- [ ] Test for Reflected Cross Site Scripting
- [ ] Test for Stored Cross Site Scripting
- [ ] Test for DOM based Cross Site Scripting
- [ ] Test for Cross Site Flashing
- [ ] Test for HTML Injection
- [ ] Test for SQL Injection
- [ ] Test for SOQL Injection
- [ ] Test for LDAP Injection
- [ ] Test for ORM Injection
- [ ] Test for XML Injection
- [ ] Test for XXE Injection
- [ ] Test for SSI Injection
- [ ] Test for XPath Injection
- [ ] Test for XQuery Injection
- [ ] Test for IMAP/SMTP Injection
- [ ] Test for Code Injection
- [ ] Test for Expression Language Injection
- [ ] Test for Command Injection
- [ ] Test for Overflow (Stack, Heap and Integer)
- [ ] Test for Format String
- [ ] Test for incubated vulnerabilities
- [ ] Test for HTTP Splitting/Smuggling
- [ ] Test for HTTP Verb Tampering
- [ ] Test for Open Redirection
- [ ] Test for Local File Inclusion

RIVER SECURITY

RECONNAISSANCE AND WEB APPLICATION PENETRATION TESTING

IT'S NOT JUST CHECKLISTS

WHO AM I?

COO, Principal and Founder at River Security

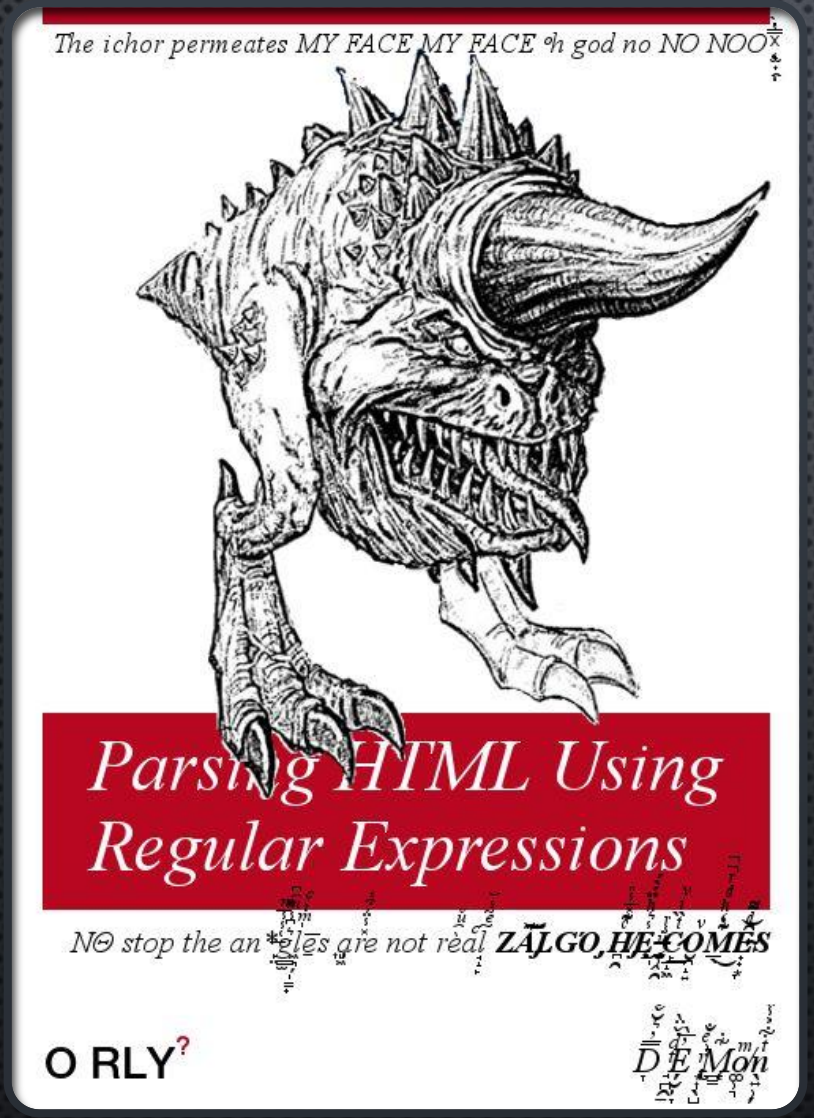
Principal Instructor at SANS

Co-Author of SEC550 – Cyber Deception, Attack Detection, Disruption and Active Defense

Short summary:

I show how criminals break-in,
and I help throw them back out…

GCIH GIAC Certified Incident Handler
GPEN GIAC Certified Penetration Tester
GSLC GIAC Security Leadership
GIAC Mobile Device Security Analyst
GDAT GIAC Defending Advanced Adversaries
GCTI GIAC Cyber Threat Intelligence
GCFA GIAC Certified Forensic Analyst



WHY THIS TALK?

- WEB IS UBIQUITOUS
- CONSIDERED BORING BY MANY
- NOT THE HIGHEST OF LEARNING CURVES.
  - YOU CAN PROVIDE VALUE FAST
- DUNNING KRUGER EFFECT
  - NOT JUST CHECKLIST, I.E. FRAMEWORKS , OWASP TOP10 , ETC.
- WEB IS REALLY A GREAT PLACE TO RESEARCH, BOUNTY AND GIVE YOUR CUSTOMERS VALUE.

RIVER SECURITY

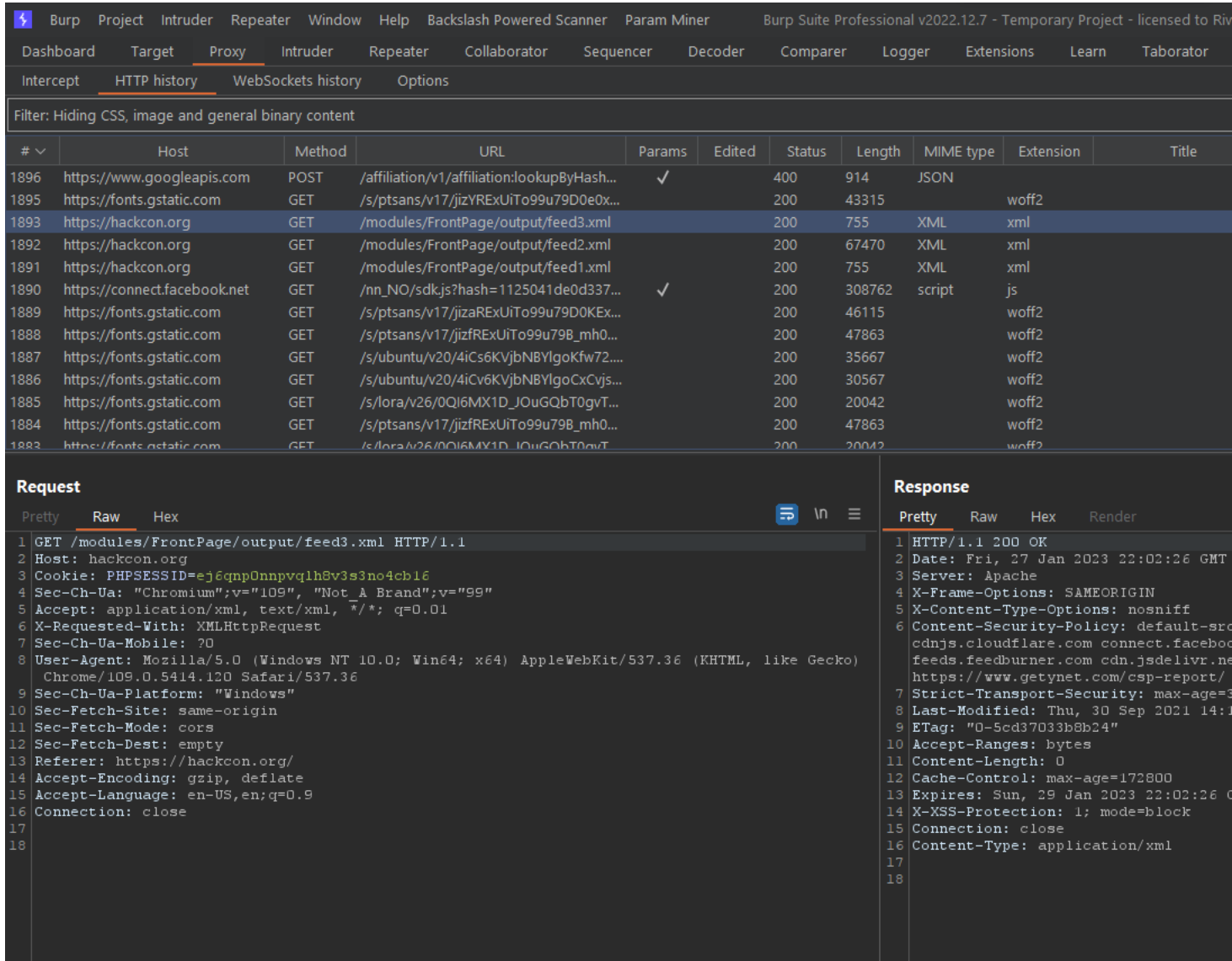
PORTSWIGGER TOP 10 ATTACKS

- 1 - Account hijacking using dirty dancing in sign-in OAuth-flows
- 2 - Browser-Powered Desync Attacks: A New Frontier in HTTP Request Smuggling
- 3 - Zimbra Email - Stealing Clear-Text Credentials via Memcache injection
- 4 - Hacking the Cloud with SAML
- 5 - Bypassing .NET Serialization Binders
- 6 - Making HTTP header injection critical via response queue poisoning
- 7 - Worldwide Server-side Cache Poisoning on All Akamai Edge Nodes
- 8 - Psychic Signatures in Java
- 9 - Practical client-side path-traversal attacks
- 10 - Exploiting Web3's Hidden Attack Surface: Universal XSS on Netlify's Next.js Library

https://portswigger.net/research/top-10-web-hacking-techniques-of-2022

Burp Suite – Tool of choice

- Defacto tool by pentester
- Strong fuzzing capabilities
- Extension support
- Very flexible and robust
- Well developed scanner
- Spidering engine with decent SPA support
- Cheat sheet: https://www.sans.org/posters/burp-suite-cheat-sheet/



Burp Extensions

Must have

- Active Scan ++
- Backslash Powered Scanner
- Param Miner
- Taborator

Nice to have

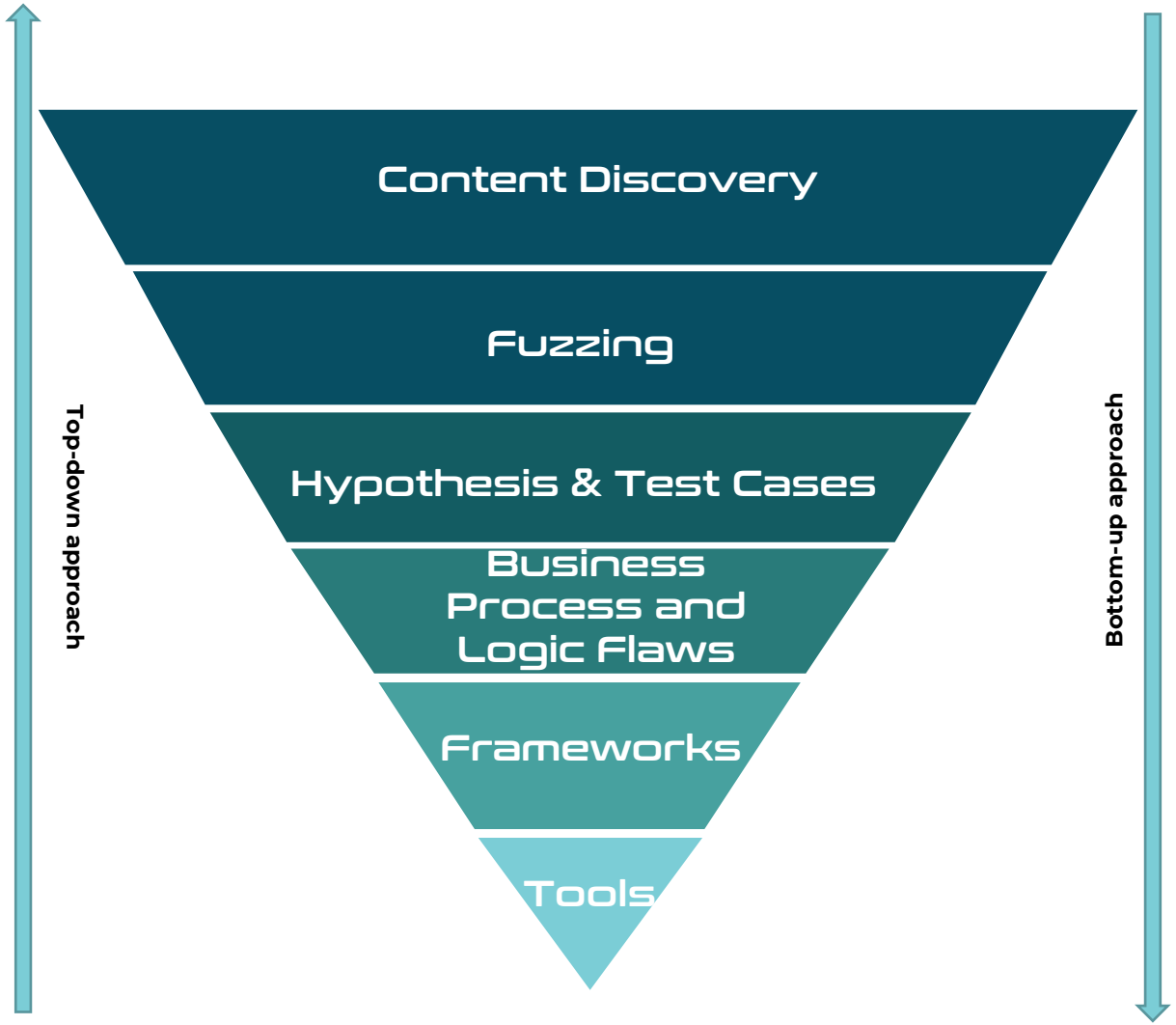
- Turbo Intruder
- Autorize
- Software Vulnerability Scanner
- Collaborator Everywhere

Honorable Mentions

- Freddy, deserialization scanner
- GraphQL raider
- JSON Web Tokens
- NTLM Challenge Decoder
- Retire.js
- Additional Scanner Checks

Finding Vulnerabilities Process Pyramid

Fully test the scope, every script and input



Producing High Value Penetration Tests

Reliable and consistent testing is important, and not relying on a single individuals' skills and efforts to complete a penetration test helps ensure the highest levels of standards.

Team Based Effort

Penetration Testing is a team effort, not an individual effort. Utilize a team to maximize the penetration test efforts.

Thoroughly Map Attack Surface

Leave no stone untouched. Many vulnerabilities are found in the "paths least travelled". Fully explore!

Reporting

Document findings, process, discrepancies and hypothesis. It will be useful now and later.

Hypothesis and Knowledge Sharing

A team is stronger. Produce hypothesis to uncover potential attacks across all layers. Strengthen the team knowledge by working as one.

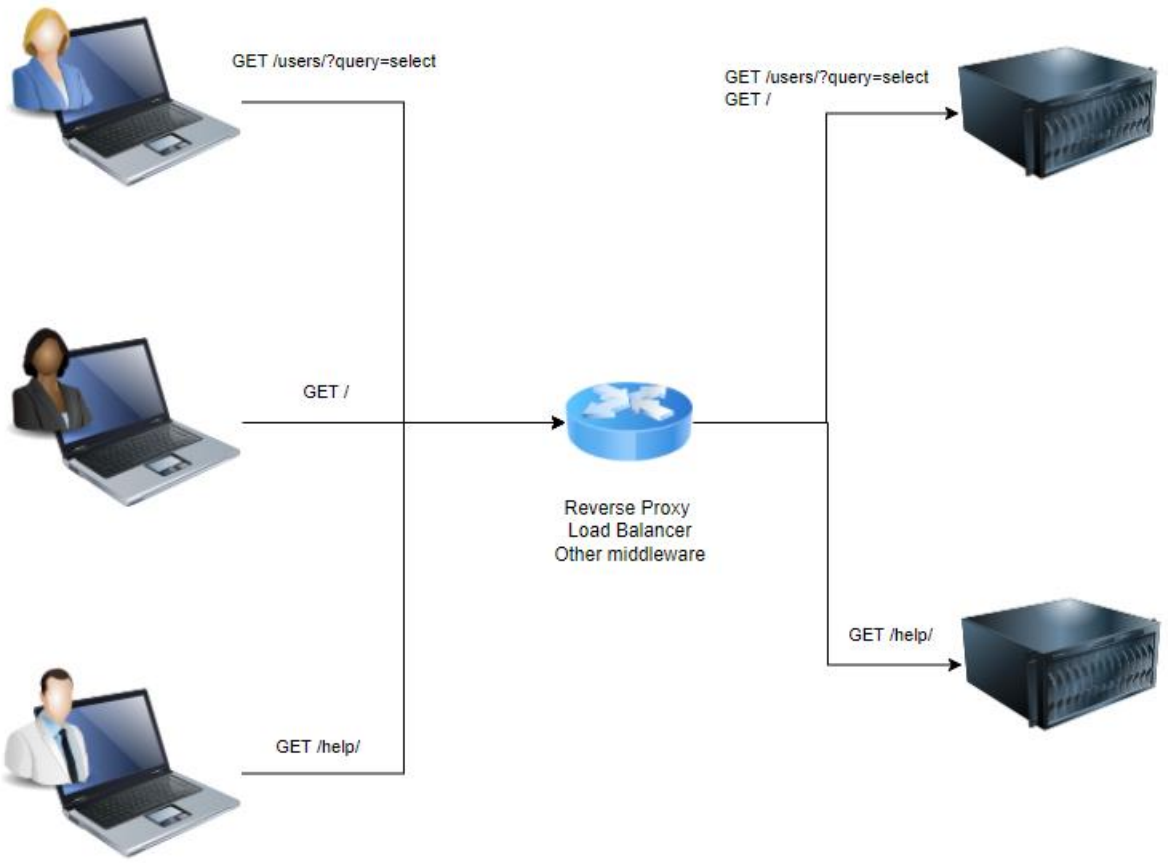
Goal: Find Everything

i. Map Browsable Attack Surface
ii. Find Unlinked Content & Params
iii. Repeat for each `Platform Distinctions` of the application

Leave no stone unturned. Many vulnerabilities are found in the "paths least travelled". Fully explore!

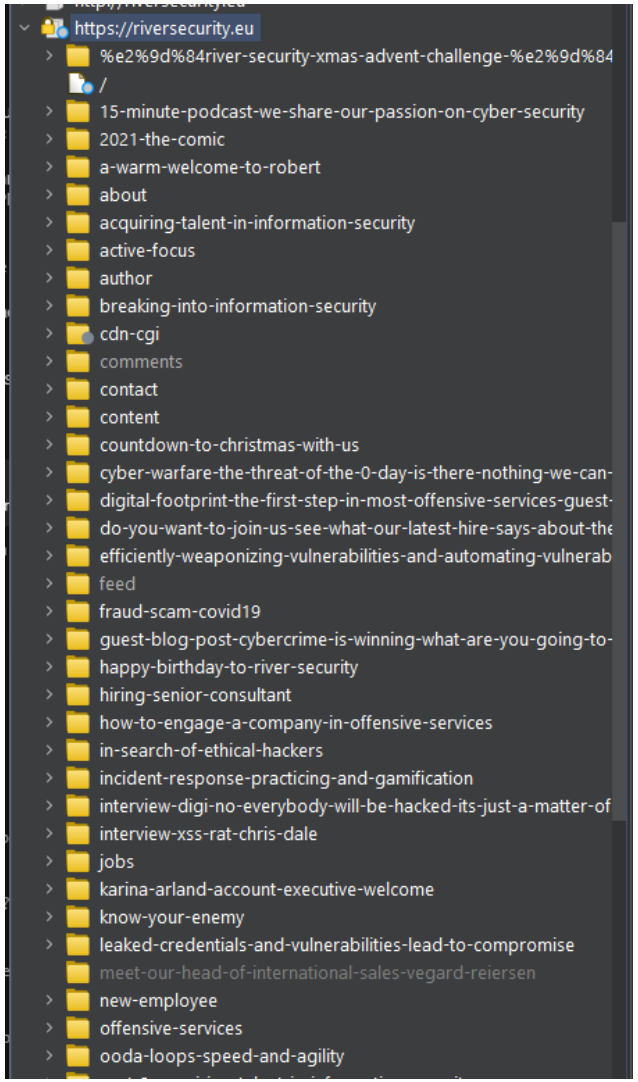
Platform Distinctions

- A web application may have several “platform distinctions”
  - Load-balancers may balance on an endpoint
  - Reverse proxies does the same
- Do your best if the target is split into different platforms
  - Each platform distinction should receive full test process



Map Browsable Attack Surface

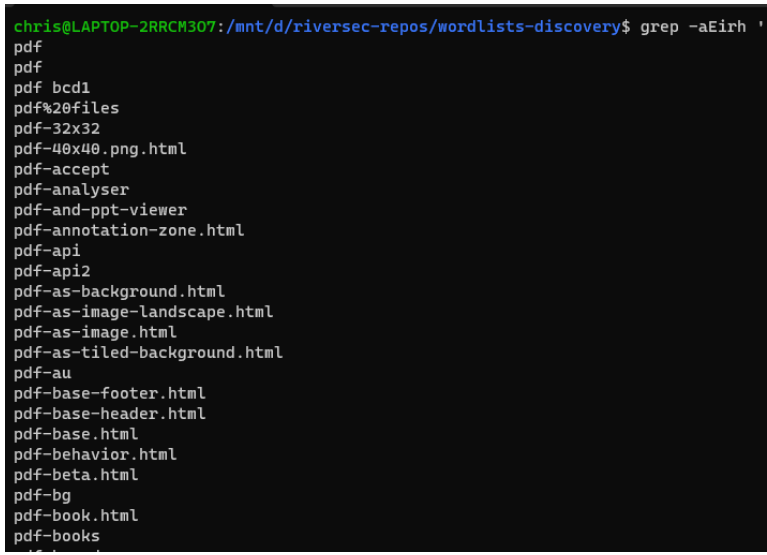
- Browse the entire application, discover all browsable content
  - Click
  - Use
  - Learn
- Use the Burp Suite Crawl feature on the top level of the application.
  - Has decent support for SPA as of Burp Suite v. >2
  - Helps build a complete sitemap
  - Use most complete configuration, which is the slowest
- For JavaScript, extract file paths and references.
  - CyberChef extract file paths module
  - GAP Burp Plugin
  - JSParser



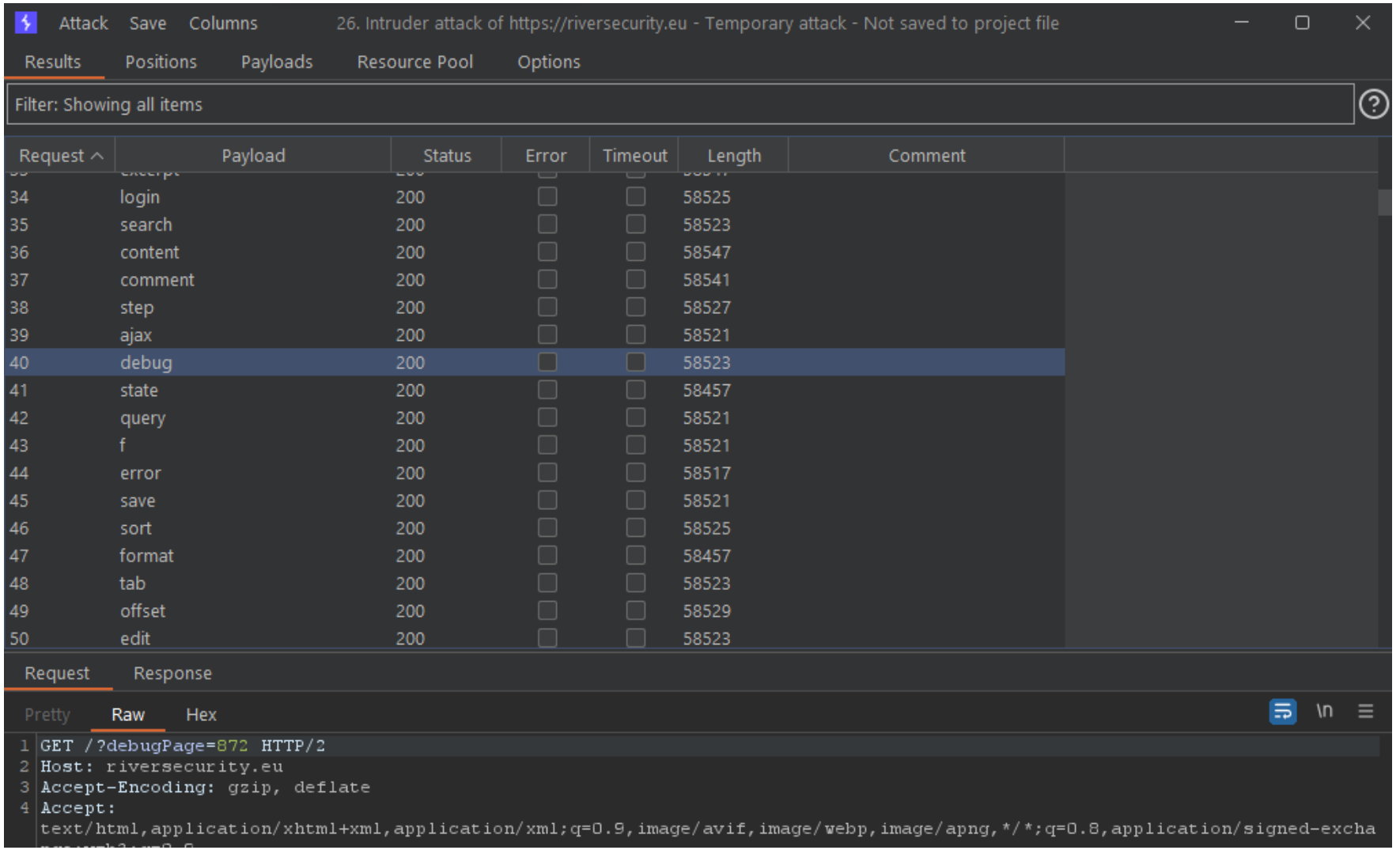
Find Unlinked Content

- Fuzz **verbs** and **functionality**, find additional content
  - For functionality such as e.g. **/?action=showUser&id=123** , try fuzzing the verb (i.e. show) with words like:
    - Add, delete, update and so on… i.e. making action=addUser, etc.
  - Useful wordlists inside of Burp:
    - Server-side variable names
    - Form field values
    - Form Field names
- Use and create wordlists based on target functionality
  - Example: A website relevant to *PDF's*

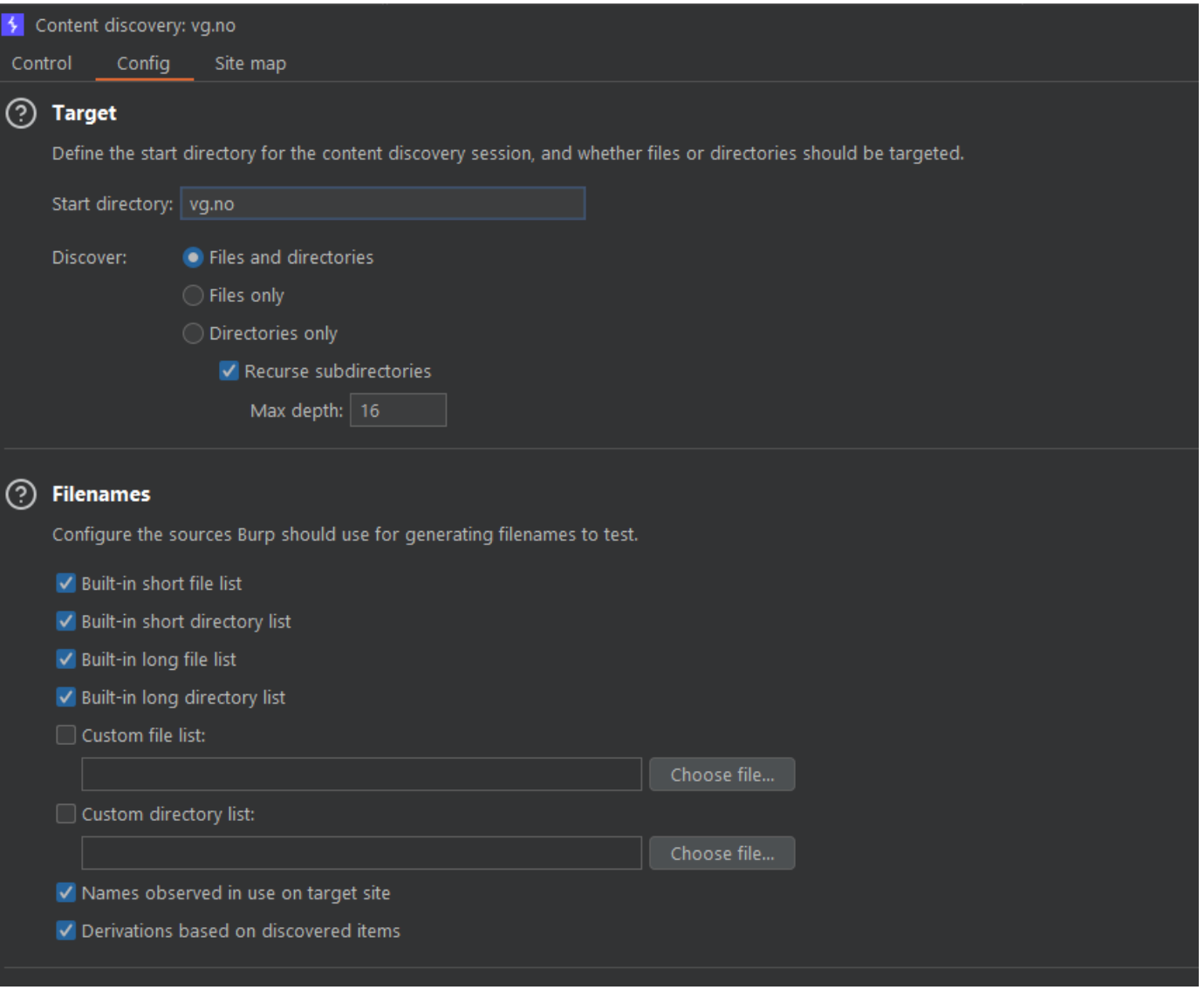
```
grep -aEirh '^pdf.*' * | sort | uniq
```

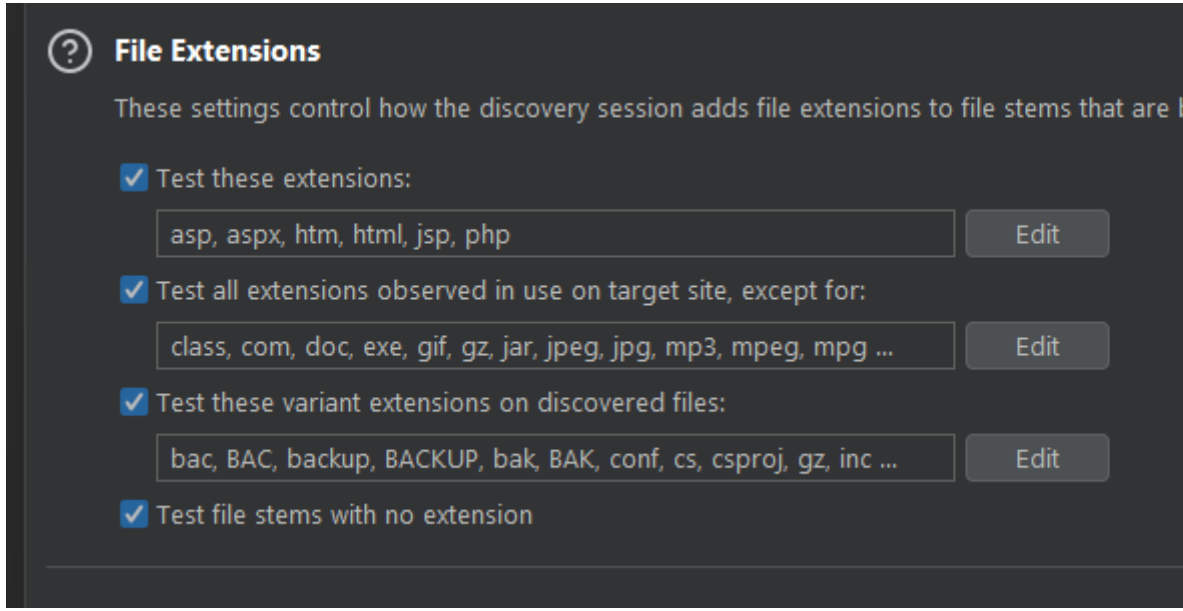


Verb Example /?page=872



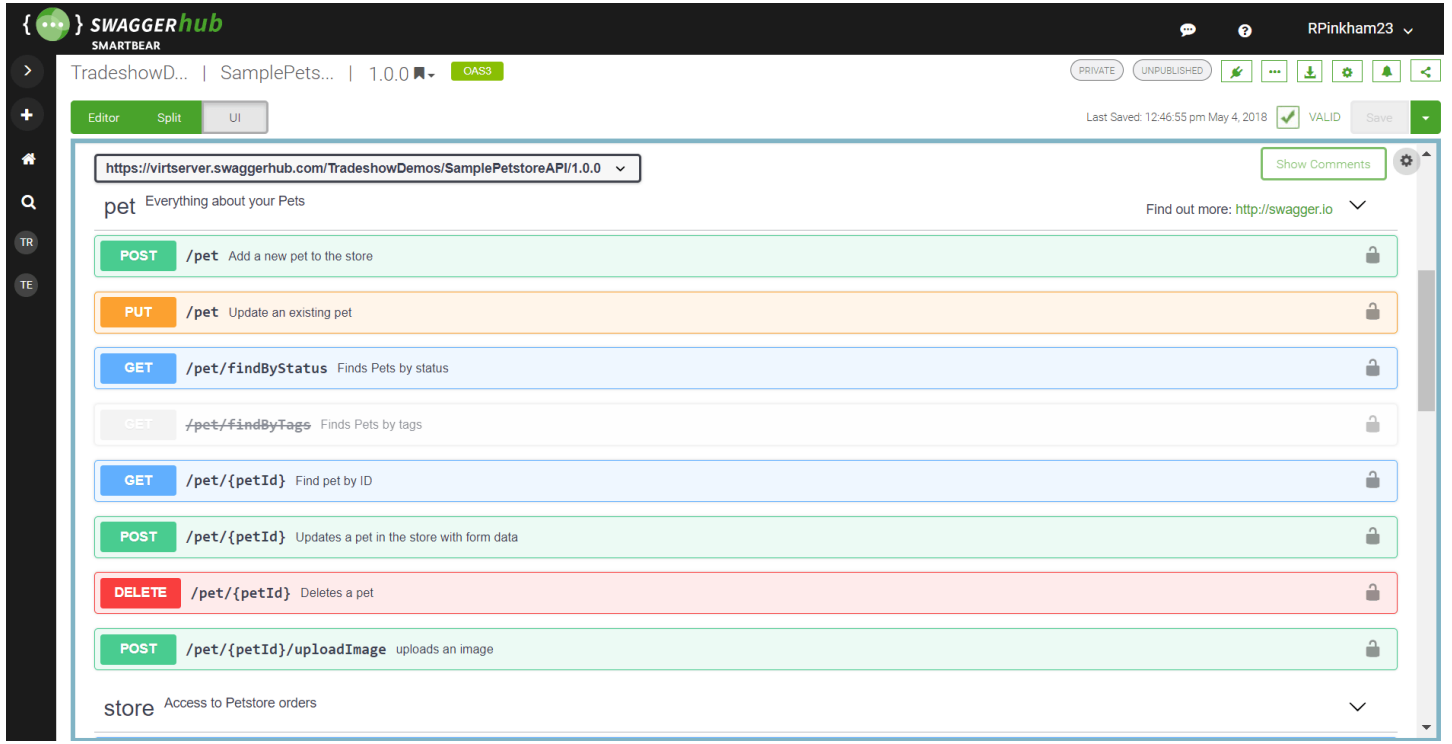
Content Discovery





OpenAPI / Swagger Specs

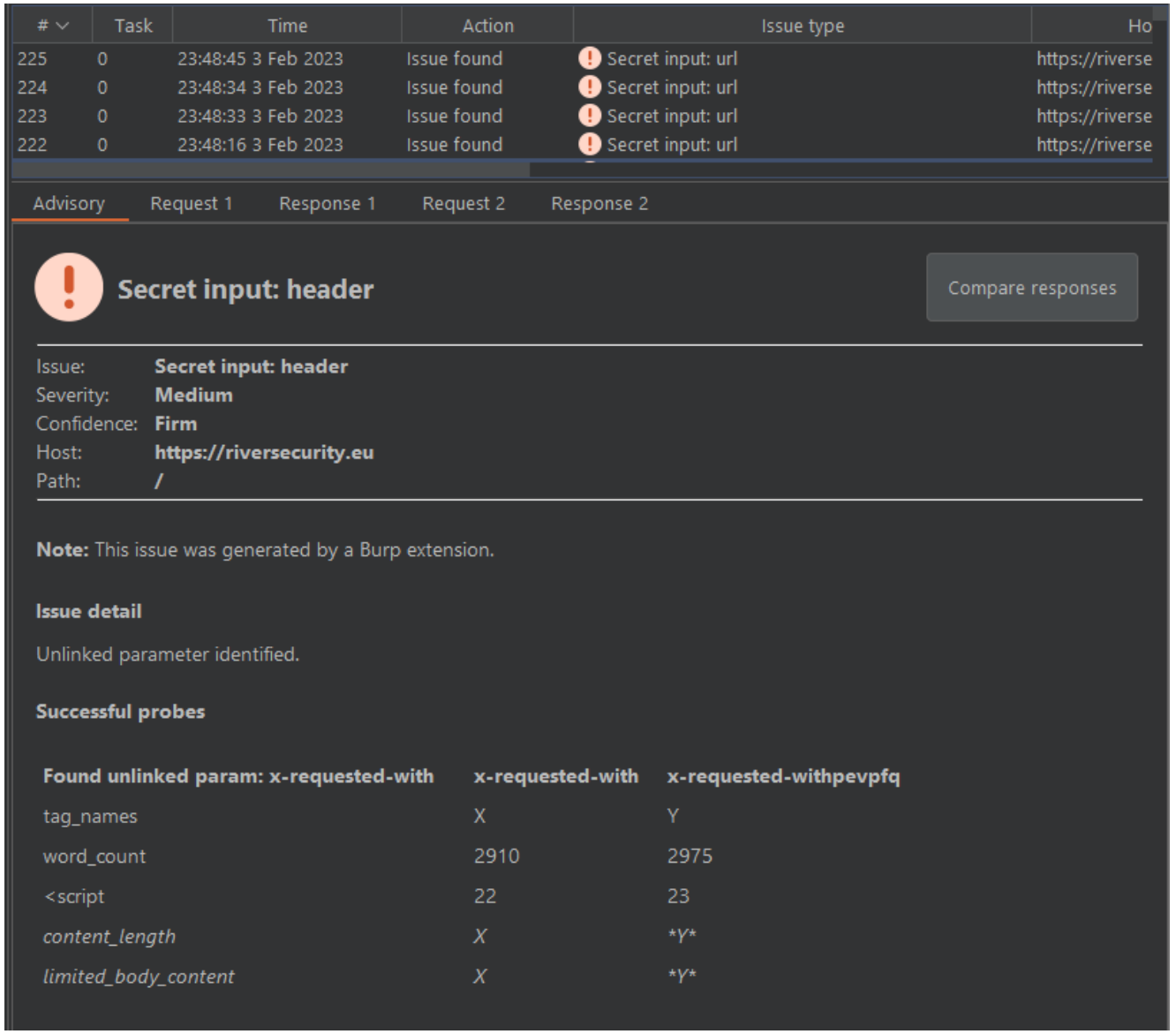
- If we can cheat, we should!
- Paints a picture of what the developers **intended** to include
- Still needs to do content discovery



Unlinked Parameters

Content Discovery

- Discover if there are any unlinked parameters
  - Particularly important on all Platform Distinctions
  - Any change based on a new parameter is interesting
  - GET, POST, Cookies, Headers
- Headers might bypass authentication
- Might find attack surface
- **Param miner extension!**

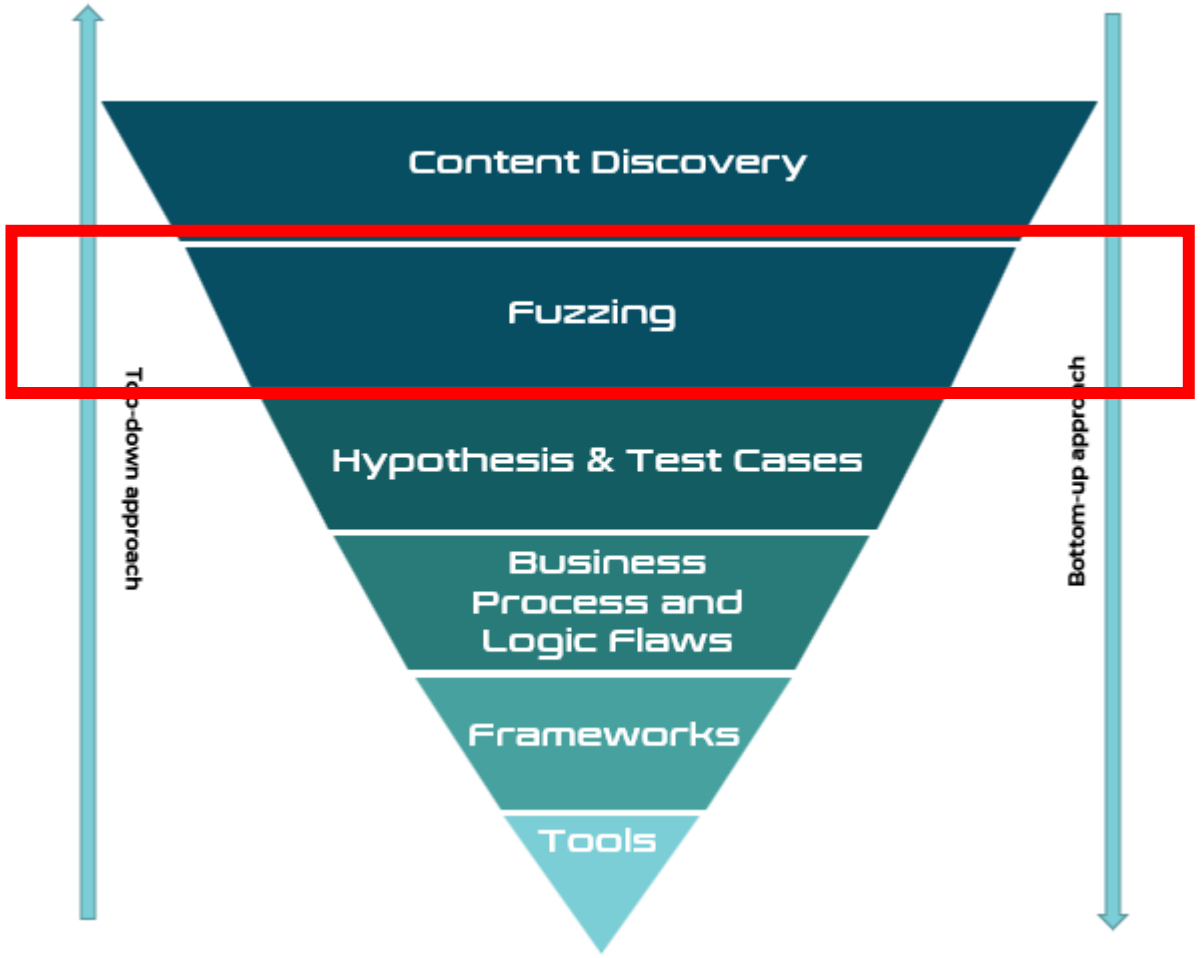


Archive.org

- WaybackRobots.py
- WaybackURLs.py

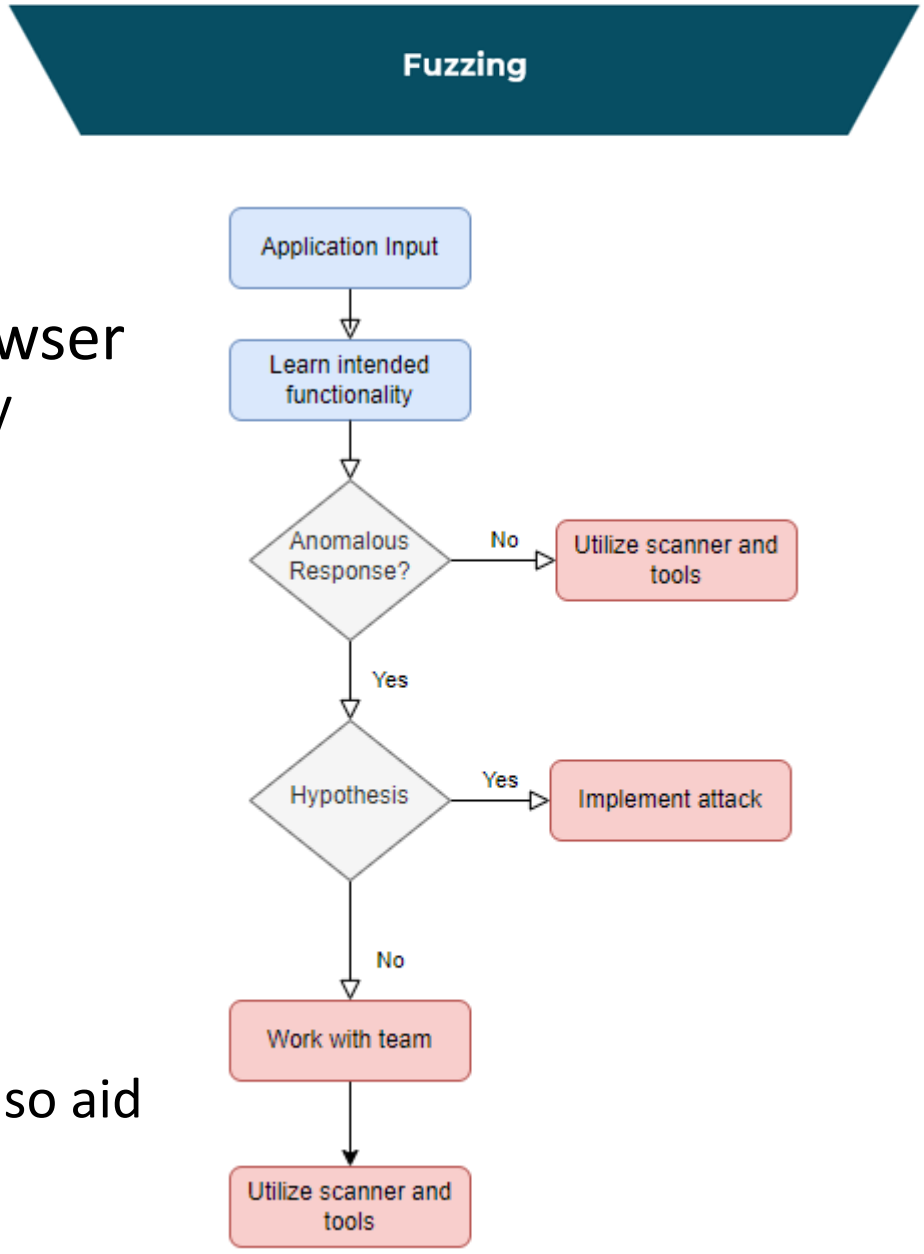
Fuzzing

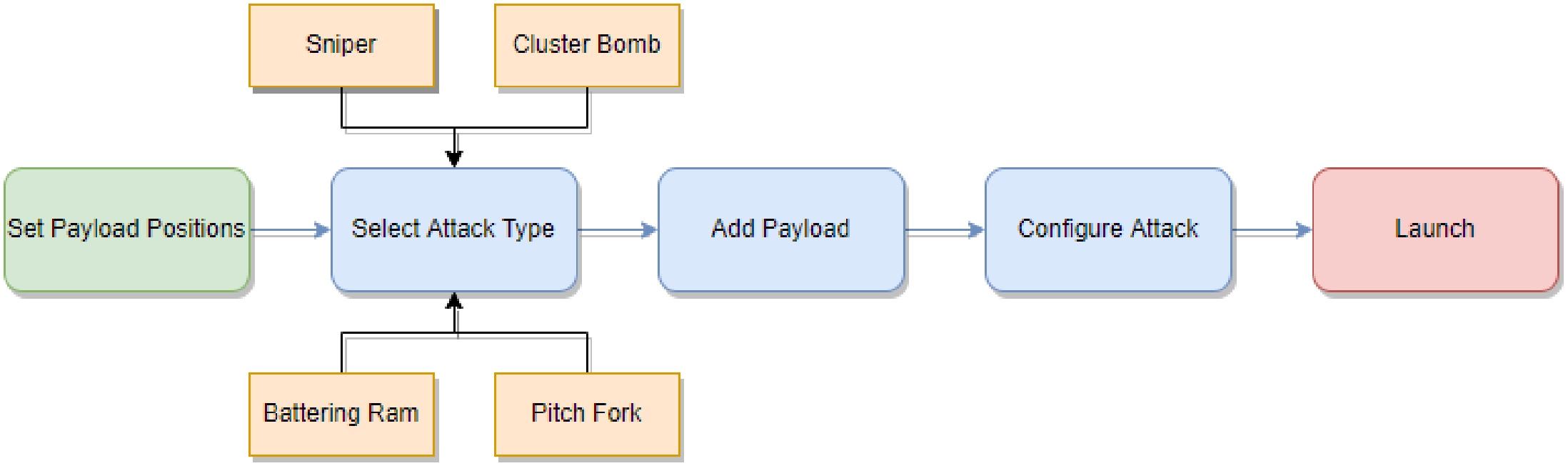
Find bytes and input producing different/unexpected results



Fuzzing Bytes 101

1. For-each script and input
2. Send their script to repeater / play with it in browser
   - Determine properly how the functionality works and try related attack
3. Send to intruder and fuzz
   - %00 to %FF
     - URL Decode targets Middleware
     - URL Encode targets App
   - Anomalies, discrepancies, interesting results?
     - Create Hypothesis
     - Work with team if you cannot produce hypothesis
   - Use wordlists
4. Utilize vulnerability scanner
   - Backslash Powered Scanner and other extensions will also aid here.
5. Scanner results? Update methodology





Two Examples

Not a one size fits all, but produces very interesting results

Asdf.asxp produces 500 server error

Fuzzing

Request

Pretty Raw Hex

```
1 GET /asdf.aspx HTTP/2
2 Host: [redacted]
3 Cookie: [redacted]
4 Sec-Ch-Ua: [redacted]
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Windows"
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/109.0.5414.120 Safari/537.36
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q
  =0.8,application/signed-exchange;v=b3;q=0.9
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16
17
```

Search... 0 matches

Response

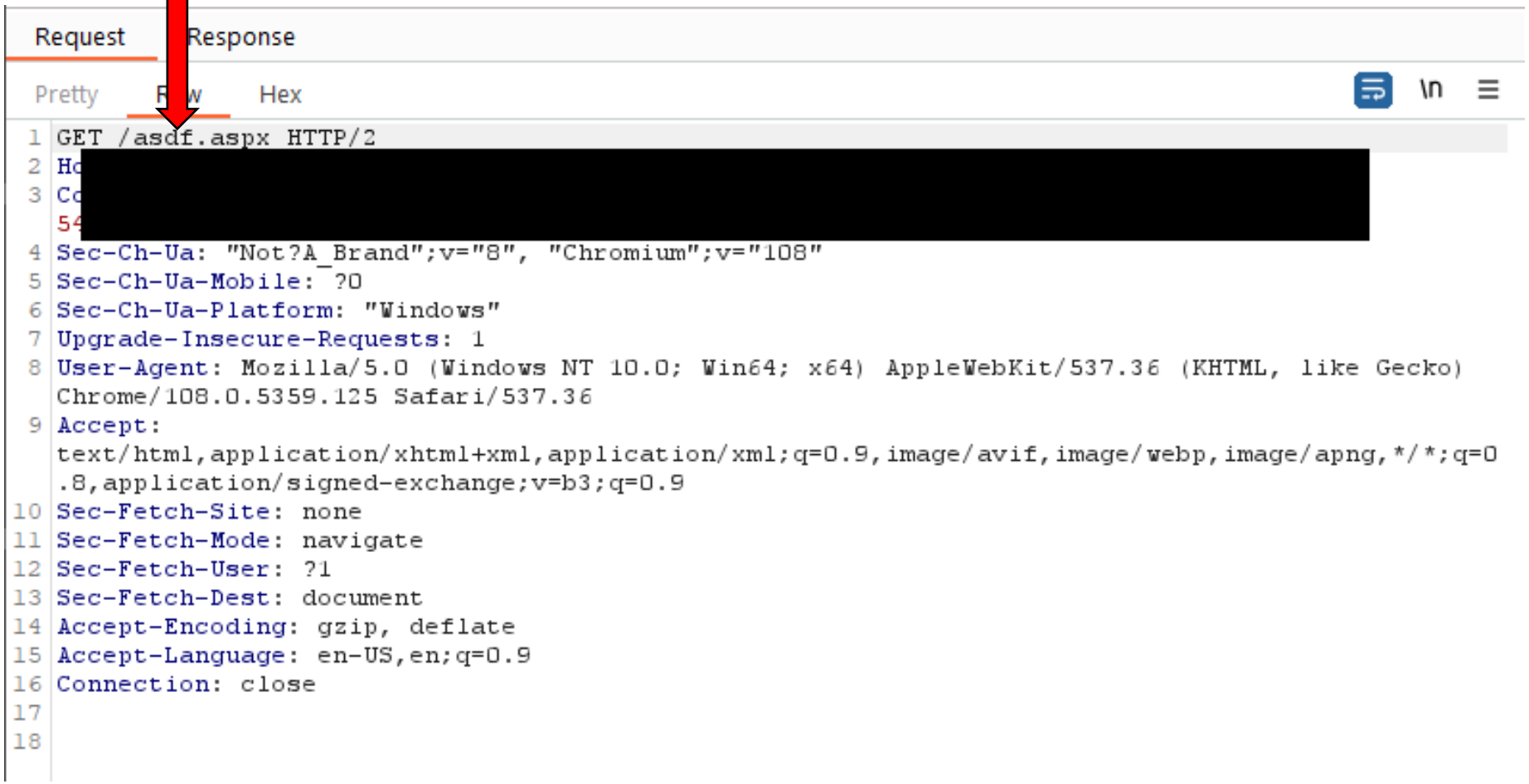
Pretty Raw Hex Render

```
1 HTTP/2 500 Internal Server Error
2 Cache-Control: private
3 Content-Type: text/html; charset=utf-8
4 Server: Microsoft-IIS/10.0
5 X-Frame-Options: SAMEORIGIN
6 X-Aspnetmvc-Version: 5.2
7 Set-Cookie: sessi[redacted]path=/; secure; SameSite=None
8 X-Content-Type-Options: nosn
9 X-Powered-By: ASP.NET
10 Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
11 Content-Security-Policy: default-src 'unsafe-eval' 'unsafe-inline' 'self' https: data:;
   frame-a
12 Content[redacted]
   'self';
13 Date: Thu, 26 Jan 2023 23:42:52 GMT
14
15
16
17 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
18
19 <html xmlns="http://www.w3.org/1999/xhtml">
20   <head>
       <title>
21
22     </title>
       <link href="css/default.css" rel="stylesheet" type="text/css" />
     </head>
23   <body>
24     <form name="form1" method="post" action=[redacted] id="form1">
```

Search... 0 matches

Bytes Examples

Payload here



Fuzzing

Attack Save Columns

Results Positions Payloads Resource Pool Options

Filter: Hiding 4xx responses

Request	Payload	Status	Error	Timeout	Length
38	%25	500			2179
39	%26	500			2179
43	%2A	500			2179
49	%3A	500			2179
51	%3C	500			2179
53	%3E	500			2179
54	%3F	500			2179
0		500			2325
33	%20	500			2325
34	%21	500			2325
35	%22	500			2325
36	%23	500			2325
37	%24	500			2325
40	%27	500			2325
41	%28	500			2325
42	%29	500			2325
45	%2C	500			2325
46	%2D	500			2325
47	%2E	500			2325
50	%3B	500			2325
52	%3D	500			2325
55	%40	500			2325
56	%5B	500			2325
58	%5D	500			2325

Request Response

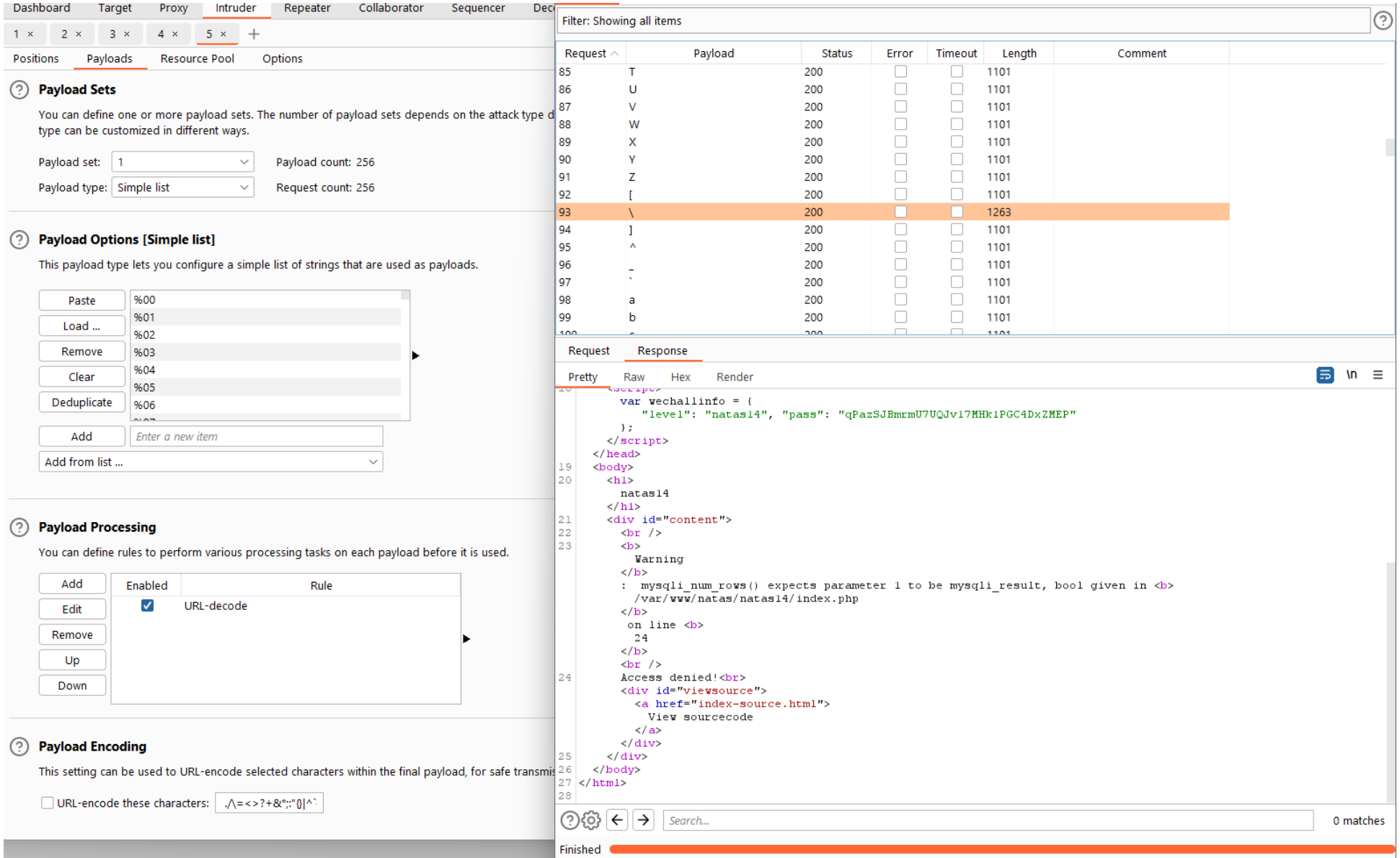
Pretty Raw Hex Render

```
1 HTTP/2 500 Internal Server Error
2 Cache-Control: private
3 Content-Type: text/html; charset=utf-8
4 Server: Microsoft-IIS/10.0
5 X-Content-Type-Options: nosniff
6 X-Powered-By: ASP.NET
7 Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
8 Content-Security-Policy: default-src 'unsafe-eval' 'unsafe-inline' 'self' htt
9 Content-Security-Policy-Report-Only: default-src 'unsafe-eval' 'unsafe-inline
                https://www.google.com                https://maps.googleap
  https://maps.gstatic.com;              frame-ancestors 'self'; form-action
10 Date: Fri, 03 Feb 2023 14:13:31 GMT
11
12
13
14 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3
15
16 <html xmlns="http://www.w3.org/1999/xhtml">
17   <head>
       <title>
18
19     </title>
       <link href="css/default.css" rel="stylesheet" type="text/css" />
```

Search...

	A	B	C	D	E
1	HTTP Status Code	Byte	URL decoded	Reasoning	Comment
2	500	%25	%	URL	
3	500	%26	&	URL	
4	500	%2A	*	FILE	Wilcard
5	500	%3A	:	FILE	ADS
6	500	%3E	>	FILE	Redirect
7	500	%3F	?	URL	
8	500	%3C	<	FILE	Redirect
9	404	%2B	+	URL	

A Single Character



Occam's Razor

Among competing hypothesis, the one with the fewest hypothesis is often correct.

Avoiding Rabbit Holes

- A rabbit hole is: A potential exploit condition which will take up a lot of time to research.
- Prioritize “**width**” rather than “**depth**”
  - Focus on rabbit holes with the time left after the scope is covered
- Structure your work scope
  - Duration of the engagement / How much time do we have left?
  - Hours spent – Work left
    - Each hour spent impacts the total value spent on the engagement
  - How many scripts, functions and other things do we have left to test?
  - **Do we need to get someone else to help us conclude a rabbit hole?**
- Large applications: split into smaller parts to help team prioritize

Using Wordlists

With our fuzzing efforts, wordlists can help produce valuable results, e.g., anomalies in cases of:

- Different results
- Timing impacted
- External server interaction

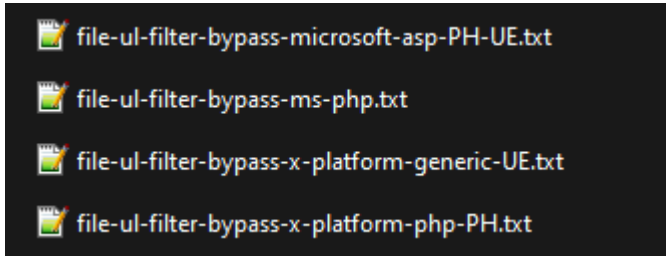
```
    Directory: D:\riversec-repos\wordlists-discovery


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10/10/2022  12:31 PM                assetnote
d-----         2/27/2022   6:24 PM                disney
d-----         2/27/2022   6:24 PM                flintstones
d-----         2/27/2022   6:24 PM                general purpose
```

Use wordlists that help you target technology and hypothesis. Great starting points:

- SecLists: https://github.com/danielmiessler/SecLists
- AssetNote: https://wordlists.assetnote.io/

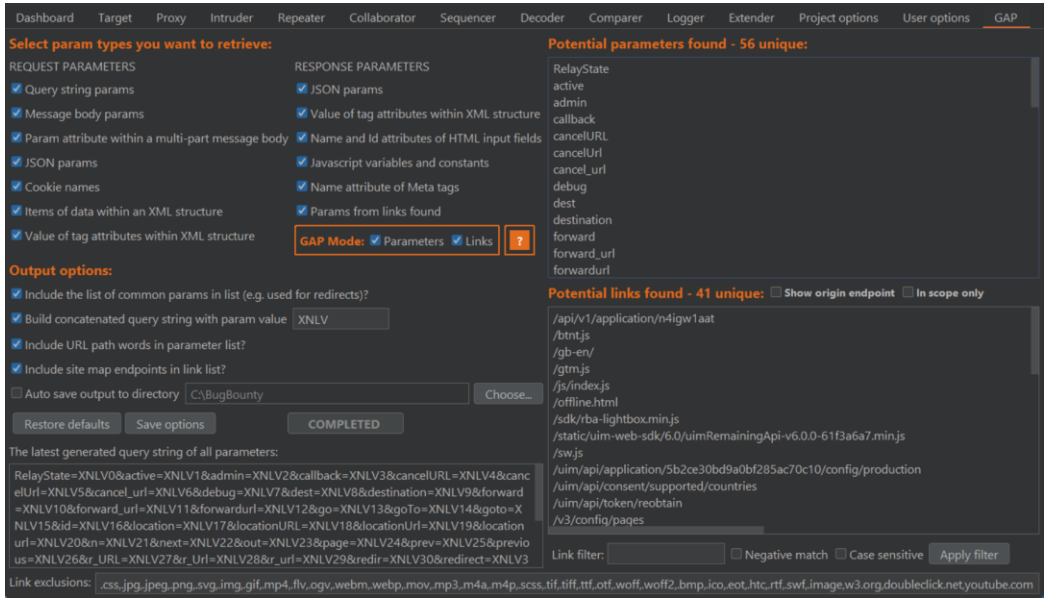
Take time to learn what these wordlists contain; it will help you learn when to apply them

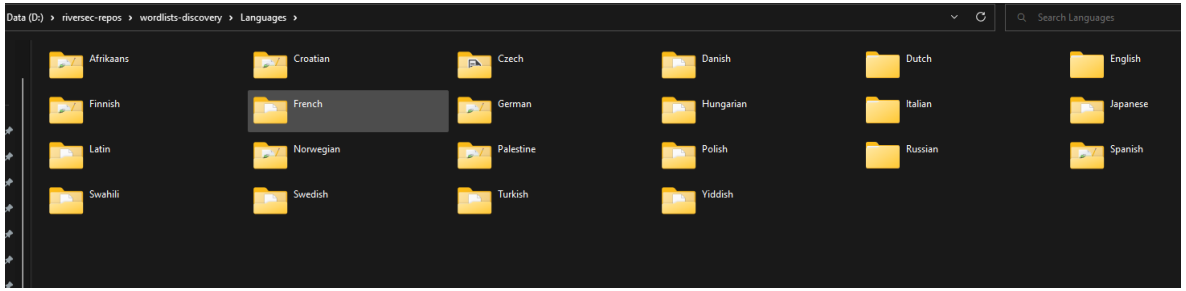


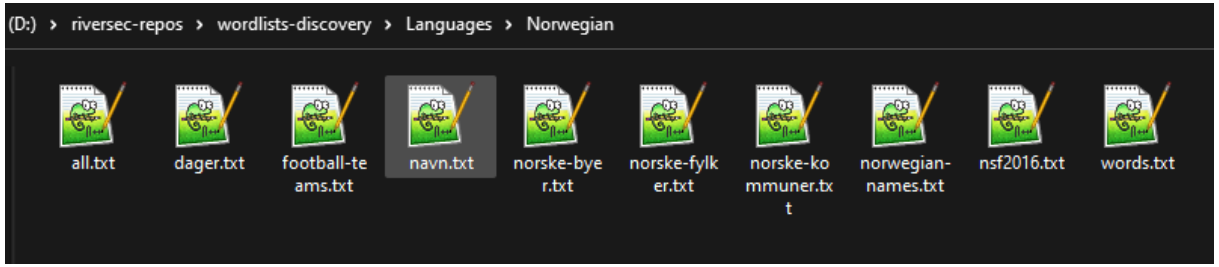
Building Good Wordlists

- DigiNinja's CeWL
  - Filter away stop-words
- Burp Suite GAP extension
- URL Shortners bruteforce results
- http_disallowed_entries_CiscoTopMillion
- Wiki's are a good source of wordlist

Fuzzing








```
grep -aEirh "^api-co*"  | tr '[:upper:]' '[:lower:]' | sort | uniq > /mnt/d/tmp/api-co.txt
grep -aEirh "^api-bc*"  | tr '[:upper:]' '[:lower:]' | sort | uniq > /mnt/d/tmp/api-bc.txt
grep -aEirh "^extern*"  | tr '[:upper:]' '[:lower:]' | sort | uniq > /mnt/d/tmp/extern.txt
grep -aEirh "^extern.*"  | tr '[:upper:]' '[:lower:]' | sort | uniq > /mnt/d/tmp/extern.txt
grep -aEirh "^extern.*$"  | tr '[:upper:]' '[:lower:]' | sort | uniq > /mnt/d/tmp/extern.txt
grep -aEirh "^besokr.*$"  | tr '[:upper:]' '[:lower:]' | sort | uniq > /mnt/d/tmp/besokr.txt
grep -aEirh "^compan.*$"  | tr '[:upper:]' '[:lower:]' | sort | uniq > /mnt/d/tmp/compan.txt
grep -aEirh "^dailyc.*$"  | tr '[:upper:]' '[:lower:]' | sort | uniq > /mnt/d/tmp/dailyc.txt
grep -aEirh "^meetin.*$"  | tr '[:upper:]' '[:lower:]' | sort | uniq > /mnt/d/tmp/meetin.txt
grep -aEirh "^flyerl.*$"  | tr '[:upper:]' '[:lower:]' | sort | uniq > /mnt/d/tmp/flyerl.txt
```

IIS Short Name Scanning

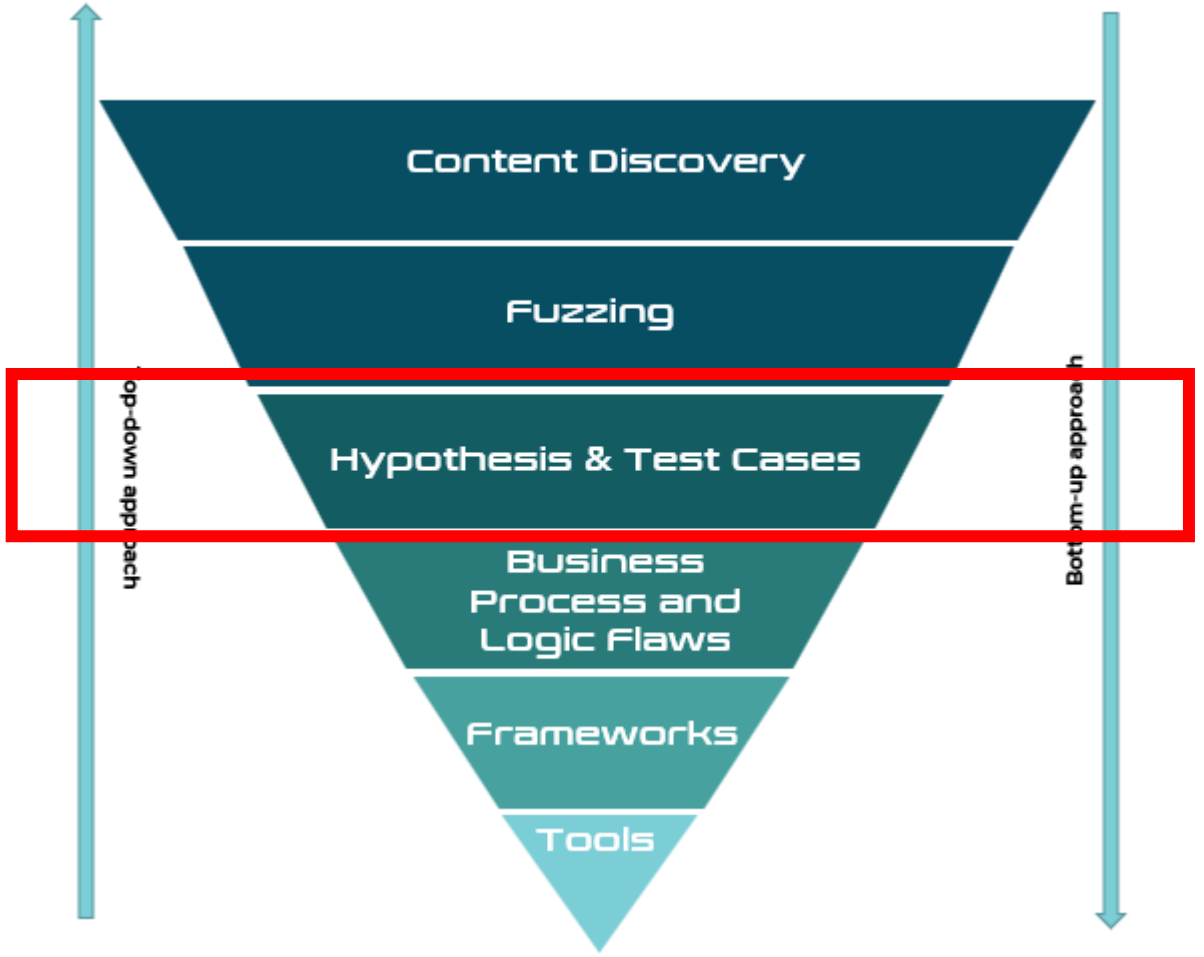
Fuzzing

```
PS C:\tmp\repos\IIS_shortname_Scanner> C:\Python27\python.exe .\iis_shortname_Scan.py https://                /metadatacard/
Server is vulnerable, please wait, scanning...
[+] /metadatacard/m~1.* [scan in progress]
[+] /metadatacard/me~1.*        [scan in progress]
[+] /metadatacard/met~1.*       [scan in progress]
[+] /metadatacard/meta~1.*      [scan in progress]
[+] /metadatacard/metad~1.*     [scan in progress]
[+] /metadatacard/metada~1.*    [scan in progress]
[+] /metadatacard/metada~1.z*   [scan in progress]
[+] /metadatacard/metada~1.zi*  [scan in progress]
[+] /metadatacard/metada~1.zip* [scan in progress]
[+] File /metadatacard/metada~1.zip*    [Done]
---------------------------------------------------------------
File: /metadatacard/metada~1.zip*
---------------------------------------------------------------
0 Directories, 1 Files found in total
```

Hypothesis and test cases

Be creative and utilize your team.

Test and conclude hypothesizes



Utilize the Team

- Pen Testing is a team effort, not an individual effort.
- Utilize a team to maximize the penetration test efforts.
- Ensure you can work together
- If you can't properly explain and create valid hypothesis
  - Ask your team
  - Work together (Knowledge transfer)
- Source your rabbit holes to team members

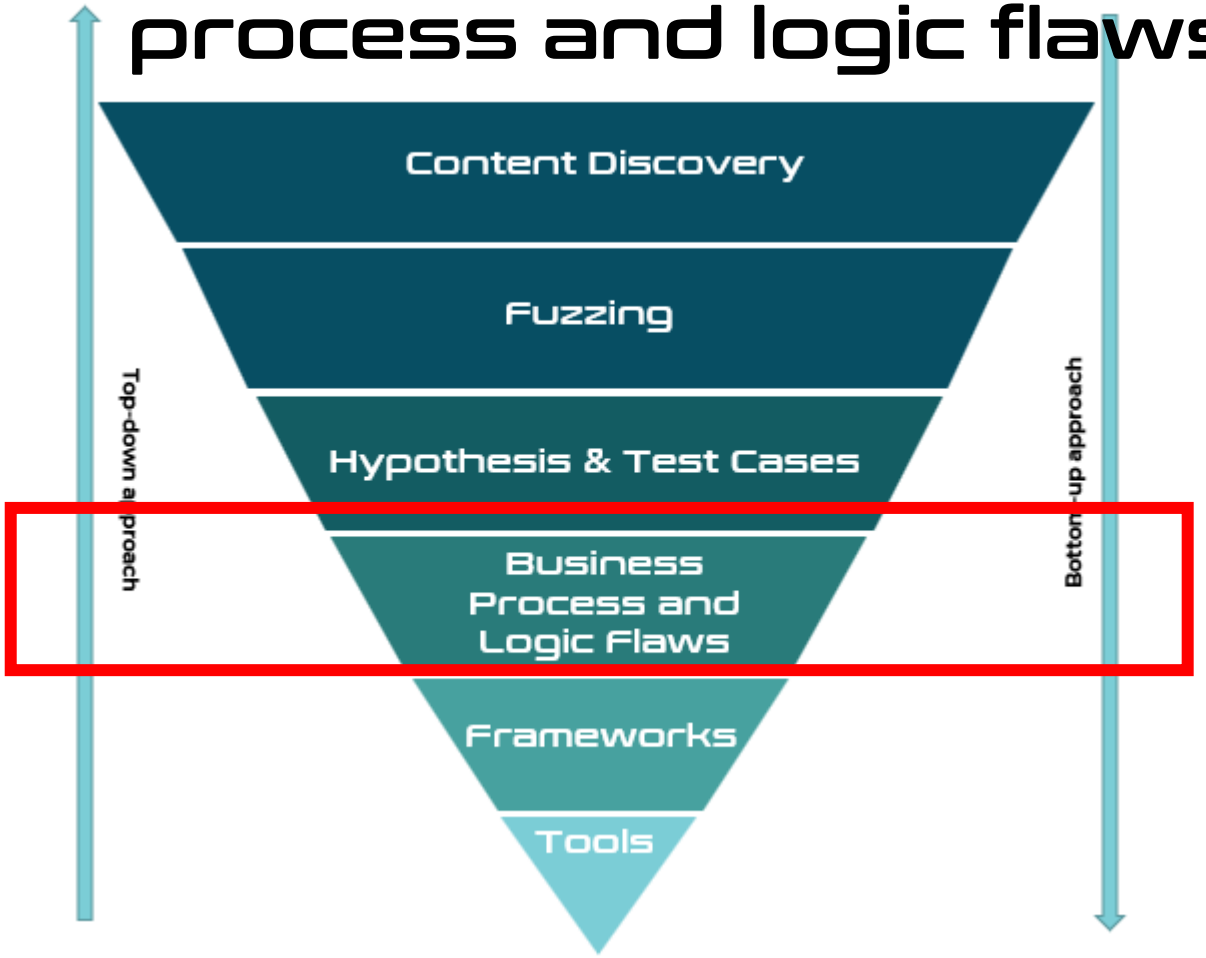
Hypothesis

I am seeing that : > < and * are influencing file reads of the file server. I want to explore Local File Inclusion, SSRF and similar kinds of vulnerabilities

	A	B	C	D	E
1	HTTP Status Code	Byte	URL decoded	Reasoning	Comment
2	500	%25	%	URL	
3	500	%26	&	URL	
4	500	%2A	*	FILE	Wilcard
5	500	%3A	:	FILE	ADS
6	500	%3E	>	FILE	Redirect
7	500	%3F	?	URL	
8	500	%3C	<	FILE	Redirect
9	404	%2B	+	URL	

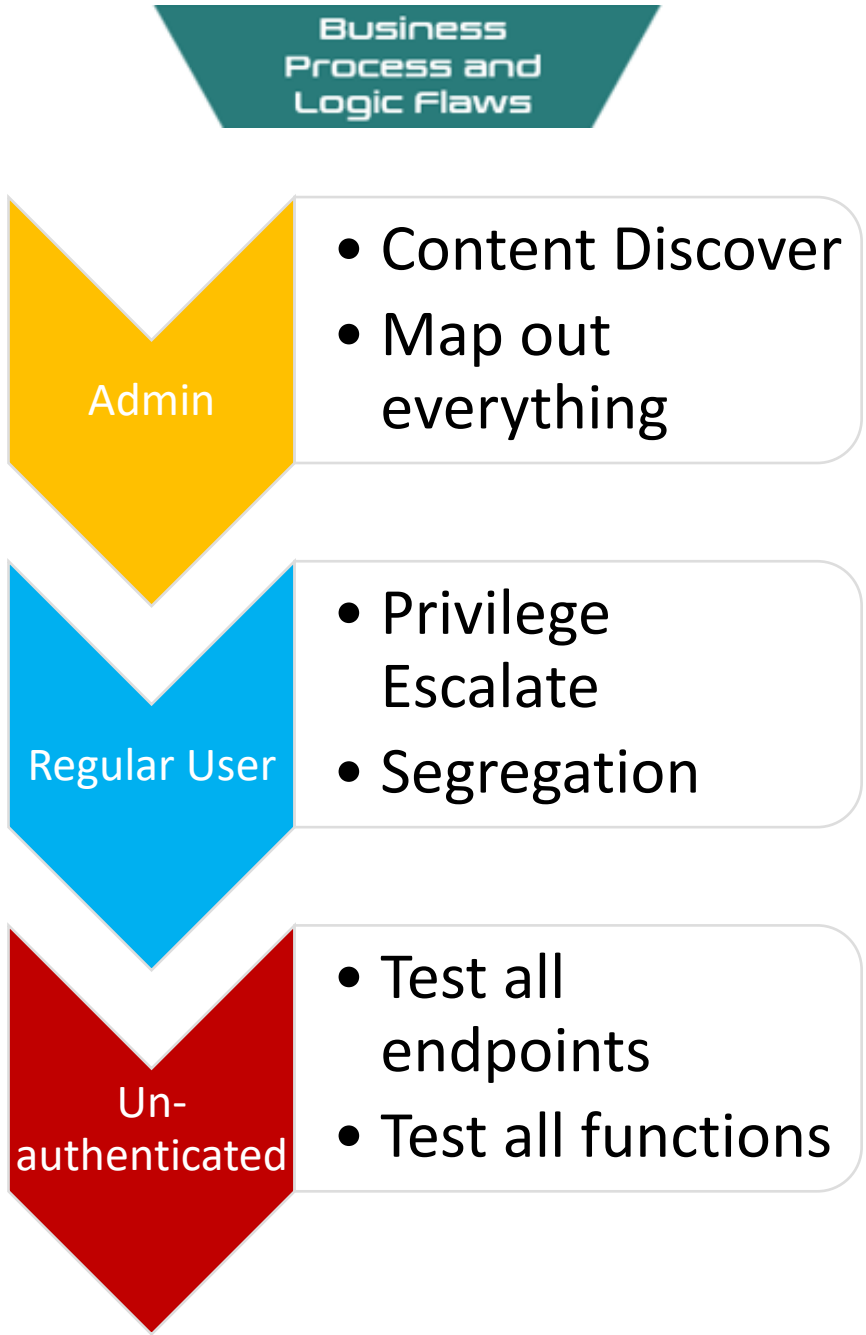
Business Process and Logic Flaws

With extensive knowledge of the target, explore process and logic flaws



A Quickie on Authentication

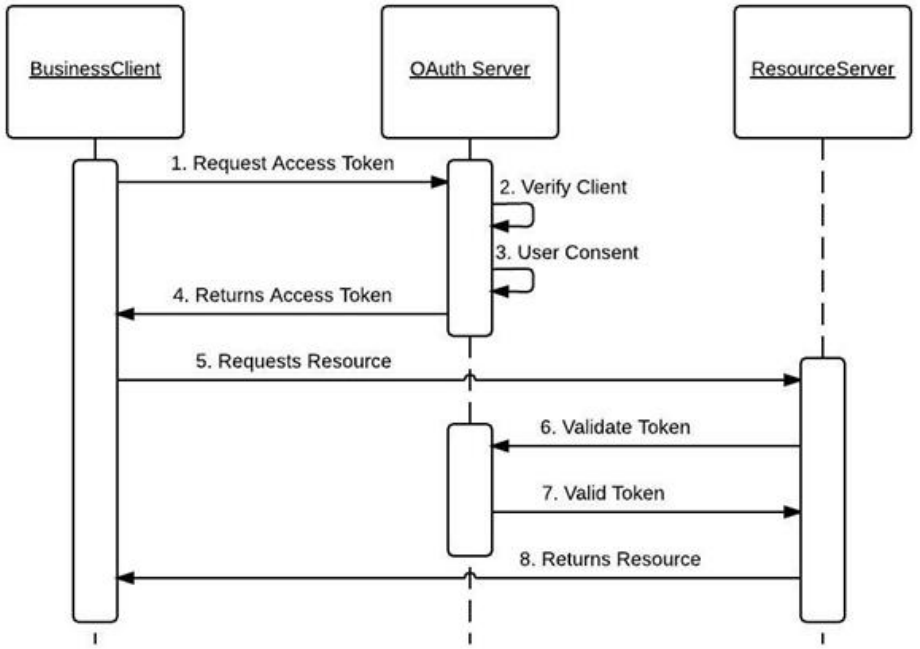
- Technically a part of discovery / scoping / planning
  - Pentesting is not a one-size fits all
  - Work with the customer to find THEIR needs
- Applications typically have different privileges levels:
  - Super Admin
  - Customer admin
  - User
  - Unauthenticated
- Regardless of the scope you have worked through with your customer, ask for super admin
  - Map out everything as super admin, you don't have to pentest it, but build overview of functionality
- Make sure customer admin, user and unauthenticated is secure, and provides segregation



Map Out Application Flows

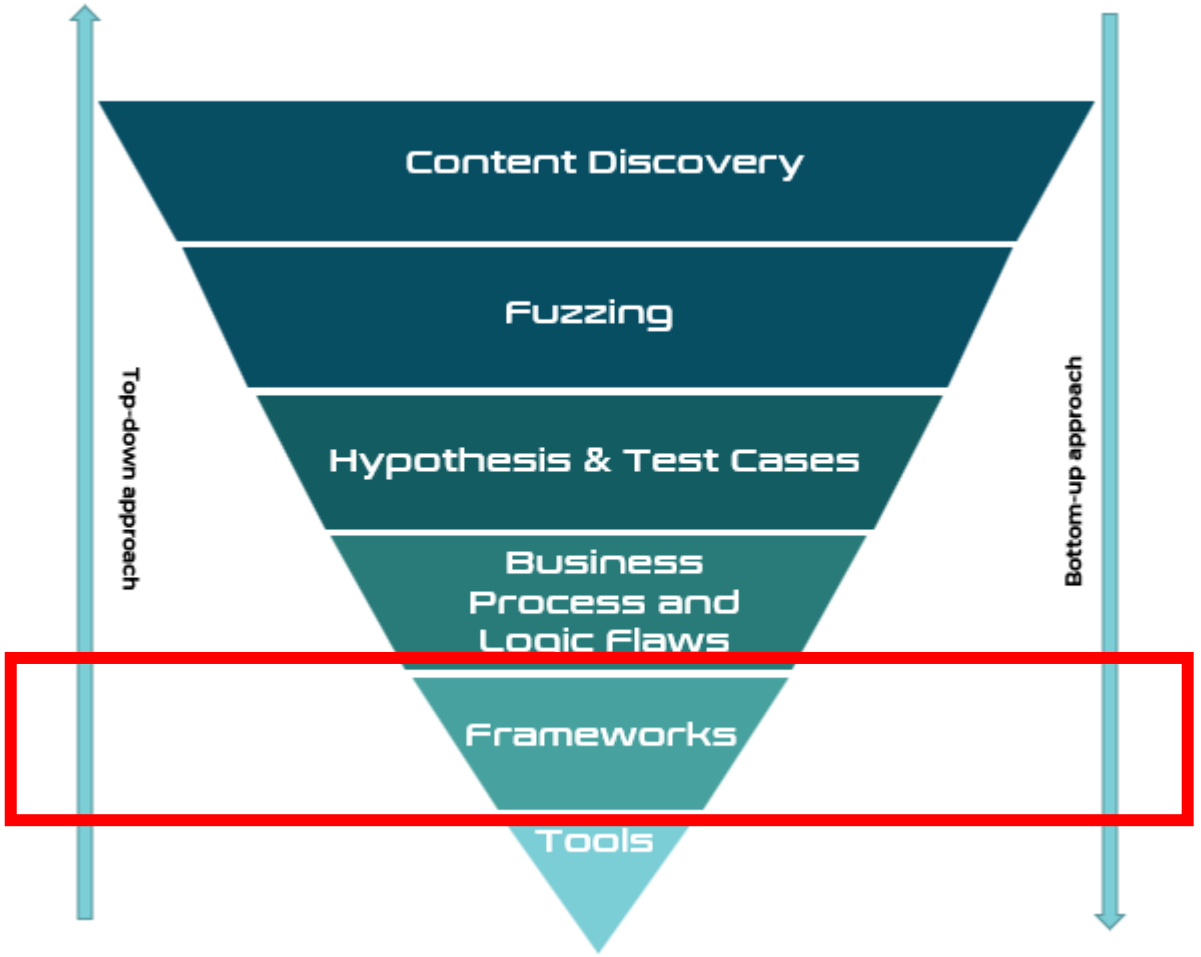
Business Process and Logic Flaws

- Mapping out the flow of behavior
- Draw.io / Diagrams.net is easy quick win
- Helps look at things from a bird eye perspective
- Map out requests and response
- Example flows:
  - Purchasing
  - Authentication
  - Impersonation / privilege escalation
  - Password reset flow
  - …



Frameworks

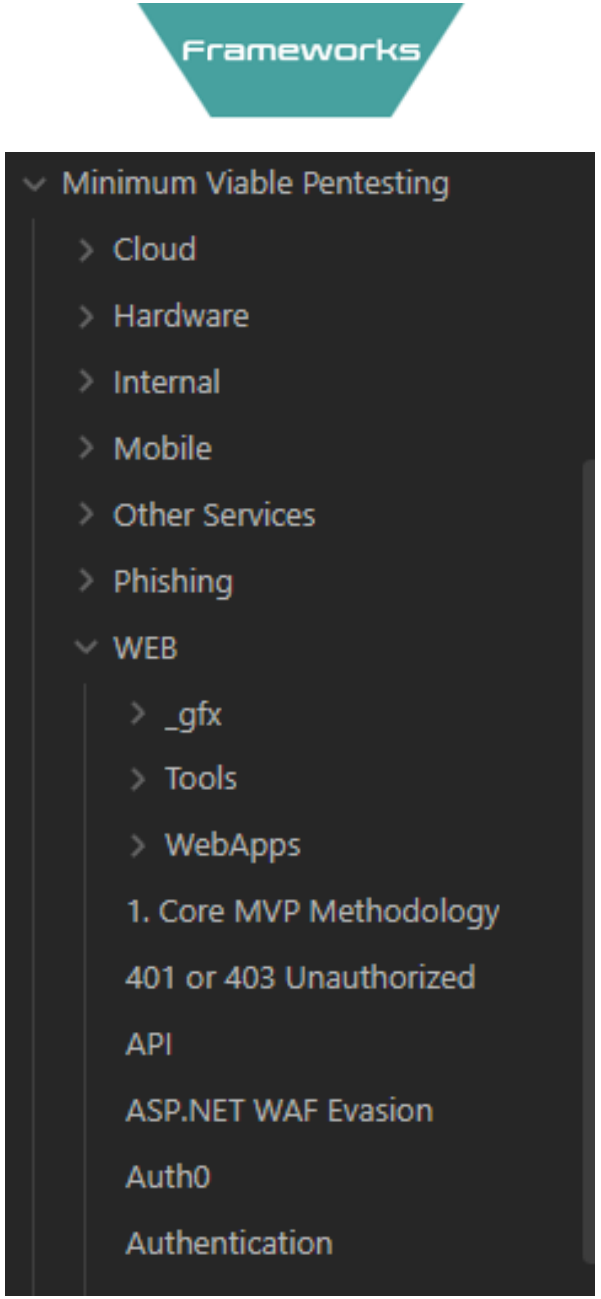
Compliance and pentest support. Utilize frameworks.



Minimum Viable Penetration Testing

Define an **absolute minimum** of activity to perform on a certain system or piece of technology or application.

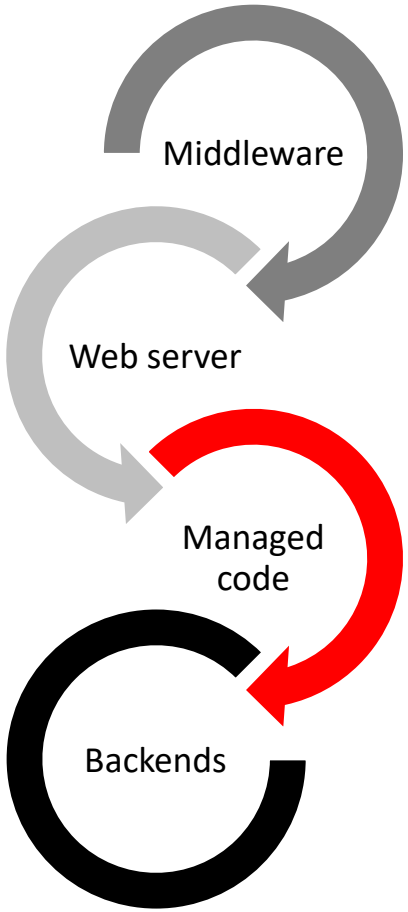
- Allow experience from previous tests to be reused
- A way to support pentesters. Don't start from scratch.
  - Your own refined Google / Hacktricks.xyz / etc.
- Not training on concepts, but simple bullets of what needs to be done
- Make pentester accountable to:
  - Check the things which needs to be checked
  - Ask team for help when there are interesting anomalies
- There are application and technology specific MVP's



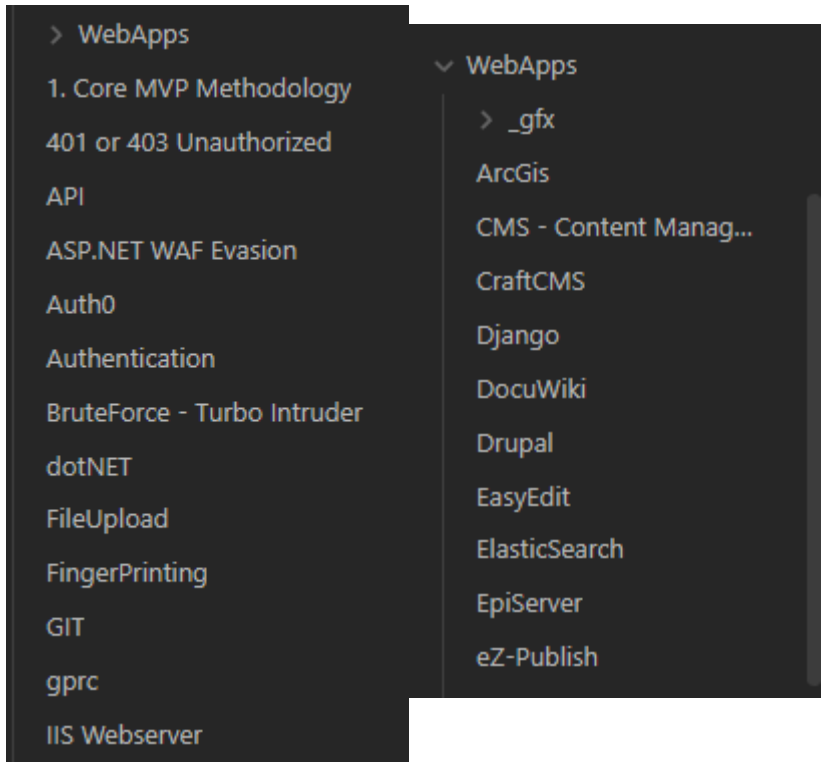
Tech and Application Specific MVP

Frameworks

Attack The Stack



Tech & App Specific MVP



Testing Frameworks

- ASVS – Application Security Verification Standard
- WSTG – Web Security Testing Guide
- …

WordPress Enumeration

```
#USERS
Chris Dale,chris
Karina Aarland,karina
Krister Kvaavik,krister
Magnus Holst,magnus
silje,silje
#POSTS
```

```
https://riversecurity.eu/wordpress/wp-content/uploads/2021/08/20210729_175011.jpg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/06/f_logo_RGB-Blue_100.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/06/LI-Logo.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/06/1-year-growth-1.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/06/1-year-growth.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/06/image.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/06/New-Project.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/River-security-01.jpg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/overview-1.jpg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/ooda-3.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/banner-042-01.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/eye-white-red-transparent.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/ben-den-engelsen-htcQ7uAWzAo-unsplash.jpg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/yue-su-77z-0VJJj6g-unsplash.jpg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/niclas-moser-ew6Guk2mqTk-unsplash.jpg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/overview-1.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/overview.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/eye-black-red_in_middle.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/daniel-malikyar-F1leFzugQfM-unsplash-1.jpg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/Vegar.jpg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/meg-rs-2.jpg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/meg-rs-1.jpg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/05/meg-rs.jpg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/04/proaktive-reactive-1.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/04/proaktive-reactive.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/04/Farmer-1.jpg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/04/1516243355397.jpeg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/01/1516243355397.jpeg
https://riversecurity.eu/wordpress/wp-content/uploads/2021/03/secret.txt
https://riversecurity.eu/wordpress/wp-content/uploads/2021/03/tv2-exchange-2.png
https://riversecurity.eu/wordpress/wp-content/uploads/2021/03/tv2-exchange.png
```

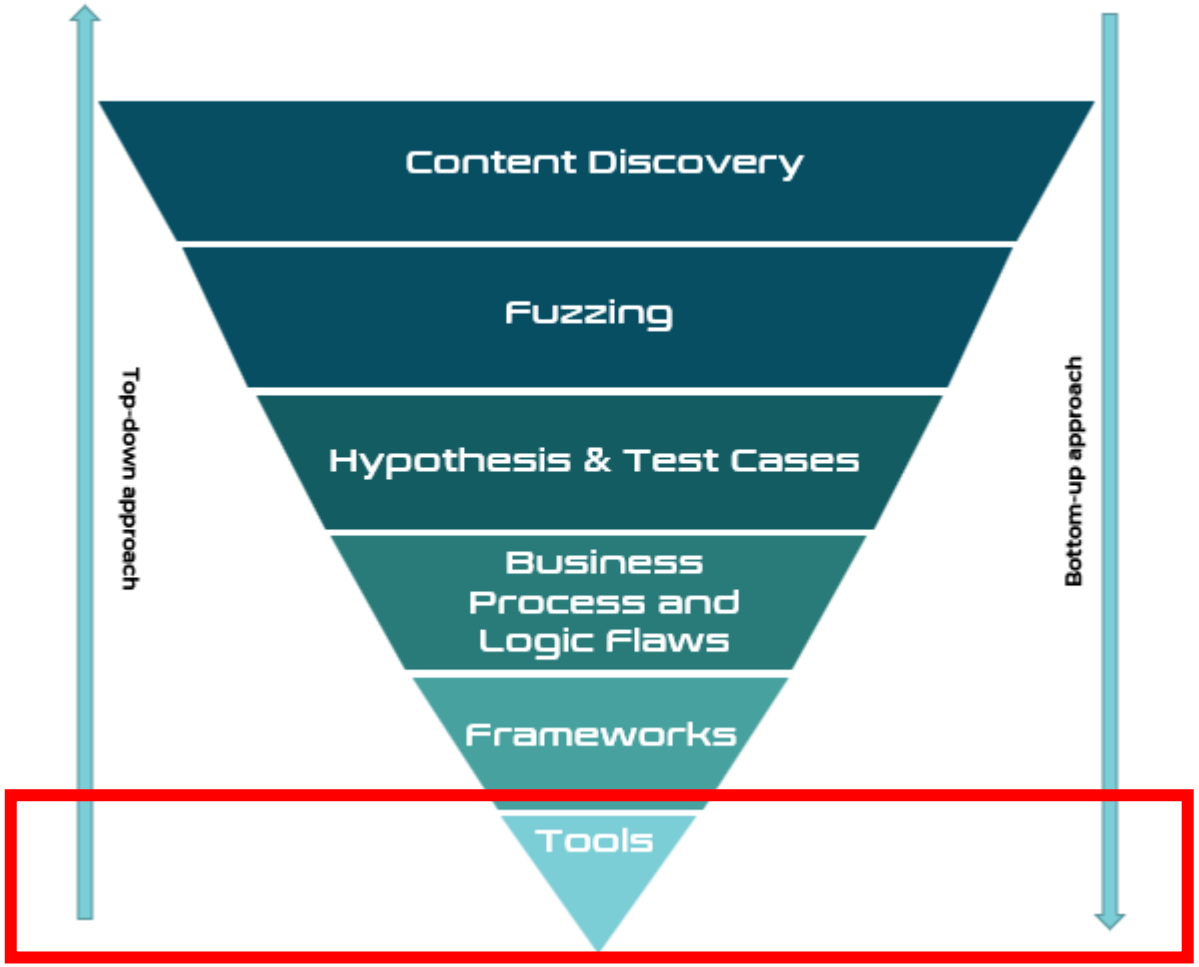
When You Don't Have MVP

Frameworks

- Create one
  - It is **minimum** viable
  - A starting point is better than nothing
- Dedicate days before the engagement to:
  - Build
  - Set-up
  - Configure
  - Break & Hack
  - Create CTF challenges ;)
- Create foundations for future hypothesis

Tools

Vulnerability scanners, application and technology specific tools



RIVER SECURITY

https://into.bio/chrisdale & https://into.bio/rivsec

Download slides here!

Twitter – https://twitter.com/ChrisADale

LinkedIn – https://www.linkedin.com/in/chrisad/

Fighting Cyber Crime – https://riversecurity.eu

💕 Work with us! We ARE hiring by attitude, and train for talents 🧠