confusion and false negatives
- Identification
  - Version/patch differences
  - URL analysis
  - HTTP Header: timestamp and last modification date
  - Cookies
  - SSL differences
  - HTML source code diff

by wishcake (flickr)

Details & Tools

- Software Configuration
  - Request methods supported
  - Default pages or PHP enabled
  - HTTP Headers and `nikto`
- Spidering
  - robots.txt
  - Manual + Automated → `burp` and `zap`

Analysis (Recon & Mapping)

- What targets are interesting?
  - Path of least resistance
- What infrastructure is in place?
- Known vulnerabilities?
- Commands supported by the technologies?

Step 3

Discovery

OWASP Top Ten (2010)

- **Injection**
- **Cross-site Scripting (XSS)**
- Broken Authentication and Session Management
- Insecure Direct Object References
- **Cross-site Request Forgery (CSRF)**
- *Security Misconfiguration*
- Insecure Cryptographic Storage
- Failure to restrict URL access
- Insufficient Transport Layer Protection
- Unvalidated Redirects and Forwards

Server-Side Discovery

- Automated
  - skipfish
  - WebSecurify
  - w3af
  - Burp**
- Manual
  - Injection, XSS, CSRF
  - **Business and Application Logic, Privilege Escalation**
  - Error messages
  - Burp + Web Developer Firefox extension

Client-Side Discovery

- Mostly manual
- Web Services – SOAP, wsdl, etc.
- Java - JAD
- Flash
  - crossdomain.xml, Flare
- AJAX(?)
  - Sprajax and ratproxy
  - If chaining ratproxy with another (like Burp) put it as the first proxy the traffic passes through

A Good Proxy Will Save Your Life

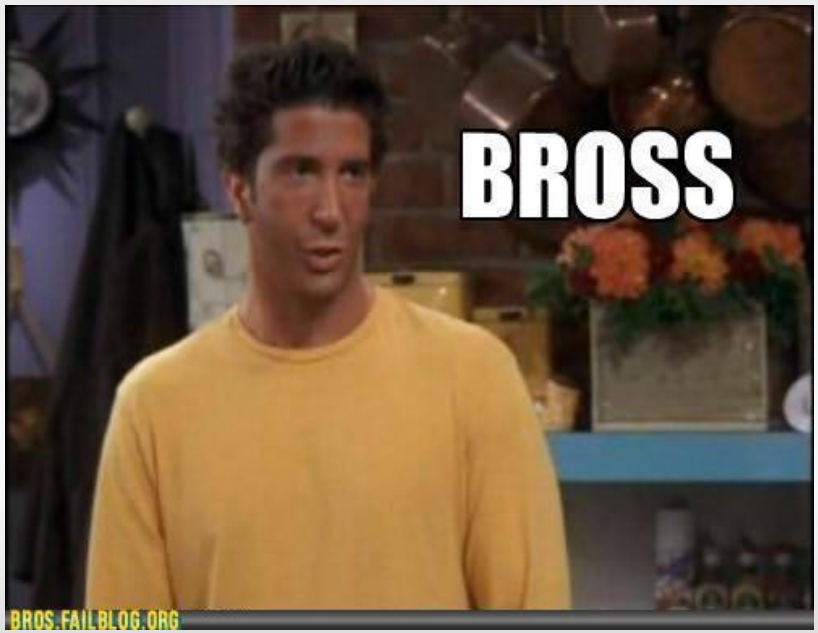
- Always verify automated results, proxy helps
- Tamper with requests
- See undisplayed data, especially in AJAX
  - Errors
  - JSON data
- Burp and ZAP

Step 4

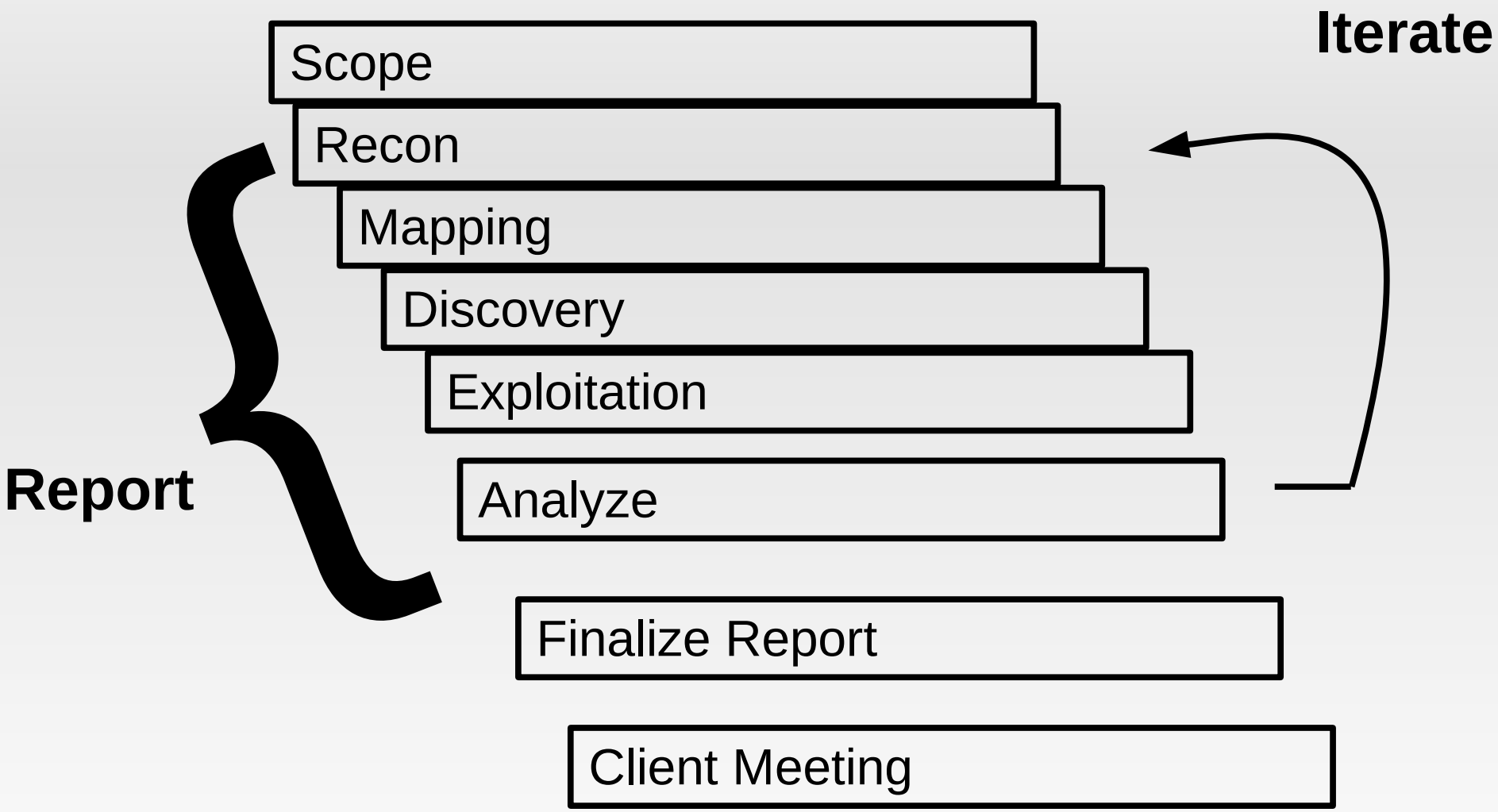
Exploitation

The Deeper Dive

- Fine line between Discovery and Exploit
- SQL → sqlmap
- XSS → durzosploit
  - Port scans, network maps
  - Zombies: BeEF, AttackAPI
  - Might need to limit targets
- CSRF → MonkeyFist



Framework


Iterate
Scope
Recon
Mapping
Discovery
Exploitation
Analyze
Report
Finalize Report
Client Meeting

Reality Check

That was all nice, but...

A real assessment

Scoping: Client

We want you to ***be*** a “hacker”

Scoping

Scoping: Client

No wait, that's not what I meant. Stay on [URL] and make sure to [ridiculously long list of contraints]. Oh, and you can do it in [1-3] days [during our short, early morning change window], right? Also, it's production, don't break anything.

Scoping

Sure, no problem.

Scoping

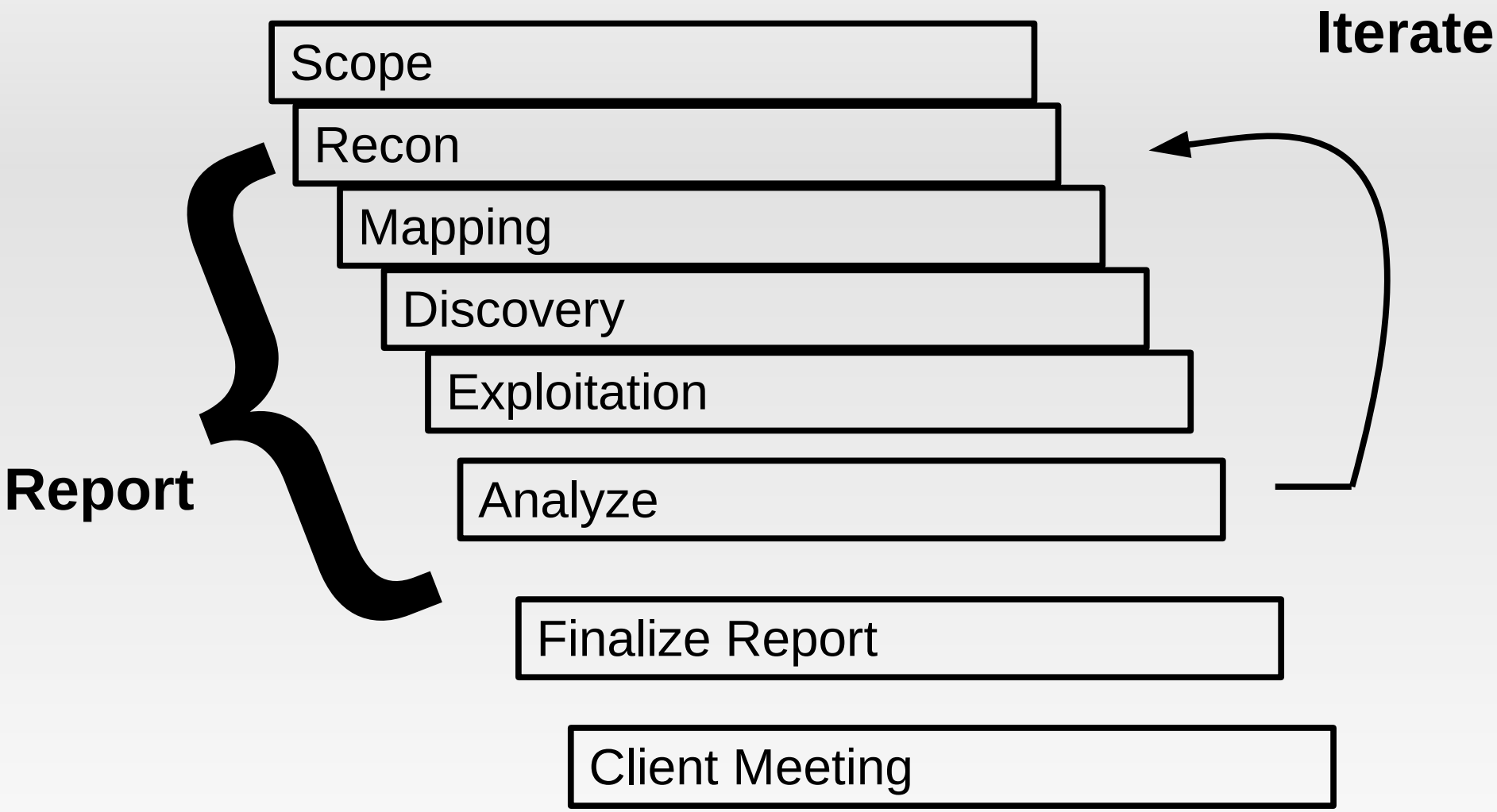
- Remember clients make *assumptions*?
- Will *forget* to tell you things
  - Critical things.
  - All the time.
  - Multiple times.
  - One after the other.
- Will try to get things for free

The Rest

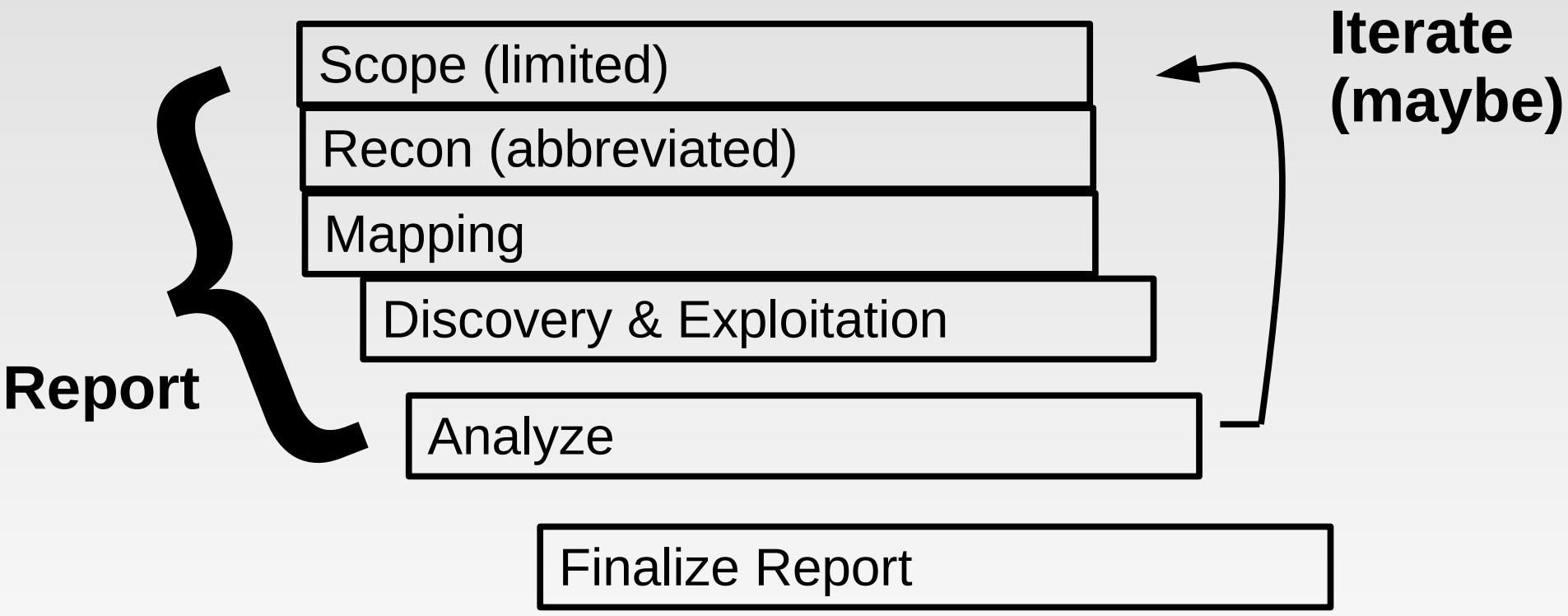
- Do your work
- Find some cool exploits

Reporting to the Client

Framework (Ideal from before)



Framework


Scope (limited)
Recon (abbreviated)
Mapping
Discovery & Exploitation
Analyze
Finalize Report
Report
Iterate
(maybe)