

- 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

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



The ichor permeates MY FACE MY FACE oh god no NO NOO



Parsing HTML Using Regular Expressions

NO stop the an *les are not real ZALGO, HE COMES

O RLY?

DE Mon

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.

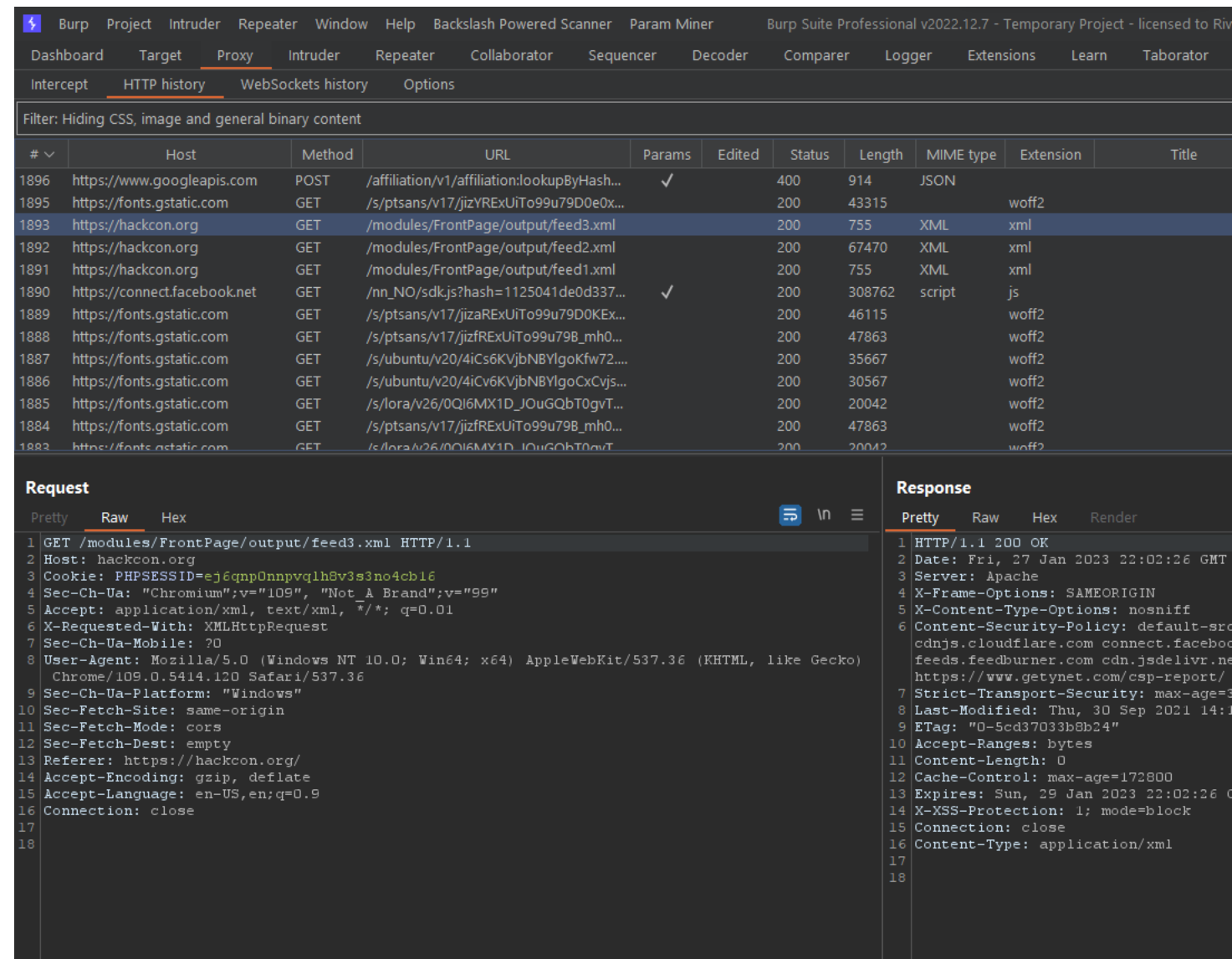
PORTSWIGGER TOP 10 ATTACKS

- 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



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 ++
- 🕒 Param Miner
- 🕒 Backslash Powered Scanner
- 🕒 Taborator

Nice to have

- 🕒 Turbo Intruder
- 🕒 Software Vulnerability Scanner
- 🕒 Authorize
- 🕒 Collaborator Everywhere

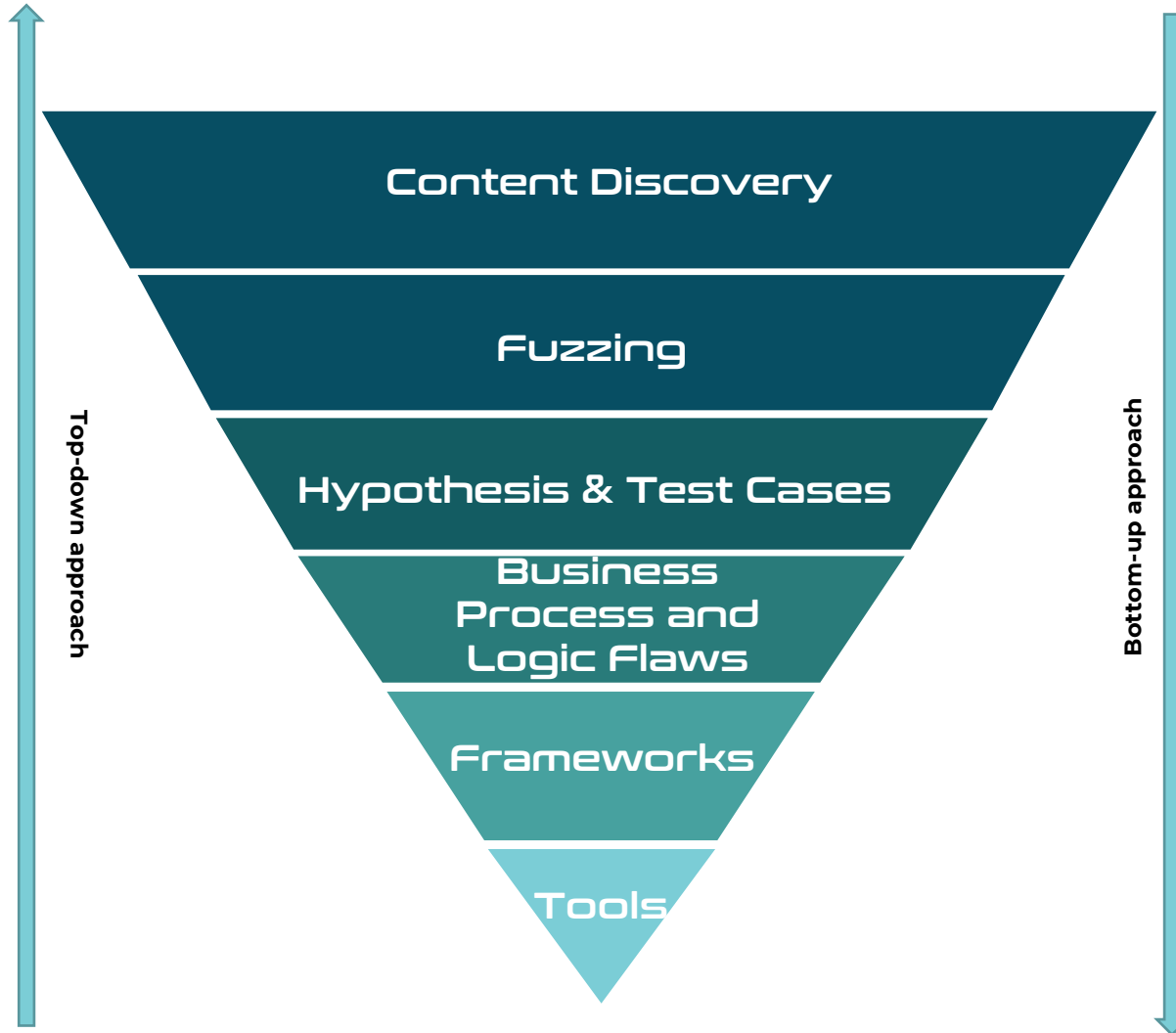
Honorable Mentions

- 🕒 Freddy, deserialization scanner
- 🕒 NTLM Challenge Decoder
- 🕒 GraphQL raider
- 🕒 Retire.js
- 🕒 JSON Web Tokens
- 🕒 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 individual's 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

Content Discovery

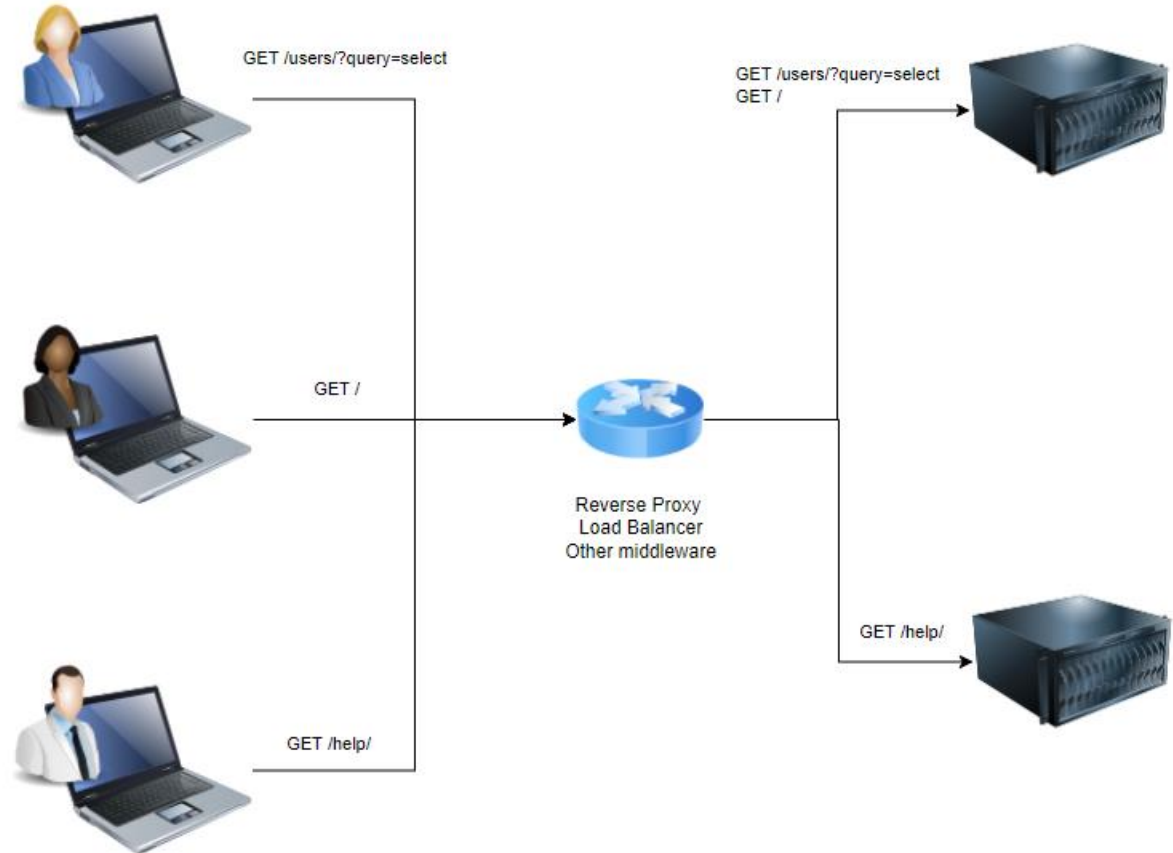
- i. Map Browsable Attack Surface**
- ii. Find Unlinked Content & Params**
- iii. Repeat for each `Platform Distinctions` of the application**

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

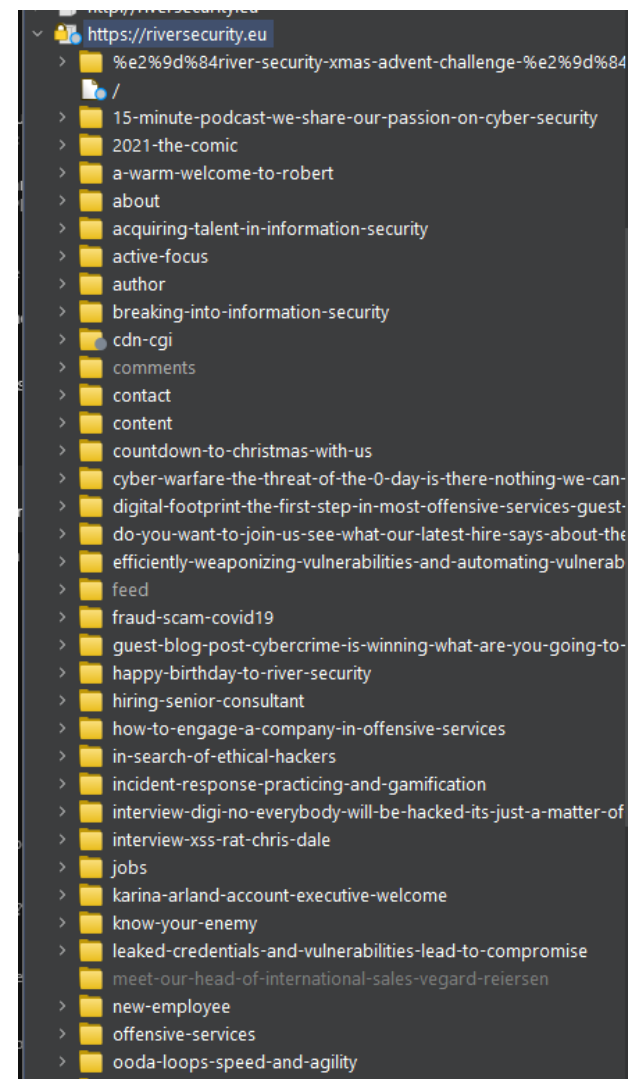


Content Discovery



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
```

```
chris@LAPTOP-2RRCM307:/mnt/d/riversec-repos/wordlists-discovery$ grep -aEirh '  
pdf  
pdf  
pdf bcd1  
pdf%20files  
pdf-32x32  
pdf-40x40.png.html  
pdf-accept  
pdf-analyser  
pdf-and-ppt-viewer  
pdf-annotation-zone.html  
pdf-api  
pdf-api2  
pdf-as-background.html  
pdf-as-image-landscape.html  
pdf-as-image.html  
pdf-as-tiled-background.html  
pdf-au  
pdf-base-footer.html  
pdf-base-header.html  
pdf-base.html  
pdf-behavior.html  
pdf-beta.html  
pdf-bg  
pdf-book.html  
pdf-books
```




Verb Example

/?page=872

Content Discovery

Attack Save Columns 26. Intruder attack of https://riversecurity.eu - Temporary attack - Not saved to project file

Results Positions Payloads Resource Pool Options

Filter: Showing all items

Request ^	Payload	Status	Error	Timeout	Length	Comment
34	login	200	<input type="checkbox"/>	<input type="checkbox"/>	58525	
35	search	200	<input type="checkbox"/>	<input type="checkbox"/>	58523	
36	content	200	<input type="checkbox"/>	<input type="checkbox"/>	58547	
37	comment	200	<input type="checkbox"/>	<input type="checkbox"/>	58541	
38	step	200	<input type="checkbox"/>	<input type="checkbox"/>	58527	
39	ajax	200	<input type="checkbox"/>	<input type="checkbox"/>	58521	
40	debug	200	<input type="checkbox"/>	<input type="checkbox"/>	58523	
41	state	200	<input type="checkbox"/>	<input type="checkbox"/>	58457	
42	query	200	<input type="checkbox"/>	<input type="checkbox"/>	58521	
43	f	200	<input type="checkbox"/>	<input type="checkbox"/>	58521	
44	error	200	<input type="checkbox"/>	<input type="checkbox"/>	58517	
45	save	200	<input type="checkbox"/>	<input type="checkbox"/>	58521	
46	sort	200	<input type="checkbox"/>	<input type="checkbox"/>	58525	
47	format	200	<input type="checkbox"/>	<input type="checkbox"/>	58457	
48	tab	200	<input type="checkbox"/>	<input type="checkbox"/>	58523	
49	offset	200	<input type="checkbox"/>	<input type="checkbox"/>	58529	
50	edit	200	<input type="checkbox"/>	<input type="checkbox"/>	58523	

Request Response

Pretty Raw Hex

```
1 GET /?debugPage=872 HTTP/2
2 Host: riversecurity.eu
3 Accept-Encoding: gzip, deflate
4 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-excha
```



Content Discovery



Content Discovery

Content discovery: vg.no

Control **Config** Site map

Target

Define the start directory for the content discovery session, and whether files or directories should be targeted.

Start directory:

Discover:

- Files and directories
- Files only
- Directories only

Recurse subdirectories

Max depth:

Filenames

Configure the sources Burp should use for generating filenames to test.

- Built-in short file list
- Built-in short directory list
- Built-in long file list
- Built-in long directory list
- Custom file list:
 - Custom directory list:
 - Names observed in use on target site
- Derivations based on discovered items

File Extensions

These settings control how the discovery session adds file extensions to file stems that are listed in the target site's directory listing.

- Test these extensions:
- Test all extensions observed in use on target site, except for:
- Test these variant extensions on discovered files:
- Test file stems with no extension



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

The screenshot displays the SwaggerHub interface for an OpenAPI specification. The top navigation bar includes the SwaggerHub logo, the user name 'RPinkham23', and a dropdown menu. Below the navigation bar, the API title 'TradeshowD... | SamplePets...' is shown along with the version '1.0.0' and the format 'OAS3'. The interface is divided into three tabs: 'Editor', 'Split', and 'UI', with 'Editor' currently selected. A status bar indicates 'Last Saved: 12:46:55 pm May 4, 2018' and 'VALID'. The main content area shows the API endpoint 'https://virtserver.swaggerhub.com/TradeshowDemos/SamplePetstoreAPI/1.0.0' and a 'Show Comments' button. The API is titled 'pet' with the description 'Everything about your Pets'. A list of endpoints is displayed, each with a colored header indicating the HTTP method: POST for adding a new pet, PUT for updating an existing pet, GET for finding pets by status or ID, and DELETE for deleting a pet. The 'store' endpoint is also visible at the bottom.

Method	Endpoint	Description
POST	/pet	Add a new pet to the store
PUT	/pet	Update an existing pet
GET	/pet/findByStatus	Finds Pets by status
GET	/pet/findByTags	Finds Pets by tags
GET	/pet/{petId}	Find pet by ID
POST	/pet/{petId}	Updates a pet in the store with form data
DELETE	/pet/{petId}	Deletes a pet
POST	/pet/{petId}/uploadImage	uploads an image



Unlinked Parameters

- 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!**

Content Discovery

#	Task	Time	Action	Issue type	Ho
225	0	23:48:45 3 Feb 2023	Issue found	Secret input: url	https://riverse
224	0	23:48:34 3 Feb 2023	Issue found	Secret input: url	https://riverse
223	0	23:48:33 3 Feb 2023	Issue found	Secret input: url	https://riverse
222	0	23:48:16 3 Feb 2023	Issue found	Secret input: url	https://riverse

Advisory	Request 1	Response 1	Request 2	Response 2
----------	-----------	------------	-----------	------------

Secret input: header Compare responses

Issue: **Secret input: header**
Severity: **Medium**
Confidence: **Firm**
Host: **https://riversecurity.eu**
Path: **/**

Note: This issue was generated by a Burp extension.

Issue detail
Unlinked parameter identified.

Successful probes

Found unlinked param: x-requested-with	x-requested-with	x-requested-withpevpfq
tag_names	X	Y
word_count	2910	2975
<script	22	23
content_length	X	*Y*
limited_body_content	X	*Y*



- WaybackRobots.py

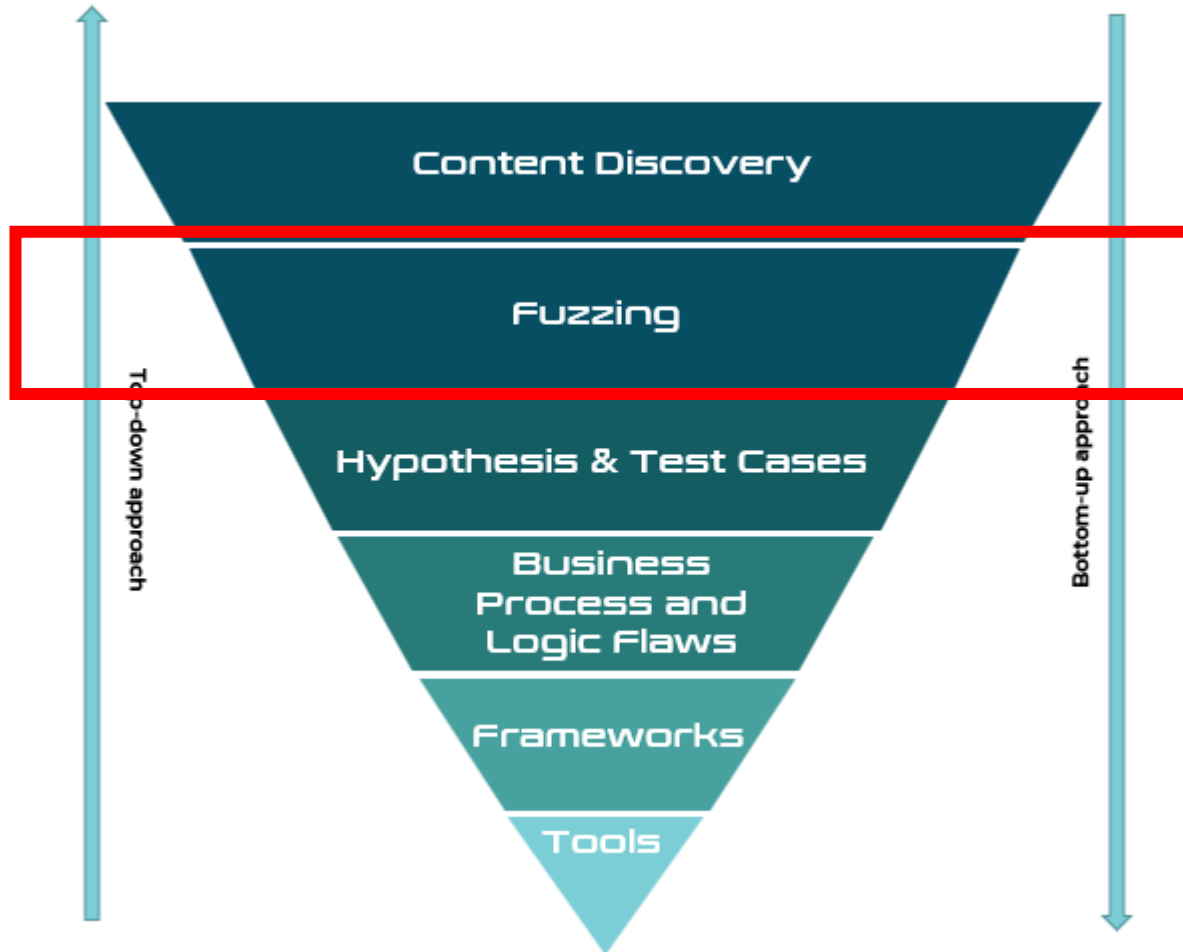
- WaybackURLs.py

The screenshot shows the HackCon website interface. At the top left is the 'HackCon' logo in red and blue. To the right, a yellow banner reads '54 dager igjen'. Below the logo is a blue navigation bar with links: 'Forside', 'Om HackCon', 'Program HackCon#1', 'Påmelding HackCon#1', and 'Kontakt/Kontakt skjema'. A left sidebar contains a list of menu items with plus signs: 'Om HackCon', 'Program HackCon#1', 'Påmelding HackCon#1', 'Administrativ informasjon', 'Konkurranser', 'Foredragsholder?', 'Om oss', 'Sponsorer', 'For media', 'Forum', 'Kontakt/Kontakt skjema', and 'Driftsmeldinger'. The main content area features the text: 'KINS (Foreningen Kommunal informasjonssikkerhet), Kompetanse- og utviklingsnettverket og HoneyNor arrangerer: HackCon#1 i Oslo den 8. og 9. februar 2006'. To the right, a yellow box states 'HackCon#1 arrangeres i Oslo 8. og 9. februar 2006'. Below this, a red box says 'Antall plasser er begrenset Fulltegning pr. 16/12: 88%'. Further down, it says 'Vi støtter nettsidene' followed by 'www.nettguiden.info' and a description: 'Bli med på å lage Norges største erfaringsdatabase om produkter og firma i Norge.' At the bottom, a footer reads: 'For åpen, fri og objektiv informasjonsflyt Better safe than sorry Denne siden ble generert av publiseringssystemet Pub:IT.Net på 0,25 sekunder'.



Fuzzing

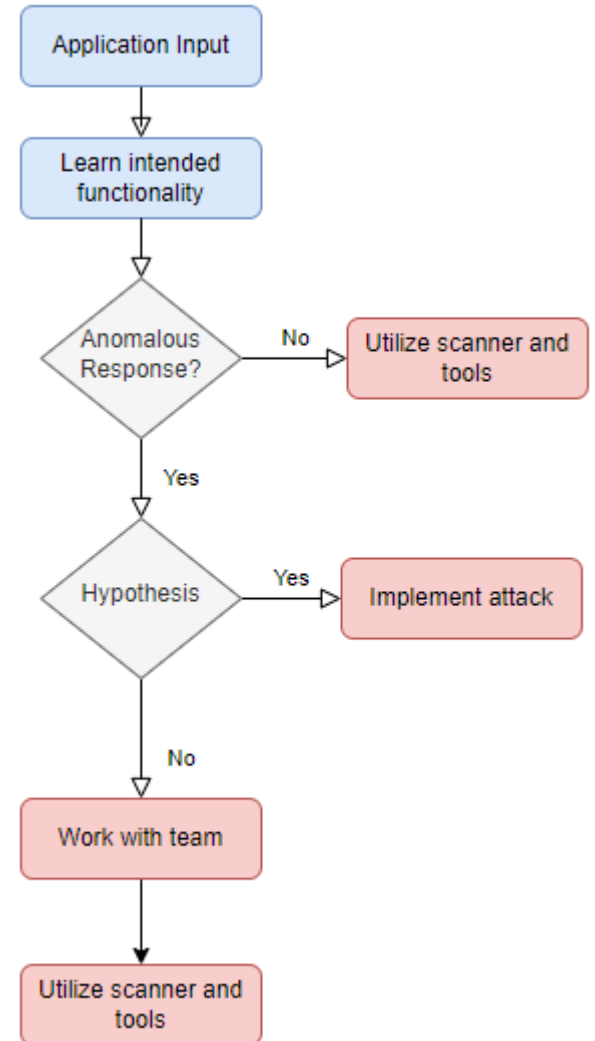
Find bytes and input producing different/unexpected results

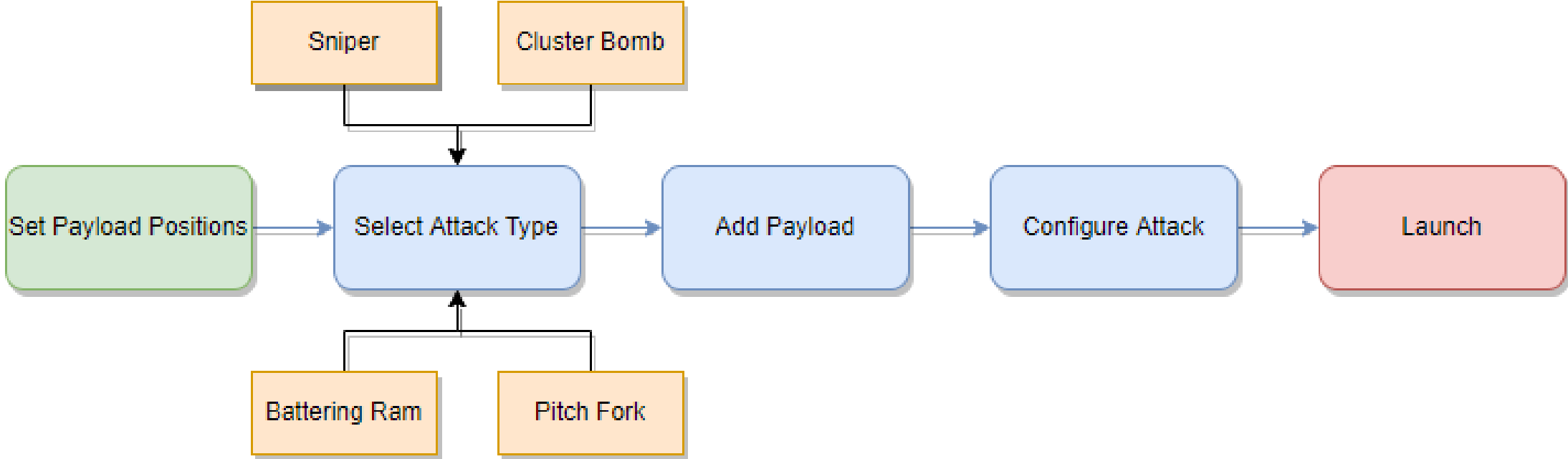




Fuzzing Bytes 101

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.aspx 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
```

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 [REDACTED]
6 X-AspNetMvc-Version: 5.2 [REDACTED]
7 Set-Cookie: sessi [REDACTED] path=/; secure; SameSite=None
8 X-Content-Type-Options: nosniff [REDACTED]
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 [REDACTED]
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>
21     <title>
22
23     </title>
24     <link href="css/default.css" rel="stylesheet" type="text/css" />
  </head>
  <body>
  <form name="form1" method="post" action="[REDACTED]" id="form1">
```

0 matches

0 matches



Bytes Examples

Fuzzing

Payload here



Request	Response
1 GET /asdf.aspx HTTP/2	
2 Ho	
3 Co	
54	
4 Sec-Ch-Ua: "Not?A_Brand";v="8", "Chromium";v="108"	
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/108.0.5359.125 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 Connection: close	
17	
18	



Attack Save Columns

Results Positions Payloads Resource Pool Options

Filter: Hiding 4xx responses

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

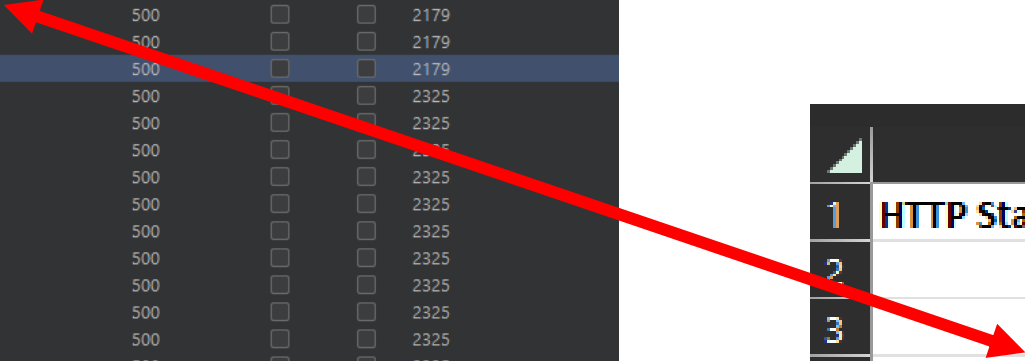
Request Response

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' http
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>
18 <title>
19 </title>
  <link href="css/default.css" rel="stylesheet" type="text/css" />
  </head>
```

Fuzzing

	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

Fuzzing

Dashboard Target Proxy **Intruder** Repeater Collaborator Sequencer Decoder

1 x 2 x 3 x 4 x **5 x** +

Positions **Payloads** Resource Pool Options

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type and the number of positions. The number of payload sets can be customized in different ways.

Payload set: 1 Payload count: 256
Payload type: Simple list Request count: 256

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste	%00
Load ...	%01
Remove	%02
Clear	%03
Deduplicate	%04
	%05
	%06
Add	Enter a new item
Add from list ...	

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Enabled	Rule
<input checked="" type="checkbox"/>	URL-decode

Payload Encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission.

URL-encode these characters:

Request	Payload	Status	Error	Timeout	Length	Comment
85	T	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
86	U	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
87	V	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
88	W	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
89	X	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
90	Y	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
91	Z	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
92	[200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
93	\	200	<input type="checkbox"/>	<input type="checkbox"/>	1263	
94]	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
95	^	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
96	_	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
97	`	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
98	a	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
99	b	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	
100	c	200	<input type="checkbox"/>	<input type="checkbox"/>	1101	

Request Response

Pretty Raw Hex Render

```
var wechallinfo = {
  "level": "natas14", "pass": "qPazSJBmrmU7UQJv17MHk1PGC4DxZMEP"
};
</script>
</head>
<body>
  <h1>
    natas14
  </h1>
  <div id="content">
    <br />
    <b>
      Warning
    </b>
    : mysqli_num_rows() expects parameter 1 to be mysqli_result, bool given in <b>
      /var/www/natas/natas14/index.php
    </b>
    on line <b>
      24
    </b>
  <br />
  Access denied!<br>
  <div id="viewsource">
    <a href="index-source.html">
      View sourcecode
    </a>
  </div>
</div>
</body>
</html>
```

0 matches

Finished



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

```
file-ul-filter-bypass-microsoft-asp-PH-UE.txt
file-ul-filter-bypass-ms-php.txt
file-ul-filter-bypass-x-platform-generic-UE.txt
file-ul-filter-bypass-x-platform-php-PH.txt
```



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



Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extender Project options User options GAP

Select param types you want to retrieve:

REQUEST PARAMETERS

- Query string params
- Message body params
- Param attribute within a multi-part message body
- JSON params
- Cookie names
- Items of data within an XML structure
- Value of tag attributes within XML structure

RESPONSE PARAMETERS

- JSON params
- Value of tag attributes within XML structure
- Name and Id attributes of HTML input fields
- Javascript variables and constants
- Name attribute of Meta tags
- Params from links found

GAP Mode: Parameters Links ?

Output options:

- Include the list of common params in list (e.g. used for redirects)?
- Build concatenated query string with param value: XNLV
- Include URL path words in parameter list?
- Include site map endpoints in link list?
- Auto save output to directory: C:\BugBounty Choose...

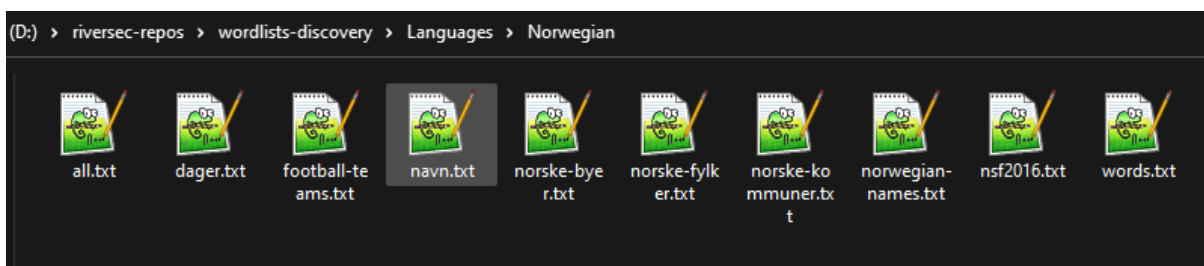
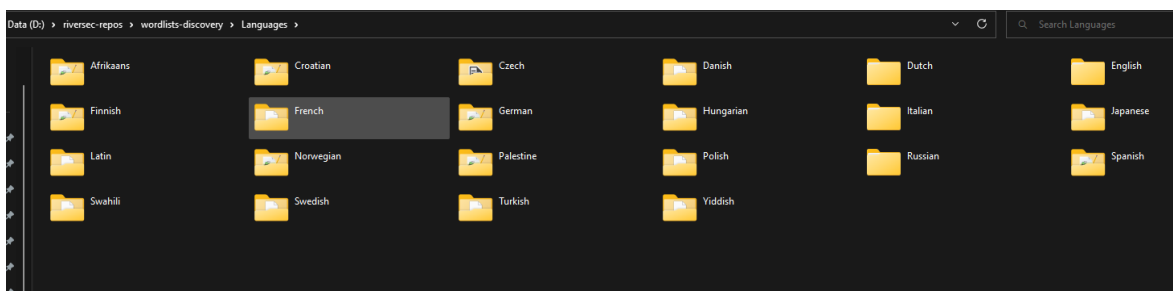
Restore defaults Save options COMPLETED

The latest generated query string of all parameters:

```
RelayState=XNLV0&active=XNLV1&admin=XNLV2&callback=XNLV3&cancelURL=XNLV4&cancelUrl=XNLV5&cancel_url=XNLV6&debug=XNLV7&dest=XNLV8&destination=XNLV9&forward=XNLV10&forward_url=XNLV11&forwardurl=XNLV12&go=XNLV13&goto=XNLV14&goto=XNLV15&id=XNLV16&location=XNLV17&locationURL=XNLV18&locationUrl=XNLV19&locationurl=XNLV20&n=XNLV21&next=XNLV22&out=XNLV23&page=XNLV24&prev=XNLV25&previus=XNLV26&r_URL=XNLV27&r_Url=XNLV28&r_Url=XNLV29&redirect=XNLV30&redirect=XNLV31
```

Link filter: Negative match Case sensitive Apply filter

Link exclusions: .css .jpg .jpeg .png .svg .img .gif .mp4 .flv .ogv .webm .webp .mov .mp3 .m4a .m4p .scss .tif .tiff .tff .otf .woff .woff2 .bmp .ico .eot .htc .rtf .swf .image.w3.org.doubleclick.net.youtube.com



```
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.*$" | tr '[:upper:]' '[:lower:]' | sort | uniq > /mnt/d/tmp/compan.txt
grep -aEirh "^daily.*$" | tr '[:upper:]' '[:lower:]' | sort | uniq > /mnt/d/tmp/daily.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

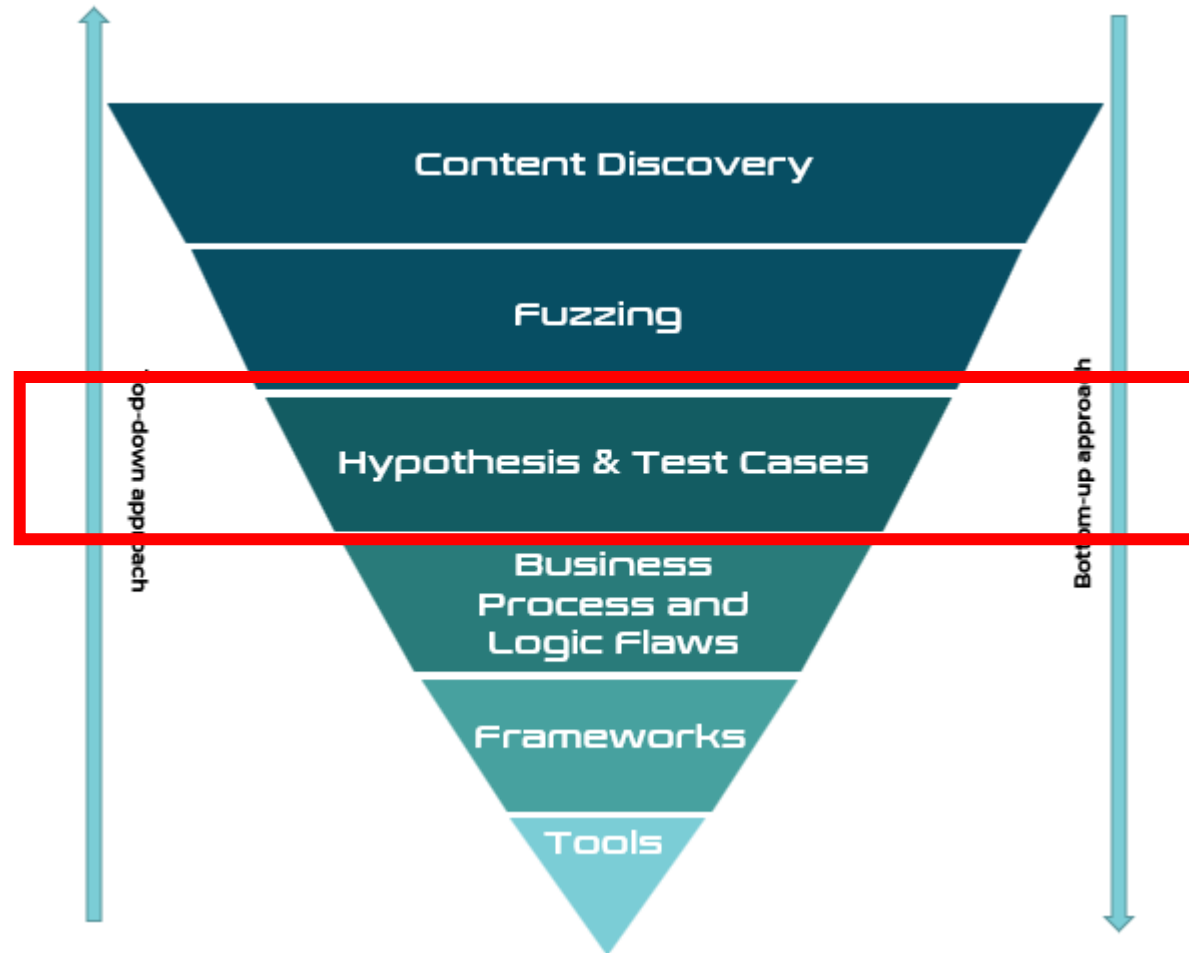
```
PS C:\tmp\repos\IIS_shortname_Scanner> C:\Python27\python.exe .\iis_shortname_Scan.py https://[redacted]/metadatacard/
Server is vulnerable, please wait, scanning...
[+] /metadatacard/m~1.* [scan in progress]
[+] /metadatacard/me~1.* [scan in progress]
[+] /metadatacard/met~1.* [scan in progress]
[+] /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

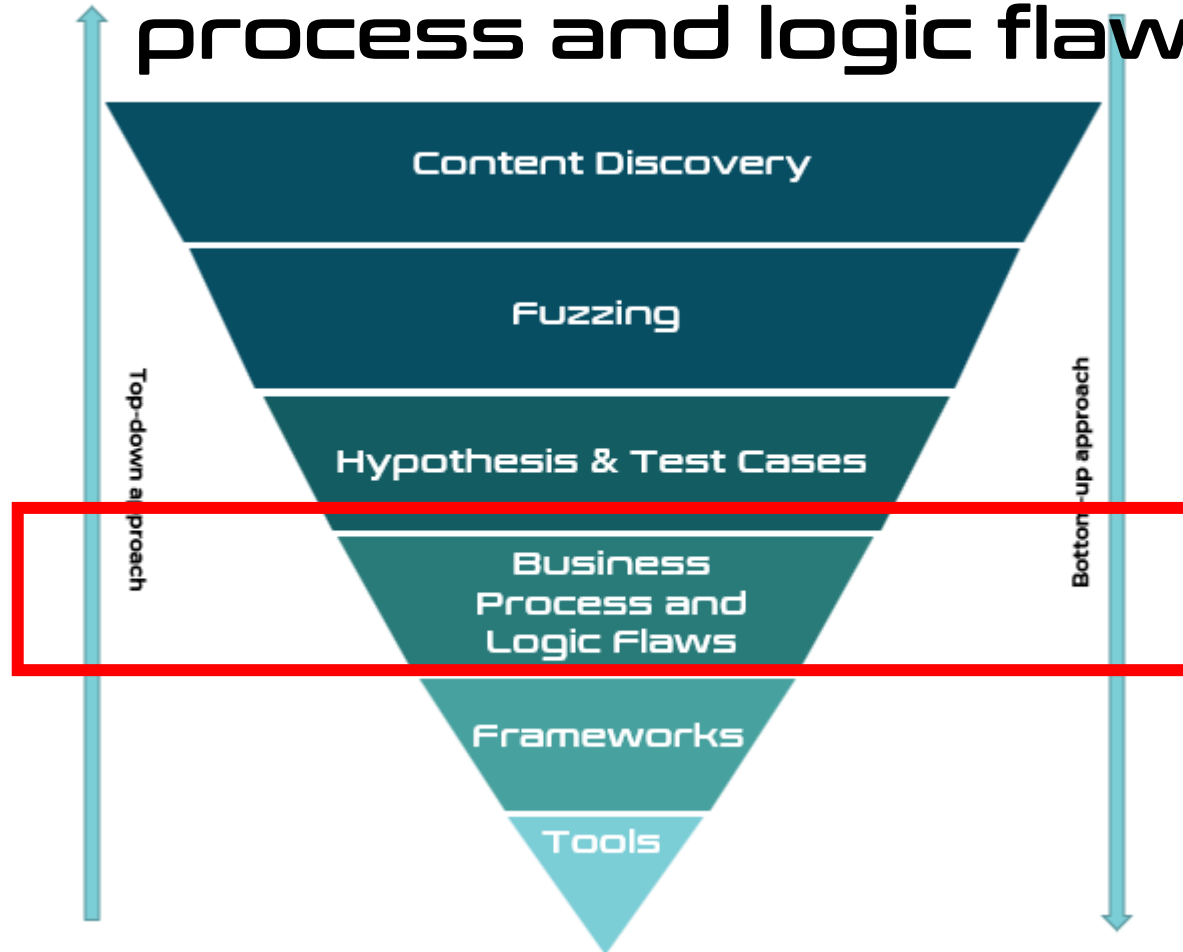
I am seeing that : > < and * are influencing file reads of the file server. I want to explore Local File Inclusion, SSRF and similar kinds of vulnerabilities

	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	Wildcard
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

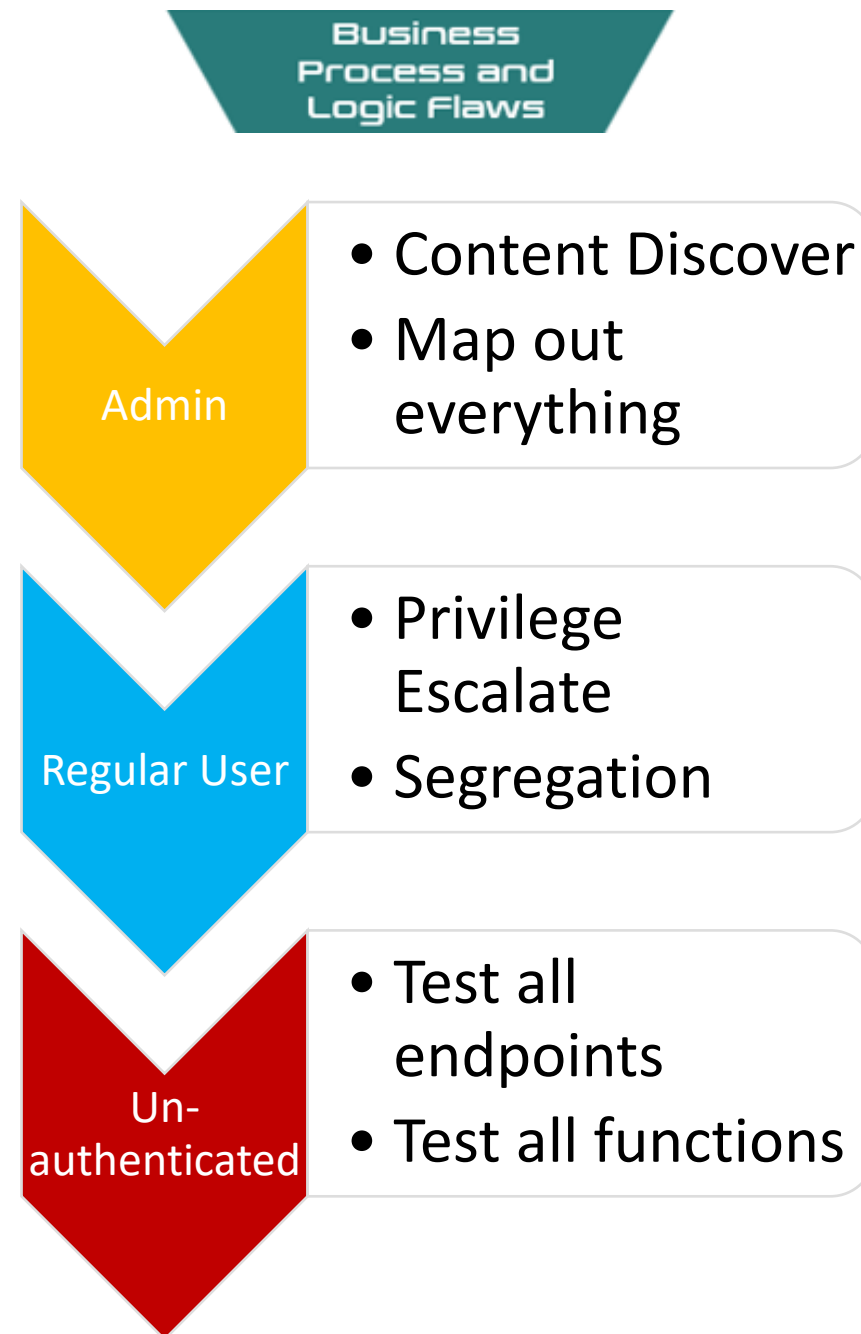
With extensive knowledge of the target, explore process and logic flaws





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

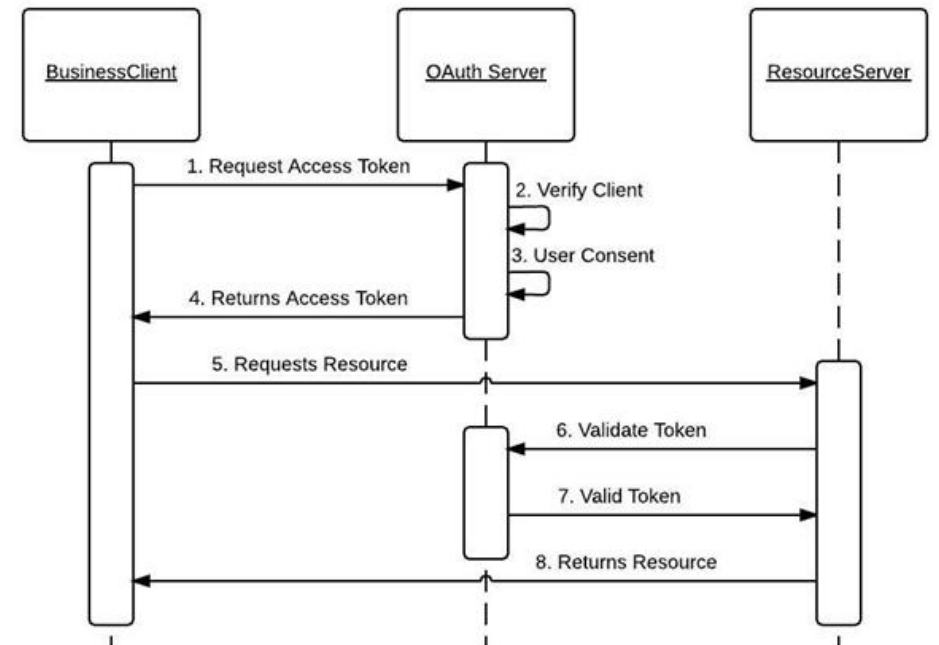




Map Out Application Flows

Business
Process and
Logic Flaws

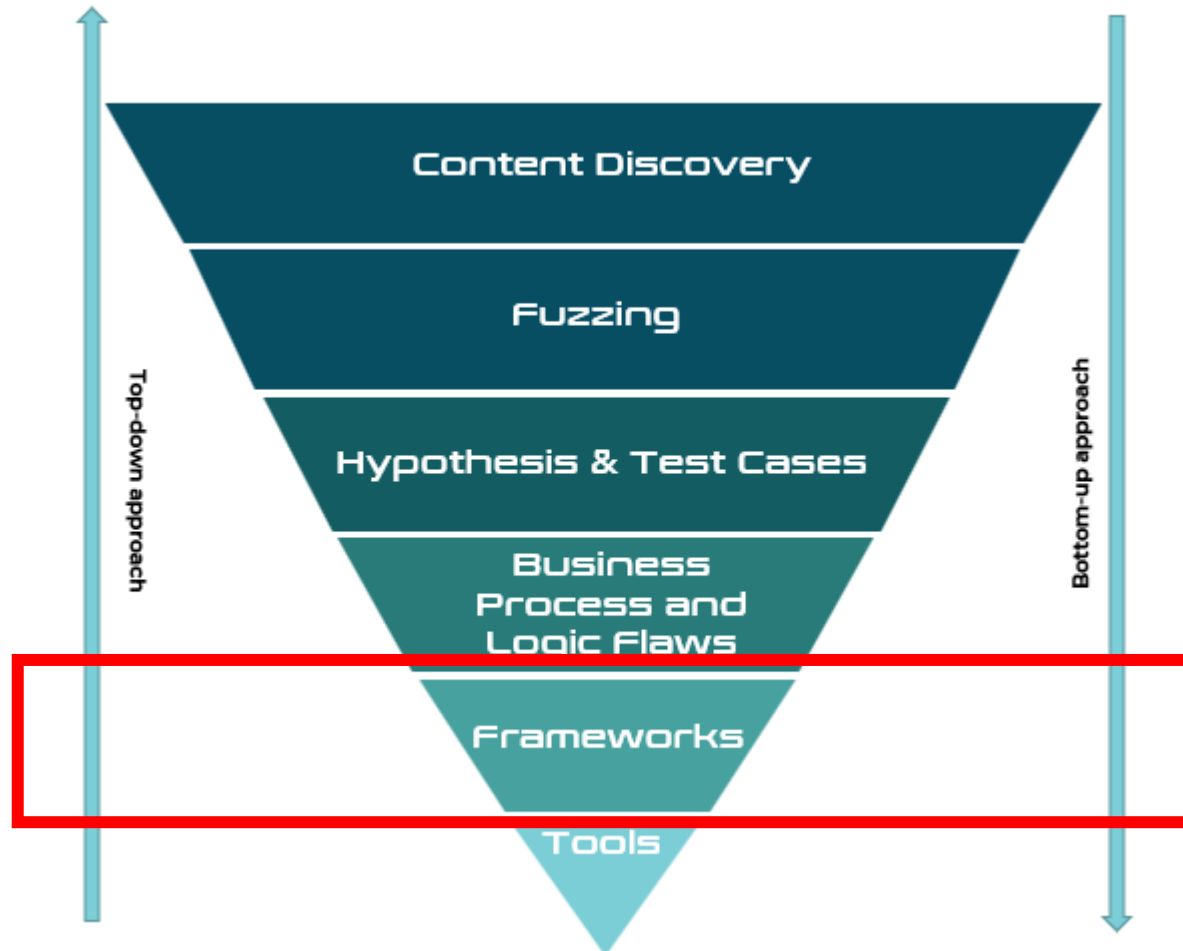
- Mapping out the flow of behavior
- Draw.io / Diagrams.net is easy quick win
- Helps look at things from a bird eye perspective
- Map out requests and response
- Example flows:
 - Purchasing
 - Authentication
 - Impersonation / privilege escalation
 - Password reset flow
 - ...





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

Frameworks

```

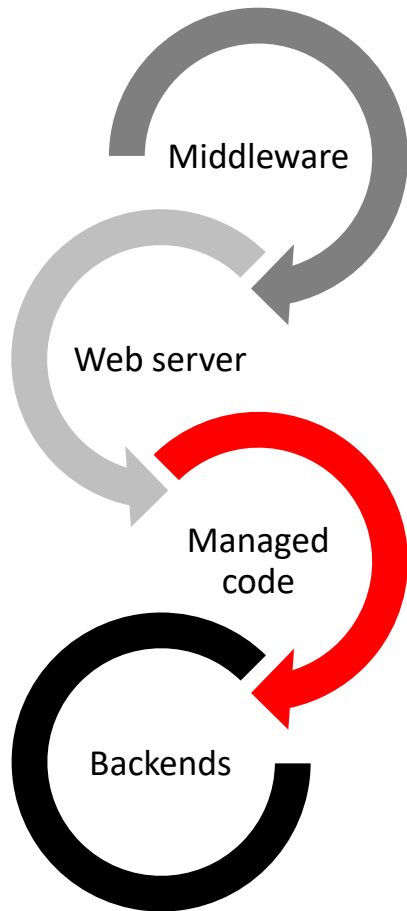
v Minimum Viable Pentesting
  > Cloud
  > Hardware
  > Internal
  > Mobile
  > Other Services
  > Phishing
  v WEB
    > _gfx
    > Tools
    > WebApps
    1. Core MVP Methodology
    401 or 403 Unauthorized
    API
    ASP.NET WAF Evasion
    Auth0
    Authentication
```



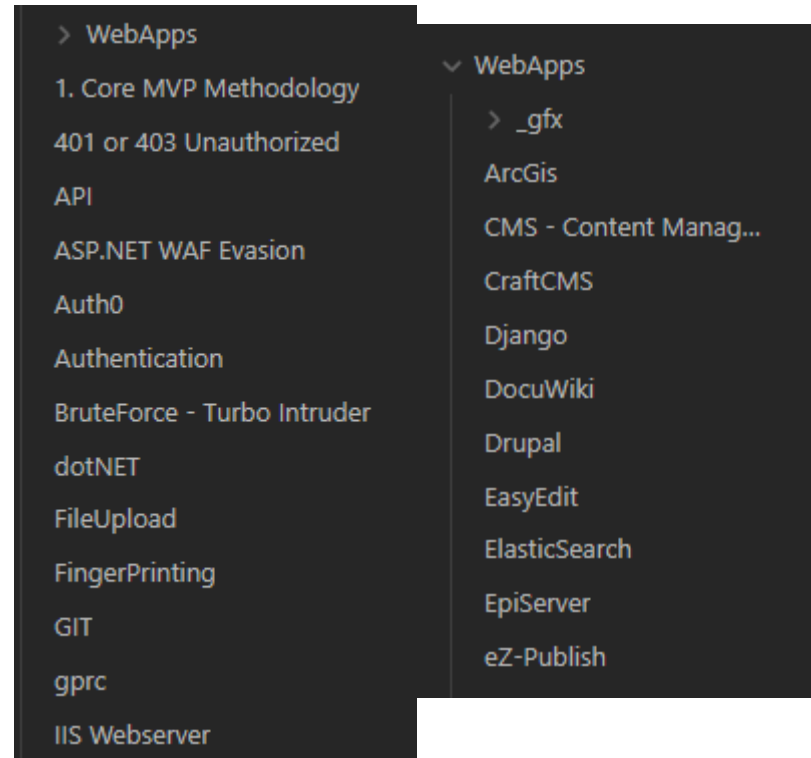
Tech and Application Specific MVP



Attack The Stack



Tech & App Specific MVP



Testing Frameworks

- 🕒 ASVS – Application Security Verification Standard
- 🕒 WSTG – Web Security Testing Guide
- 🕒 ...

WordPress Enumeration

```
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-FileFzugQfM-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
```

#USERS

Chris Dale, chris
Karina Aarland, karina
Krister Kvaavik, krister
Magnus Holst, magnus
silje, silje

#POSTS



When You Don't Have MVP



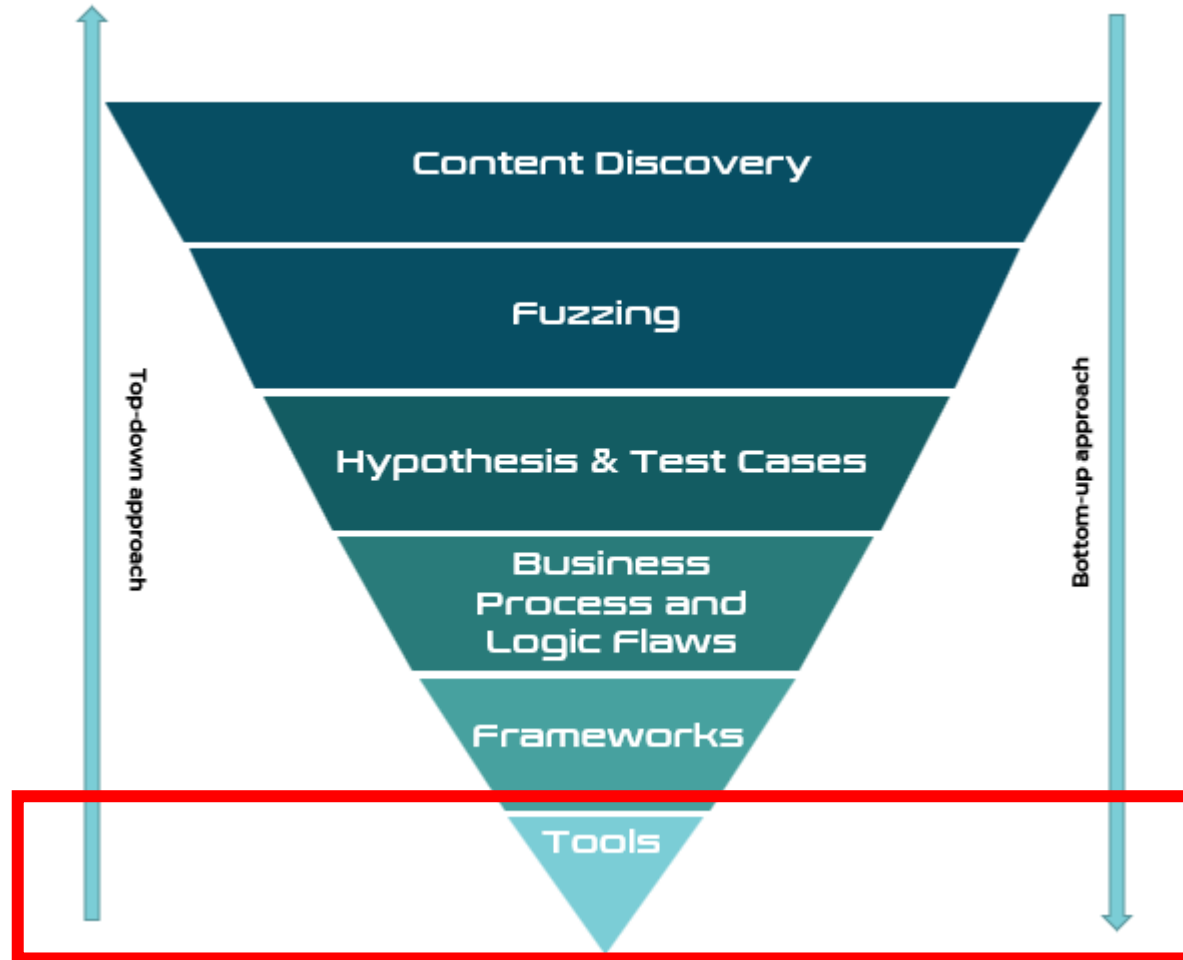
- Create one
 - It is **minimum** viable
 - A starting point is better than nothing
- Dedicate days before the engagement to:
 - Build
 - Set-up
 - Configure
 - Break & Hack
 - Create CTF challenges ;)
- Create foundations for future hypothesis





Tools

Vulnerability scanners, application and technology specific tools





<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 🗣️