



Pentest Methodology

A reform is needed



CHRIS DALE

- CHO AT RIVER SECURITY
- PRINCIPAL INSTRUCTOR AT SANS
- EX CISO

SHORT WHOAMI:

**I SHOW HOW CRIMINALS BREAK-IN,
AND I HELP THROW THEM BACK OUT...**

CERTS

- GCIH** GIAC Certified Incident Handler
- GPEN** GIAC Certified Penetration Tester
- GSLC** GIAC Security Leadership
- GIAC** GIAC Mobile Device Security Analyst
- GDAT** GIAC Defending Advanced Adversaries
- GCTI** GIAC Cyber Threat Intelligence
- GCFA** GIAC Certified Forensic Analyst
- GXIH** GIAC Experienced Incident Handler
- GXPT** GIAC Experience Penetration Tester
- GSP** GIAC Security Professional



Provider in the Offensive Security Space

- And a touch of Incident Response
- 20 employees
- Boot strapped
- 4 years in business
- 41+ public customer testimonials
- 8+ customer cases





Provider in the Offensive Security Space

- And a touch of Incident Response
- 20 employees
- Boot strapped
- 4 years in business
- 41+ public customer testimonials
- 8+ customer cases



50+ Companies in Norway providing pentesting



Ahmad A. • 1st

Fagansvarlig Offensive Security - Author - Speaker - Content Creator - Nerd

2w • Edited •

50+ selskaper som tilbyr sikkerhetstesting (pentest) i Norge

1. Experis AS
2. River Security AS
3. Netsecurity AS
4. Kovert
5. Binary Security AS
6. Watchcom (nå Telenor Cyberdefence)
7. EVERY AS
8. Semaphore
9. Atea AS
10. Aztek AS (nå Accelerate at Iver)
11. Promon AS
12. Mnemonic AS
13. Fencenordic
14. Secure-NOK AS
15. Itera Consulting Norway AS
16. NorSIS
17. Horangi Cyber Security
18. PwC Norge
19. Capgemini Norge AS
20. Accenture AS

21. Motit AS
22. Visma Consulting AS (nå Twoday)
23. KPMG Norge
24. Defendable AS
25. Teknograd AS
26. Secure Practice AS
27. Greenberg Traurig Norway AS
28. DataGardens AS
29. Nixu / DNV
30. Sopra Steria AS
31. Encripto AS
32. EY Norge
33. Banshie
34. BDO AS
35. Orange Cyberdefense
36. Agenda Risk AS
37. Knowit Secure AS
38. CyberIntelsys
39. CGI Norge
40. Mily
41. ShieldTech AS
42. NetNordic Norway
43. NORMA Cyber
44. Konfitech AS
45. Capra Consulting AS

Let us take
a look at



Pentest
Methodologies

All

Images

Videos

Short videos

Forums

News

Web

More

Tools

Vulnerability assessment

Ossstmm

External

Standard

Nist

Mobile application security testing

Security flaws

Ec council

Web application penetration

Software testing

PENETRATION TESTING STAGES

- 01 Planning and reconnaissance: Test goals are defined and intelligence is gathered.
- 02 Scanning: Scanning tools are used to understand how a target responds to intrusions.
- 03 Gaining access: Web application attacks are staged to uncover a target's vulnerabilities.
- 04 Maintaining access: APTs are initiated to see if a vulnerability can be used to maintain access.
- 05 Analysis and WAF configuration: Results are used to configure WAF settings before testing is run again.

Imperva
What is Penetration Testing | Step-By-Step Process & Methods | Imperva



Cobalt
Web Application Penetration Testing | Step-By-Step Process & Methods | Cobalt



Aress Software
Top 5 Penetration Testing Methodology | Aress Software

- Information Gathering
- Analysis and Planning
- Vulnerability Identification

Top 5 Penetration Testing Methodologies and Standards

- 1 OWASP: Open Web Application Security Project
- 2 NIST: National Institute of Standards and Technology
- 3 PTES: Penetration Testing Execution Standard
- 4 ISSAF: Information System Security Assessment Framework
- 5 OSSTMM: Open-Source Security Testing Methodology Manual

Astra Security
Top 5 Penetration Testing Methodology | Astra Security



TechTarget
Pen testing guide: Types, Steps, and Best Practices | TechTarget



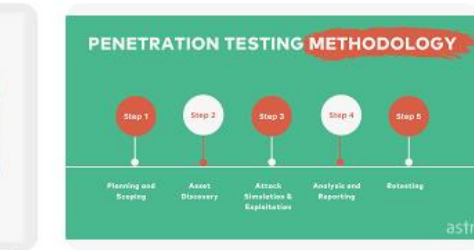
eLuminous Technologies
Web Application Penetration Testing | eLuminous Technologies



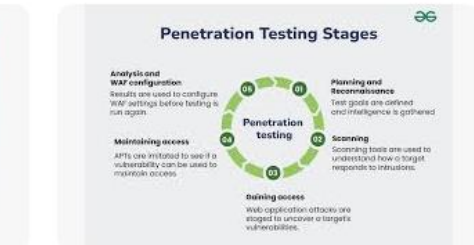
The Different Stages in Penetration Testing Methodology Are:

- Pre-engagement and Planning
- Intelligence Gathering
- Vulnerability Analysis & Exploitation
- Post-Exploitation (Remediation)
- Reporting & Certification

Sprinto
Top 5 Penetration Testing Methodology | Sprinto



Astra Security
NIST Penetration Testing: A Step-By-Step Process & Methods | Astra Security



GeeksforGeeks
Penetration Testing - Software Engineering | GeeksforGeeks



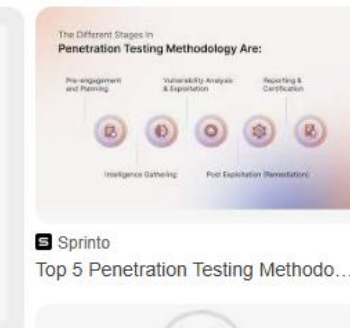
PENETRATION TESTING STAGES

- 01 Planning and reconnaissance: Test goals are defined and intelligence is gathered.
- 02 Scanning: Scanning tools are used to understand how a target responds to intrusions.
- 03 Gaining access: Web application attacks are staged to uncover a target's vulnerabilities.
- 04 Maintaining access: APTs are initiated to see if a vulnerability can be used to maintain access.
- 05 Analysis and WAF configuration: Results are used to configure WAF settings before testing is run again.

Imperva
What is Penetration Testing | Step-By-Step Process & Methods | Imperva

Images may be subject to copyright. Learn More

Share Save



im Imperva

Analysis and WAF configuration
Results are used to configure WAF settings before testing is run again.

Maintaining access
APTs are imitated to see if a vulnerability can be used to maintain access.

Planning and reconnaissance
Test goals are defined and intelligence is gathered.

Scanning
Scanning tools are used to understand how a target responds to intrusions.

Gaining access
Web application attacks are staged to uncover a target's vulnerabilities.

PENETRATION TESTING STAGES

01 02 03 04 05

What is Penetration Testing | Step-By-Step Process & Methods | Imperva

Images may be subject to copyright. [Learn More](#)

[Share](#) [Save](#) [Visit >](#)

Sprinto

The Different Stages In
Penetration Testing Methodology Are:

Pre-engagement and Planning
Intelligence Gathering

Vulnerability Analysis & Exploitation

Reporting & Certification
Post Exploitation (Remediation)

1,024 x 675

Top 5 Penetration Testing Methodology to Follow in 2025

Images may be subject to copyright. [Learn More](#)

[Share](#) [Save](#) [Visit >](#)

IG IGspectrum

Network Penetration Testing Methodology

Initial Scoping & Objectives Agreement

Reconnaissance

Scanning

Exploitation

Reporting

Remediation

Penetration Testing - IGspectrum Limited

subject to copyright. [Learn More](#)

[Share](#) [Save](#) [Visit >](#)

Security Metrics

6 Steps of Penetration Testing

1 Pre-Engagement Interactions

2 Reconnaissance

3 Vulnerability Identification

4 Exploitation

5 Post Exploitation

6 Reporting

6 Steps to a Penetration Test

Images may be subject to copyright. [Learn More](#)

[Share](#) [Save](#) [Visit >](#)

can you list out the phases of a pentest?

Certainly — here's a solid breakdown of the classic pentest phases, which you can adjust depending on your engagement style (goal-based, red teaming, assumed breach, etc.):

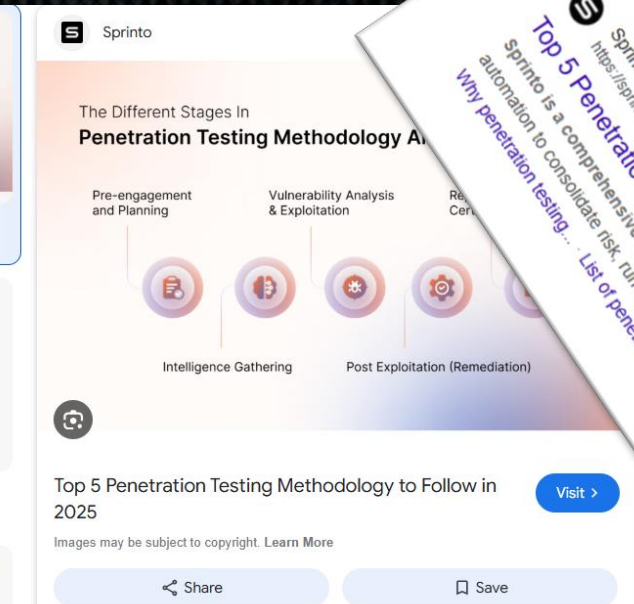
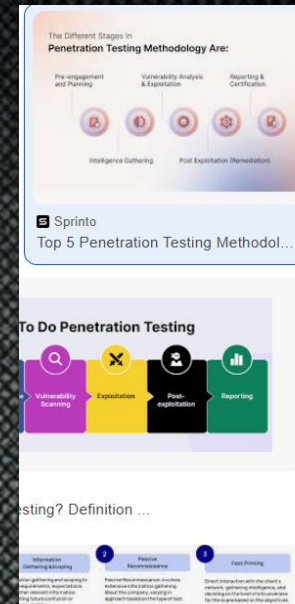
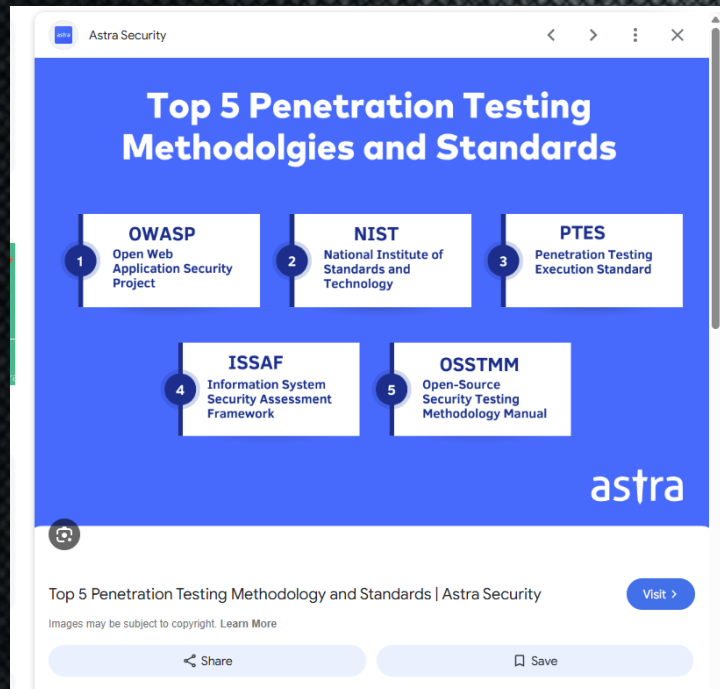
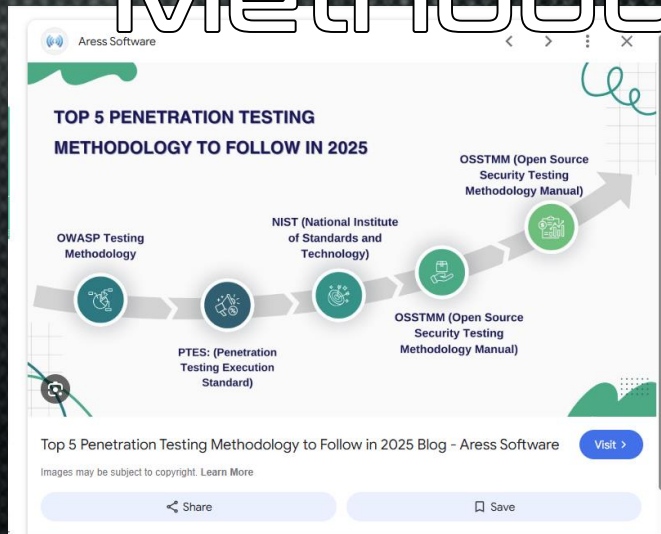
1. Reconnaissance (Passive & Active)

- **Passive Recon:** Harvesting intel without direct interaction (e.g., WHOIS, social media, DNS records, breach data).
- **Active Recon:** Touching the target (e.g., port scanning, banner grabbing, service enumeration).

Do we do reconnaissance in a modern pentesting?

Or do we just tell what ever the customer asks us to?

The Latest and Greatest Methodologies!



SPRINTO Frameworks Platform Resources Integrations Company

Penetration Testing Execution Standard (PTES)

Rate Share Subscribe

Penetration Testing Execution Standard (PTES)

PTES, or the Penetration Testing Execution Standard, is a penetration testing framework tailored to serve as a standard for conducting penetration testing. It was developed by a group of security experts to offer a consistent and repeatable methodology for testing.

Different penetration testing methods can be implemented for different types of

to secure system and network infrastructure. These include multiple o penetration testing, such as application penetration testing, web-based testing, static analysis, dynamic analysis, and social engineering tests, s to prevent hacker attacks and ensure infrastructure security.

ires of PTES include:

value your privacy

se cookies to enhance your browsing experience, s personalized ads or content, and analyze our traffic. icking "Accept All", you consent to our use of les.

Customise Accept All

Top 5 Penetration Testing Methodology to Follow in 2025

Sprinto is a comprehensive compliance and security solution that leverages the power of automation to consolidate risk, run fully automated checks, and map...

Why penetration testing... List of penetration testing...

Use Your Security Testing Provider

If you want to simulate a cyber attack and understand the consequences of vulnerabilities being exploited, you should perform a **penetration test**. Such a test will tell you whether it is possible to break into the company's network and achieve specific goals.

Make sure to check out the following list on how to choose the right provider and maximize project value:

What You Are Buying

provider, what you get when buying a penetration test can vary greatly. There is no unanimous standard for and how it is supposed to be conducted. It is therefore important for you to ask the provider about what standard they are following. If the answer is "my own", there is reason to worry.

to maximize value of the test, the provider should follow one of the international standards for pentesting, such as **Penetration Testing Execution Standard (PTES)** or **OWASP** for application testing. First of all this will ensure you a structured process. Second, you will have an idea of what you are buying.

What Button Test

The market is expanding, and so is the number of pentest providers. There are two types of providers, the ones that are serious about what they do, and the ones that are in it for the money.

Many providers probably offer you what is called "big fat button" test. This test consists of a security consultant directing an automated tool against a network or application, and thereafter letting the tool do all the work. Unfortunately a tool can only find obvious vulnerabilities. This approach will provide little value to organizations that have a mature security.

service

We set up a plan for continuous pentesting throughout the year together with you. You get:

- Thorough practical examination
- Tailored to the actual risks of your business and industry
- The penetration testing is using frameworks like the *Penetration Testing Execution Standard (PTES)*
- Full report including recommendations and solutions to improve your cyber security
- Work together to remedy the important findings and retest to verify them

Book a pentest meeting

All pentests in Norway performed by [redacted] follow the Penetration Testing Execution Standard (PTES) framework:



PTES, OSTMM, OWASP

Are you as confused as I am?



[Main page](#)
[PTES Technical Guideline](#)
[In the Media](#)
[FAQ](#)

[Tools](#)
[What links here](#)
[Related changes](#)
[Special pages](#)
[Printable version](#)
[Permanent link](#)
[Page information](#)

[Main page](#)

[Read](#)

[View source](#)

[View history](#)

[Q](#)

Main Page

High Level Organization of the Standard

The penetration testing execution standard consists of seven (7) main sections. These cover everything related to a penetration test - from the initial communication and reasoning behind a pentest, through the intelligence gathering and threat modeling phases where testers are working behind the scenes in order to get a better understanding of the tested organization, through vulnerability research, exploitation and post exploitation, where the technical security expertise of the testers come to play and combine with the business understanding of the engagement, and finally to the reporting, which captures the entire process, in a manner that makes sense to the customer and provides the most value to it.

This version can be considered a v1.0 as the core elements of the standard are solidified, and have been "road tested" for over a year through the industry. A v2.0 is in the works soon, and will provide more granular work in terms of "levels" - as in intensity levels at which each of the elements of a penetration test can be performed at. As no pentest is like another, and testing will range from the more mundane web application or network test, to a full-on red team engagement, said levels will enable an organization to define how much sophistication they expect their adversary to exhibit, and enable the tester to step up the intensity on those areas where the organization needs them the most. Some of the initial work on "levels" can be seen in the intelligence gathering section.

Following are the main sections defined by the standard as the basis for penetration testing execution:

- [Pre-engagement Interactions](#)
- [Intelligence Gathering](#)
- [Threat Modeling](#)
- [Vulnerability Analysis](#)
- [Exploitation](#)
- [Post Exploitation](#)
- [Reporting](#)

As the standard does not provide any technical guidelines as far as how to execute an actual pentest, we have also created a technical guide to accompany the standard itself. The technical guide can be reached via the link below:

- [Technical Guidelines](#)

For more information on what this standard is, please visit:

- [The Penetration Testing Execution Standard: FAQ](#)

This page was last edited on 16 August 2014, at 20:14.

Content is available under [GNU Free Documentation License 1.2](#) unless otherwise noted.

[Privacy policy](#) [About The Penetration Testing Execution Standard](#) [Disclaimers](#)

XSS

<Contribution Needed>

CSRF

<Contribution Needed>

Ad-Hoc Networks

<Contribution Needed>

- Information Leakage

Detection bypass

<Contribution Needed>

- FW/WAF/IDS/IPS Evasion
- Human Evasion
- DLP Evasion

Resistance of Controls to attacks

<Contribution Needed>

Type of Attack

<Contribution Needed>

- Client Side
- Phishing (w/pretext)
- Service Side
- Out of band
- Post-Exploitation
- Infrastructure analysis

The Social-Engineer Toolkit

The Social-Engineering Toolkit (SET) is a python-driven suite of custom tools which solely focuses on attacking the hu attacks and allow the tester to effectively test how a targeted attack may succeed. Currently SET has two main metho malicious website (which you can clone whatever one you want) that ultimately delivers your payload. The second me open-mail relay, a customized sendmail open-relay, or Gmail integration to deliver your payloads through e-mail. The (You can see detailed [tutorials here](#) or by downloading the [user manual here](#).

This page was last edited on 30 April 2012, at 06:04.

Content is available under [GNU Free Documentation License 1.2](#) unless otherwise noted.

[Privacy policy](#) [About The Penetration Testing Execution Standard](#) [Disclaimers](#)

OSSTMM 3 – The Open Source Security Testing Methodology Manual

Version Information

The current version of the Open Source Security Testing Methodology Manual (OSSTMM) is 3.02. This version of the OSSTMM ends the 2.x series. All OSSTMM versions prior to 3.0 including 3.0 release candidates (RC versions) are now obsolete.

The original version was published on Monday, December 18, 2000. This current version is published on Tuesday, December 14, 2010.

what is the latest version os OSSTMM



The latest version of the Open Source Security Testing Methodology Manual (OSSTMM) is version 3.02, published on December 14, 2010. ISECOM This version marked a significant evolution from the 2.x series, introducing a unified methodology applicable across all channels: Human, Physical,

what is OWASP WSTG?



OWASP WSTG (Web Security Testing Guide) is a comprehensive manual for testing the security of web applications. It's maintained by the [OWASP](#) Foundation and is widely used by penetration testers, developers, and security professionals.



[PROJECTS](#) [CHAPTERS](#) [EVENTS](#) [ABOUT](#) [Q](#)

[Store](#)

[Donate](#)

[Join](#)

[Watch](#) 354 [Star](#) 7,790

WSTG - v4.2

[Home](#) > [V42](#)

Table of Contents

0. Foreword by Eoin Keary

1. Frontispiece

2. Introduction

2.1 The OWASP Testing Project

2.2 Principles of Testing

2.3 Testing Techniques Explained

2.4 Manual Inspections and Reviews

2.5 Threat Modeling

2.6 Source Code Review

The **OWASP® Foundation** works to improve the security of software through its community-led open source software projects, hundreds of chapters worldwide, tens of thousands of members, and by hosting local and global conferences.

WSTG Contents (v4.2)

- [0. Foreword by Eoin Keary](#)
- [1. Frontispiece](#)
- [2. Introduction](#)
- [2.1 The OWASP Testing Project](#)
- [2.2 Principles of Testing](#)
- [2.3 Testing Techniques Explained](#)
- [2.4 Manual Inspections and Reviews](#)
- [2.5 Threat Modeling](#)
- [2.6 Source Code Review](#)
- [2.7 Penetration Testing](#)
- [2.8 The Need for a Balanced Approach](#)
- [2.9 Deriving Security Test Requirements](#)

WSTG Contents (v4.2)

- 0. Foreword by Eoin Keary
- 1. Frontispiece
- 2. Introduction
 - 2.1 The OWASP Testing Project
 - 2.2 Principles of Testing
 - 2.3 Testing Techniques Explained
 - 2.4 Manual Inspections and Reviews
 - 2.5 Threat Modeling
 - 2.6 Source Code Review
 - 2.7 Penetration Testing
 - 2.8 The Need for a Balanced Approach
 - 2.9 Deriving Security Test Requirements
 - 2.10 Security Tests Integrated in Development and Testing Workflows
 - 2.11 Security Test Data Analysis and Reporting
- 3. The OWASP Testing Framework
 - 3.1 The Web Security Testing Framework
 - 3.2 Phase 1 Before Development Begins
 - 3.3 Phase 2 During Definition and Design
 - 3.4 Phase 3 During Development
 - 3.5 Phase 4 During Deployment
 - 3.6 Phase 5 During Maintenance and Operations
 - 3.7 A Typical SDLC Testing Workflow
 - 3.8 Penetration Testing Methodologies
- 4. Web Application Security Testing
 - 4.0 Introduction and Objectives
 - 4.1 Information Gathering
 - 4.1.1 Conduct Search Engine Discovery Reconnaissance for Information Leakage
 - 4.1.2 Fingerprint Web Server
 - 4.1.3 Review Webserver Metafiles for Information Leakage
 - 4.1.4 Enumerate Applications on Webserver

- 4.1.4 Enumerate Applications on Webserver
 - 4.1.5 Review Webpage Content for Information Leakage
 - 4.1.6 Identify Application Entry Points
 - 4.1.7 Map Execution Paths Through Application
 - 4.1.8 Fingerprint Web Application Framework
 - 4.1.9 Fingerprint Web Application
 - 4.1.10 Map Application Architecture
 - 4.2 Configuration and Deployment Management Testing
 - 4.2.1 Test Network Infrastructure Configuration
 - 4.2.2 Test Application Platform Configuration
 - 4.2.3 Test File Extensions Handling for Sensitive Information
 - 4.2.4 Review Old Backup and Unreferenced Files for Sensitive Information
 - 4.2.5 Enumerate Infrastructure and Application Admin Interfaces
 - 4.2.6 Test HTTP Methods
 - 4.2.7 Test HTTP Strict Transport Security
 - 4.2.8 Test RIA Cross Domain Policy
 - 4.2.9 Test File Permission
 - 4.2.10 Test for Subdomain Takeover
 - 4.2.11 Test Cloud Storage
 - 4.3 Identity Management Testing
 - 4.3.1 Test Role Definitions
 - 4.3.2 Test User Registration Process
 - 4.3.3 Test Account Provisioning Process
 - 4.3.4 Testing for Account Enumeration and Guessable User Account
 - 4.3.5 Testing for Weak or Unenforced Username Policy
 - 4.4 Authentication Testing
 - 4.4.1 Testing for Credentials Transported over an Encrypted Channel
 - 4.4.2 Testing for Default Credentials
 - 4.4.3 Testing for Weak Lock Out Mechanism
 - 4.4.4 Testing for Bypassing Authentication Schema
 - 4.4.5 Testing for Vulnerable Remember Password

- 4.4.6 Testing for Browser Cache Weaknesses
 - 4.4.7 Testing for Weak Password Policy
 - 4.4.8 Testing for Weak Security Question Answer
 - 4.4.9 Testing for Weak Password Change or Reset Functionalities
 - 4.4.10 Testing for Weaker Authentication in Alternative Channel
 - 4.5 Authorization Testing
 - 4.5.1 Testing Directory Traversal File Include
 - 4.5.2 Testing for Bypassing Authorization Schema
 - 4.5.3 Testing for Privilege Escalation
 - 4.5.4 Testing for Insecure Direct Object References
 - 4.6 Session Management Testing
 - 4.6.1 Testing for Session Management Schema
 - 4.6.2 Testing for Cookies Attributes
 - 4.6.3 Testing for Session Fixation
 - 4.6.4 Testing for Exposed Session Variables
 - 4.6.5 Testing for Cross Site Request Forgery
 - 4.6.6 Testing for Logout Functionality
 - 4.6.7 Testing Session Timeout
 - 4.6.8 Testing for Session Puzzling
 - 4.6.9 Testing for Session Hijacking
 - 4.7 Input Validation Testing
 - 4.7.1 Testing for Reflected Cross Site Scripting
 - 4.7.2 Testing for Stored Cross Site Scripting
 - 4.7.3 Testing for HTTP Verb Tampering
 - 4.7.4 Testing for HTTP Parameter Pollution
 - 4.7.5 Testing for SQL Injection
 - 4.7.5.1 Testing for Oracle
 - 4.7.5.2 Testing for MySQL
 - 4.7.5.3 Testing for SQL Server
 - 4.7.5.4 Testing PostgreSQL
 - 4.7.5.5 Testing for MS Access
 - 4.7.5.6 Testing for NoSQL Injection
 - 4.7.5.7 Testing for ORM Injection
 - 4.7.5.8 Testing for Client-side
 - 4.7.6 Testing for LDAP Injection



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>

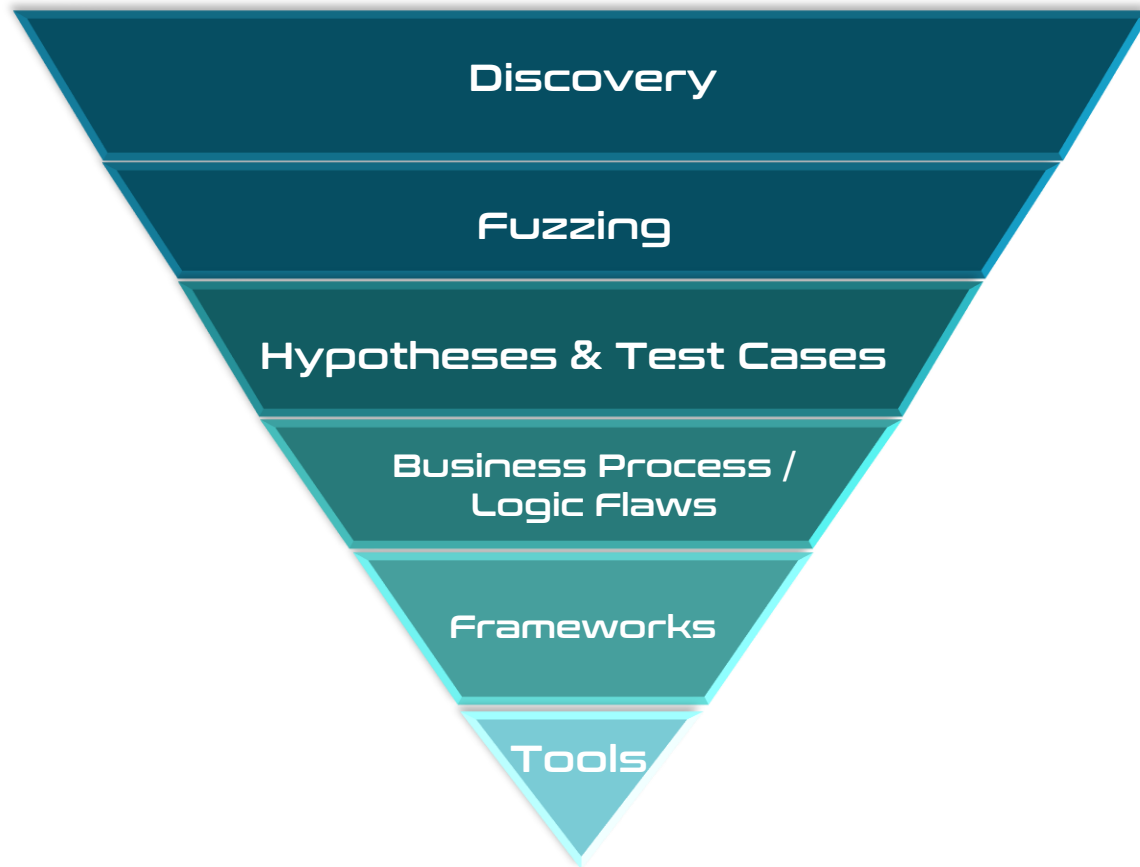
renewed 2023, 2024

Most Valuable Pentesting

TIMELESS METHODOLOGY FOR PENTESTING
TRADECRAFT

Most Valuable Pentesting (MVP) Methodology

Be Honest, Find Vulnerabilities, Improve Gradually



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 unturned. Many vulnerabilities are found in the "paths least travelled". Fully explore!



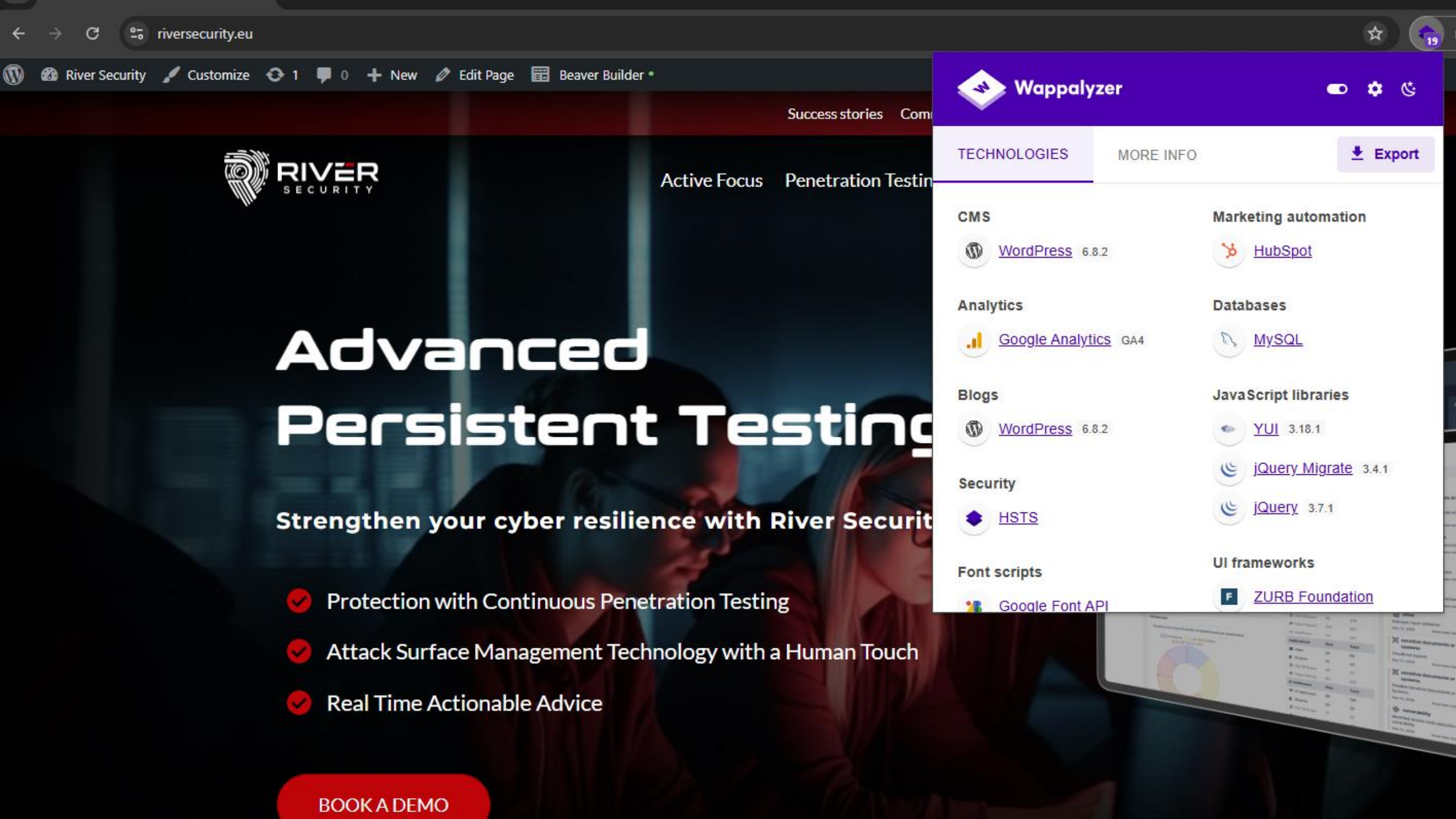
Reporting

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.



Advanced Persistent Testing

Strengthen your cyber resilience with River Security

- ✓ Protection with Continuous Penetration Testing
- ✓ Attack Surface Management Technology with a Human Touch
- ✓ Real Time Actionable Advice

BOOK A DEMO



Wappalyzer



TECHNOLOGIES

MORE INFO

Export

CMS

[WordPress](#) 6.8.2

Analytics

[Google Analytics](#) GA4

Blogs

[WordPress](#) 6.8.2

Security

[HSTS](#)

Font scripts

[Google Font API](#)

Marketing automation

[HubSpot](#)

Databases

[MySQL](#)

JavaScript libraries

[YUI](#) 3.18.1

[jQuery Migrate](#) 3.4.1

[jQuery](#) 3.7.1

UI frameworks

[ZURB Foundation](#)

WordPress Enumeration

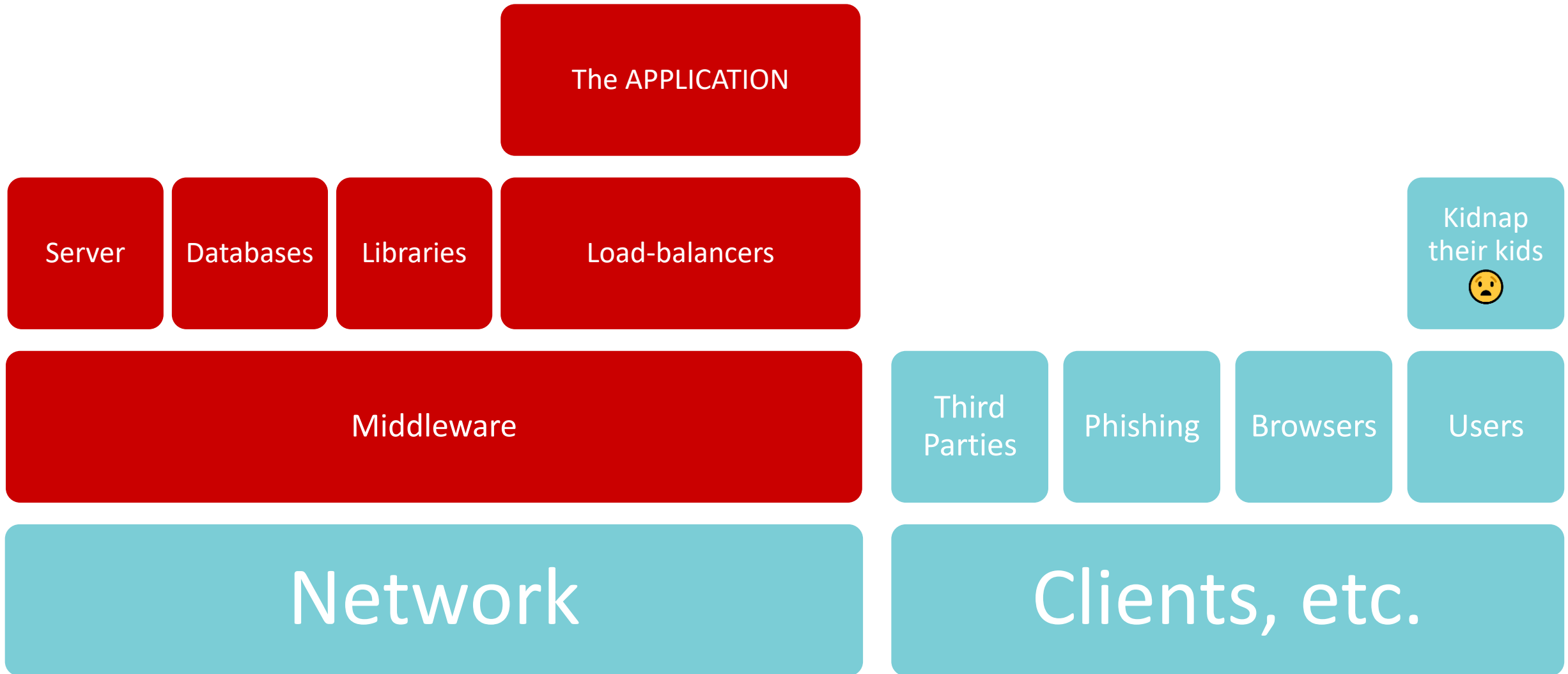
#USERS

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-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>

What to attack in the pentest?





WordPress (MVP)



Page



Discussion



Contents

[hide](#)

- 1 [Discovery](#)
- 2 [Fuzzing](#)
- 3 [Test Cases & Hypotheses](#)
 - 3.1 [Wordpress Debug.log](#)
- 4 [Business Process and Logic Flaws](#)
- 5 [Framework](#)
- 6 [Tools](#)

Discovery



[edit](#) [edit source](#)

- Run [wordpress-rest-enum](#) to look for **inappropriate content**, **sensitive file** and other interesting the API might expose
- Use [wpscan](#) to help identify themes and plugins, this is where you might strike gold.
 - Themes and plugins can be vulnerable, should be worked on individually. At the very list, do your best to identify all themes, plugins, customization and other entypoints.
- Unless looking for a zero-day, the core engine of wordpress is likely to be quite hardened on its own. Look for known CVE's for the engine.
- Use Wordpress dictionary files and check all files and folders for access control issues

Fuzzing



[edit](#) [edit source](#)

- The plugins might be open-source. Check them out, find all entypoints/endpoints and look for vulnerabilities



Category:MVP Type

[? Help](#)

[Category](#) [Discussion](#) [★](#)

[Edit](#) [Edit source](#) [History](#)

A MVP can be for different things, for example:

- A [framework](#), e.g. [Django \(MVP\)](#), [PHP \(MVP\)](#), [Angular \(MVP\)](#), [React \(MVP\)](#), etc.
- A [middleware](#), e.g. [IIS \(MVP\)](#), [Apache \(MVP\)](#), [Nginx \(MVP\)](#), [CosmosDB \(MVP\)](#)
- A specific type of [application](#), e.g. [WordPress \(MVP\)](#), [Salesforce \(MVP\)](#)
- A [feature](#) or functionality, for example [File Upload \(MVP\)](#), [Login \(MVP\)](#), [MFA \(MVP\)](#), [Payment \(MVP\)](#)
- A [general](#) type of testing type, e.g. [Commercial of the Shelf \(MVP\)](#), [Mobile Testing \(MVP\)](#), [Hardware Testing \(MVP\)](#)

Subcategories

This category has the following 9 subcategories, out of 9 total.

A

- ▶ [Application-MVP \(45 P\)](#)

F

- ▶ [Feature-MVP \(12 P\)](#)
- ▶ [Framework-MVP \(16 P\)](#)

G

- ▶ [General-MVP \(17 P\)](#)

M

- ▶ [Middleware-MVP \(9 P\)](#)
- ▶ [MVP Type \(9 C\)](#)

N

- ▶ [Notype-MVP \(1 P\)](#)

P

- ▶ [Protocol-MVP \(5 P\)](#)
- ▶ [Provider-MVP \(10 P\)](#)



IIS Short Name Scanning

```
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
```

- Angular (MVP)
- AngularJS (MVP)
- Apache (MVP)
- API (MVP)
- ArcGis (MVP)
- Argo CD (MVP)
- Auth0 (MVP)
- Authentication (MVP)
- Azure API Management (MVP)
- Azure Function (MVP)

B

- Backstage (MVP)
- Bankid (MVP)
- Blazor (MVP)
- Bookstack (MVP)
- Bynder (MVP)

C

- Citrix Gateway (MVP)
- Cloudflare (MVP)
- Cognito (MVP)
- Commercial of the Shelf (MVP)
- Content Management System (MVP)
- CorePublish (MVP)
- CosmosDB (MVP)
- CPANEL (MVP)

- Hybridauth (MVP)

I

- IIS (MVP)
- Ivanti (MVP)

J

- Javascript frameworks (MVP)
- Jira (MVP)
- JSON-RPC (MVP)

L

- Lambda (MVP)
- Local Privilege Escalation (Windows) (MVP)
- Login (MVP)

M

- Machform (MVP)
- Magento (MVP)
- Mediawiki (MVP)
- Mega Upload (MVP)
- Methodology (Hardware)
- Methodology (WIFI)
- MFA (MVP)

S

- S3.amazonaws.com (MVP)
- Salesforce (MVP)
- Salesforce Marketing Cloud (MVP)
- Salto (MVP)
- Sanity (MVP)
- SAP CX Backoffice (MVP)
- SAP NetWeaver (MVP)
- Search (MVP)
- ServiceStack (MVP)
- Sharepoint (Authenticated) (MVP)
- Sharepoint (MVP)
- Sharepoint Foundation (MVP)
- Signicat (MVP)
- Sinatra (MVP)
- SIP (MVP)
- SparQL (MVP)
- SQL Injection (MVP)
- SSH (External) (MVP)
- SSRF (MVP)
- Subdomain Takeover (MVP)
- Sysero (MVP)

T

- Third Party Application (MVP)

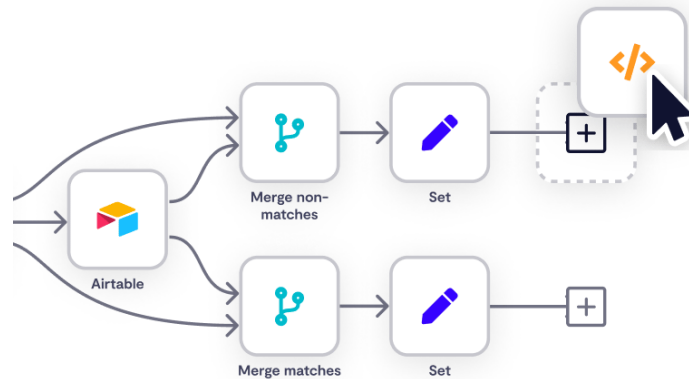
- Test for Reflected Cross Site Scripting
- Test for Stored Cross Site Scripting
- Test for DOM based Cross Site Scripting
- Test for Cross Site Flashing
- Test for HTML Injection
- Test for SQL Injection
- Test for SOQL Injection
- Test for LDAP Injection
- Test for ORM Injection
- Test for XML Injection
- 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

WEB APPLICATION PENETRATION TESTING

IT'S NOT JUST CHECKLISTS



AI as a Pentester Companion



Caption: Illustrative image of agentic workflows from n8n.io

- AI has knowledge of the entity you are testing
 - What features
 - What technology (tech stack, libraries, plugins, you name it)
 - This is based on automated collected results
 - Think scanners, collectors / sensors, ASM data
- When pentester “arrives” to the asset, they can be presented with DYNAMIC checklists
- AI companion can also make sure documentation of work is done according to attack surface



When You Don't Have MVP

- Create one
 - Set up the **minimum** of what you know
 - A starting point is better than nothing
 - Get the ball rolling with your team
- Dedicate days before the engagement to:
 - Build
 - Set-up
 - Configure
 - Break & Hack
 - Create CTF challenges ;)
- Create foundations for future hypothesis





Incomplete MVP's

DocuWiki (MVP)

[Page](#) [Discussion](#) [★](#)

- Relevant: [Mediawiki \(MVP\)](#)

Contents [hide](#)

- 1 [Discovery](#)
- 2 [Fuzzing](#)
- 3 [Test Cases & Hypotheses](#)
- 4 [Business Process and Logic Flaws](#)
- 5 [Framework](#)
- 6 [Tools](#)

```
docker run -d --name dokuwiki \  
-p 8080:8080 -p 8443:8443 \  
-e DOKUWIKI_USERNAME=admin \  
-e DOKUWIKI_PASSWORD=supersecurepassword \  
-e DOKUWIKI_FULL_NAME="Admin User" \  
-e DOKUWIKI_EMAIL=admin@example.com \  
-v dokuwiki_data:/bitnami/dokuwiki \  
bitnami/dokuwiki:latest
```

Discovery [edit](#) [edit source](#)


Get Version: Version `http://<domain>/VERSION`

- Interesting urls:
 - Login: `doku.php?do=login`
 - Register: `doku.php?do=register`
 - Recent changes: `doku.php?do=recent&show_changes=pages`
 - Media Files `doku.php?id=mfiles:mfiles&do=media&ns=mfiles`
- TODO: Generate Plugin fuzzing list, multiple plugins are bundled.
 - Located `http://<domain>/lib/plugins/`

Fuzzing [edit](#) [edit source](#)

- Check read permission on `/conf/` directory, fuzz for known backup extension for. Use a good wordlist for backup extensions
 - `/conf/local.php`
 - `/conf/users.auth.php`
 - `/conf/acl.auth.php`


Test Cases & Hypotheses [edit](#) [edit source](#)

 Can this MVP section be expanded? The reason given is: *What are good test cases to aim for? Any hypothesis to test?*

Business Process and Logic Flaws [edit](#) [edit source](#)

 Can this MVP section be expanded? The reason given is: *Any business process or logic flaws to consider?*

Framework [edit](#) [edit source](#)

 Can this MVP section be expanded? The reason given is: *Any good frameworks or external resources to help test for this?*

Tools [edit](#) [edit source](#)

 Can this MVP section be expanded? The reason given is: *What tools exist to aid testing? Any wordlists?*

[PayloadAllTheThings](#)
[Hacktricks](#)
[Exploit-DB](#)

Notice: Feel free to add important steps to complete during testing.

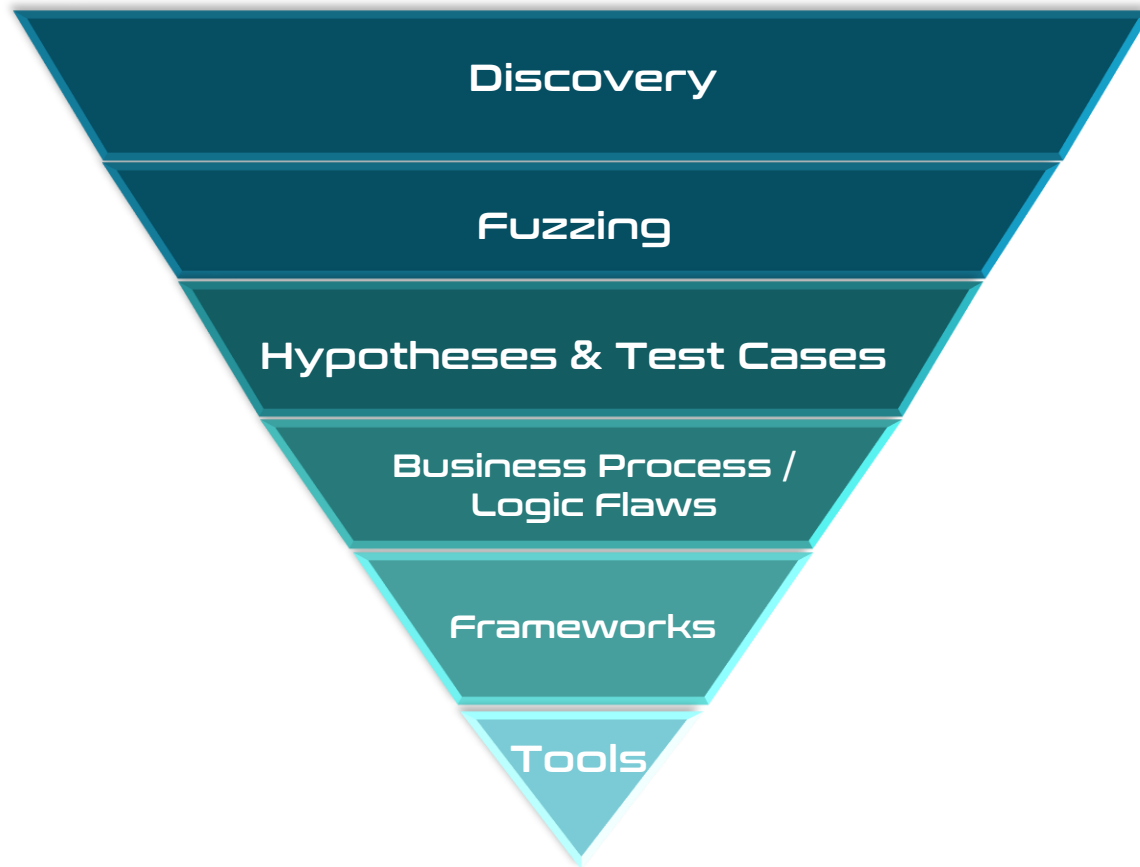
Keep in mind, this is not a page to learn about this technology, only steps to hack it. It should not contain how-to guides, but instead precise bullet points about what to achieve. See [How to write MVP](#). Use/Create [DocuWiki](#) for other documentation, e.g. tutorials, best practices, and other valuable information pertaining this [Technology](#).

Have you experience testing on this MVP? Add yourself to the list of users, so other pentesters can reach out to you for collaboration 🙌.

MVP
Application
Pages to be

Most Valuable Pentesting (MVP) Methodology

Be Honest, Find Vulnerabilities, Improve Gradually



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 unturned. 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.



Discovery

- What are you going to attack?
- What are you NOT going to attack?
 - Keep documenting what you are NOT attacking
- What can you source to your co-workers?
 - What skills do they have, what are they good at?
- Can you find every script, every piece of content and function?
 - Content enumeration is key. What if there is a gaping vulnerability in a script you didn't find?



Goal: Find Everything

Content Discovery

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

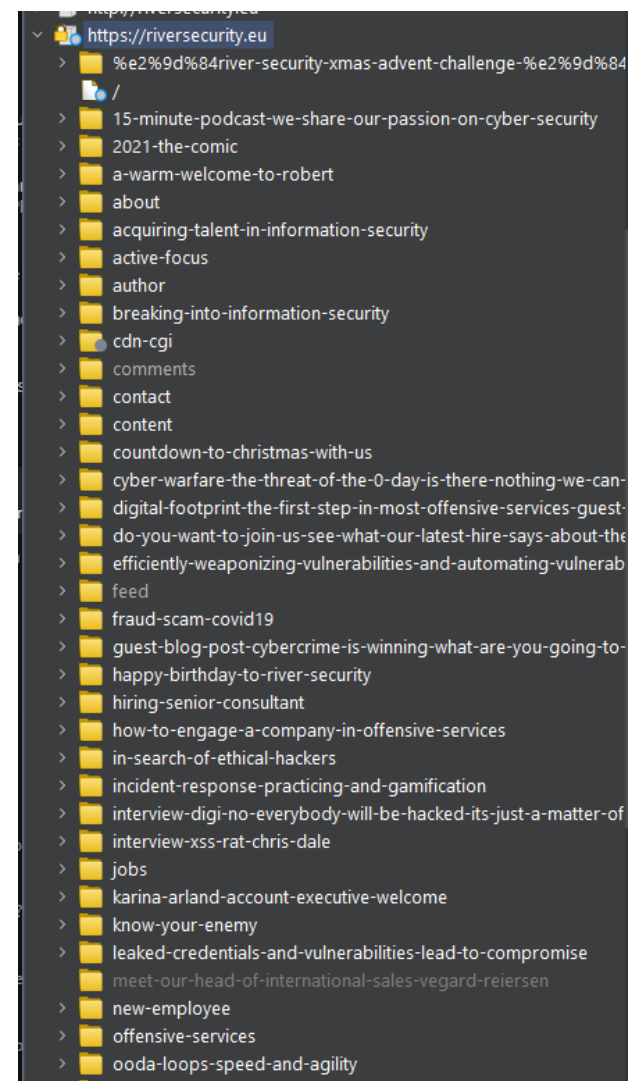
Leave no stone unturned. Many vulnerabilities are found in the "paths least travelled". Fully explore!



Map Browsable Attack Surface

Content Discovery

- 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





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:

Severity:

Confidence:

Host:

Path:

Secret input: header

Medium

Firm

https://riversecurity.eu

/

Note:

This issue was generated by a Burp extension.

Issue detail

Unlinked parameter identified.

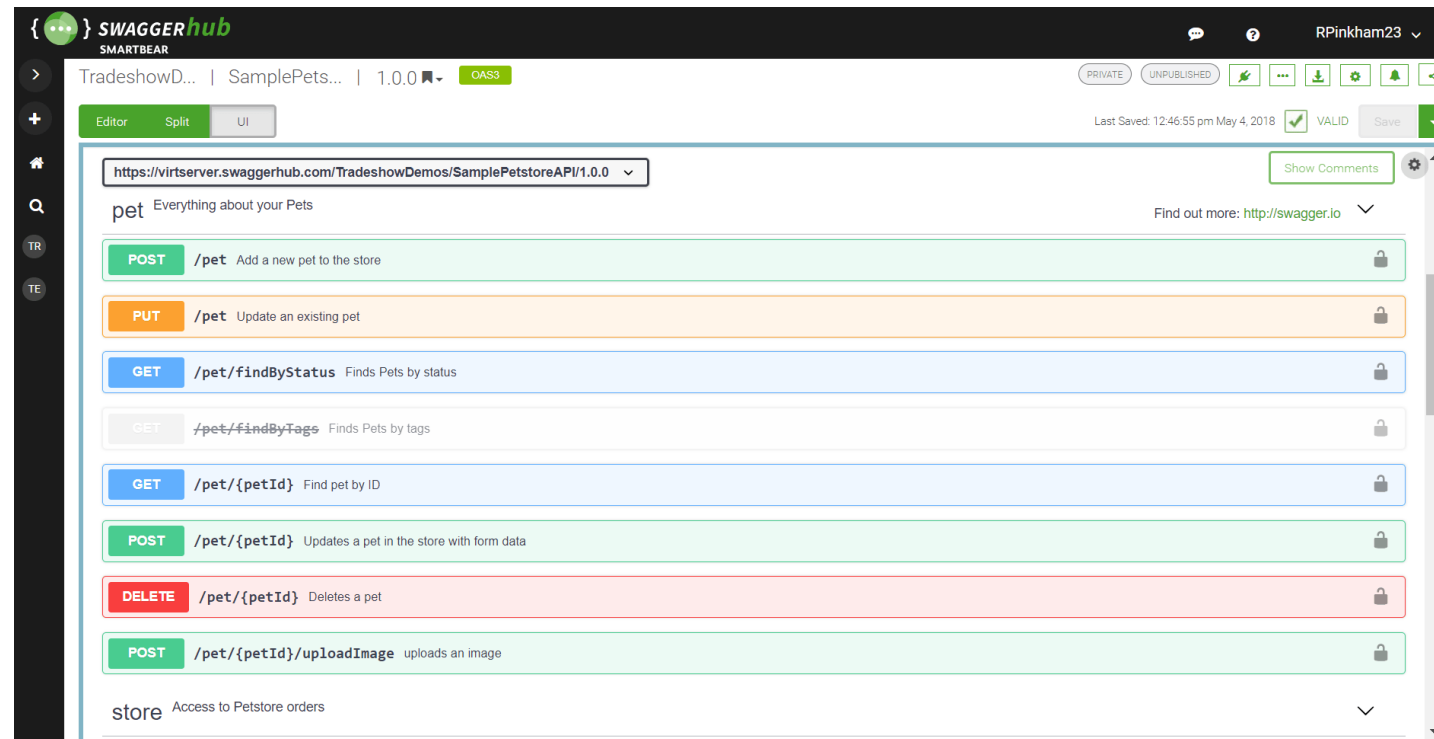
Successful probes

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*



OpenAPI / Swagger Specs

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





Archive.org

Content Discovery

- WaybackRobots.py

- WaybackURLs.py

SANS Institute
o n l i n e
A Cooperative Education & Research Organization

- ▶ [About the SANS Institute](#)
- ▶ [Contact SANS](#)
- ▶ [Search SANS](#)
- ▶ [SANS Key : Local copy](#)
- ▶ [Security News of the Week](#)
- ▶ [Northcutt Interview](#)
- ▶ [Whether Certification Matters](#)
- ▶ [SANS Forum](#)
- ▶ **ALERT:** [BIND DNS Buffer Overflow \(1/29/01\)](#)

Events

Major Conferences
[SANS 2001](#) Baltimore, MD
May 13 - 20, 2001
[SANS Parliament Sq.](#) London, England
June 20 - 23, 2001
[SANSFIRE](#) Washington DC
July 30 - August 3, 2001
[Call for Papers](#)
[SANS Parliament Hill](#) Ottawa, Canada
August 8 - 17, 2001
Info. available soon
[SANS Network Security 2001](#)
San Diego, CA October, 2001
Brochure available online July 2001

Regional Conferences
[SANS Darling Harbour](#) Sydney, Australia
February 12 - 15, 2001
[SANS Aloha II](#) Honolulu, Hawaii
February 27 - March 2, 2001

[Security KickStart](#)
[SANS Security Essentials](#)
[Windows NT 4.0 Security Step-by-Step](#)
NEW! [Advanced Incident Handling & Hacker Exploits](#)

[Today's GIAC Detects](#)
[Global Incident Analysis Center](#)
[SANS GIAC Security Skills Certification Program](#)
[Information Security Reading Room](#)
[Intrusion Detection FAQ](#)

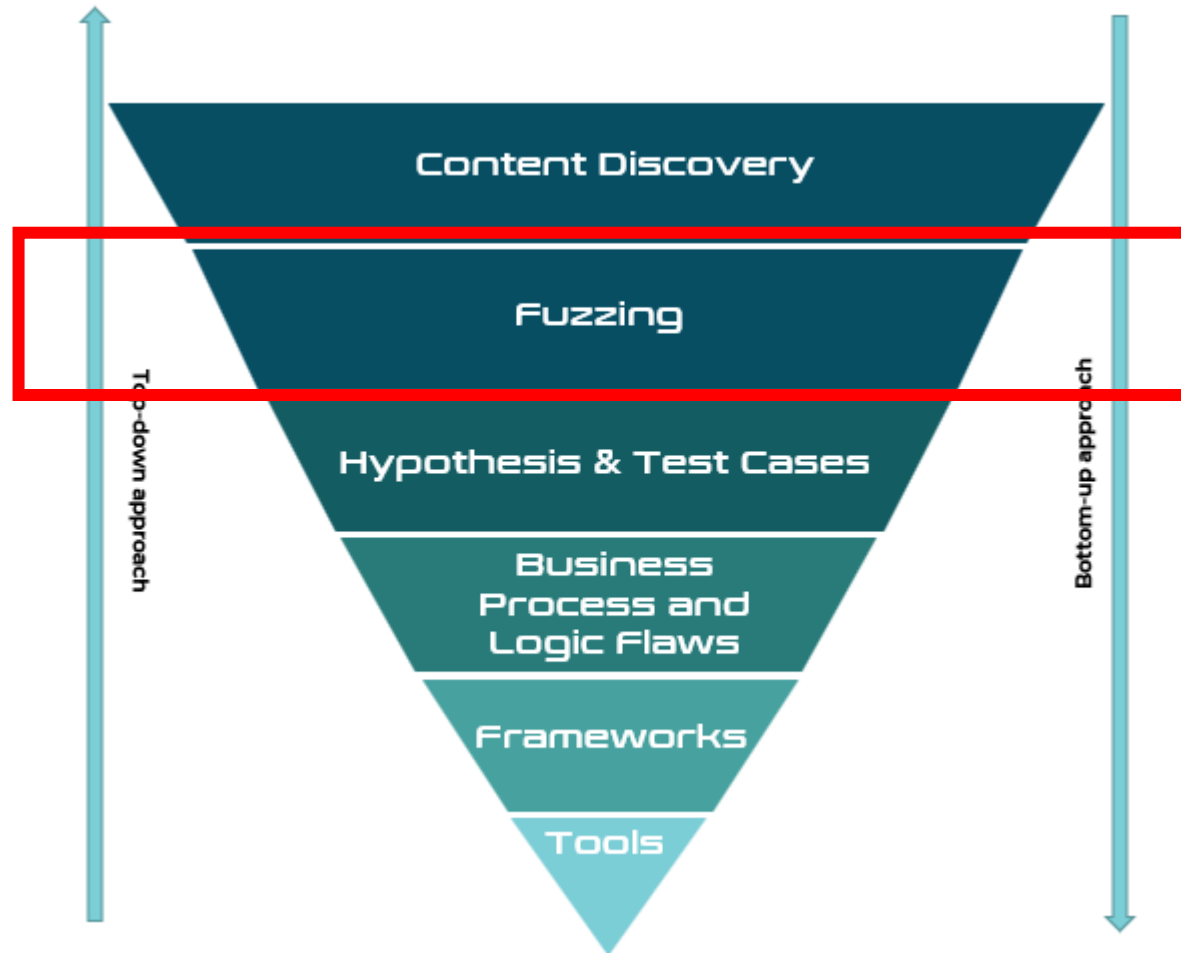
Resources

NEW! [Late Arriving SANS @ Night Presentation from SANS Security 2001 \(Programming Perl on NT by Harlan Carvey\)](#)



Fuzzing

Find bytes and input producing different/unexpected results

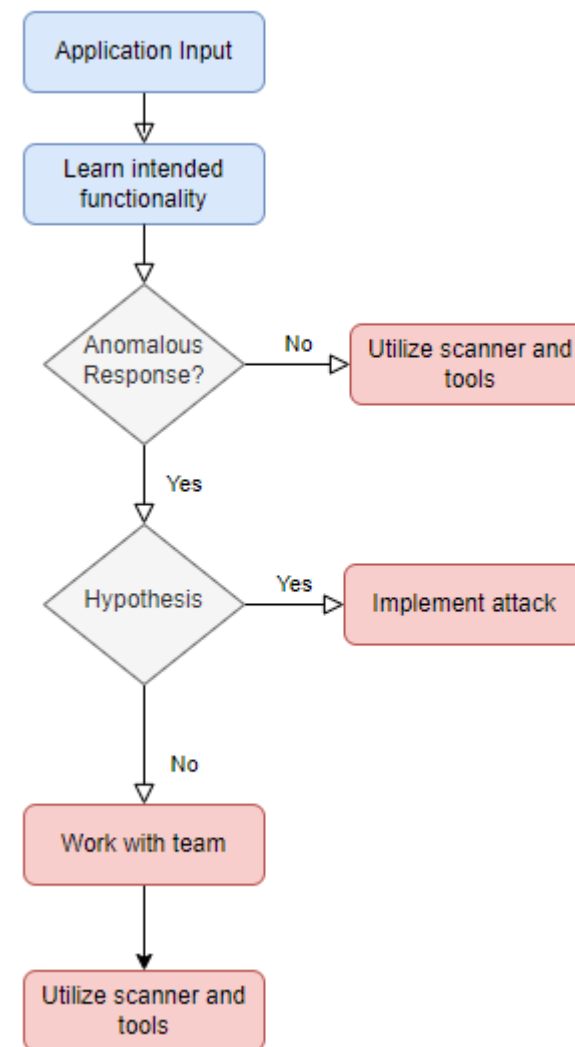




Fuzzing Bytes 101

Fuzzing

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





Second example: A Single Character

Fuzzing

DashboardTargetProxyIntruderRepeaterCollaboratorSequencerDecoder

1 x2 x3 x4 x5 x+

PositionsPayloadsResource PoolOptions

?

Payload Sets

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

Payload set: 1Payload count: 256

Payload type: Simple listRequest count: 256

?

Payload Options [Simple list]

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

PasteLoad ...RemoveClearDeduplicateAddAdd from list ...

%00%01%02%03%04%05%06

?

Payload Processing

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

AddEditRemoveUpDown

Enabled

Rule

☒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: .\^=<>?+&";'[]|'`

Filter: Showing all items

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	

RequestResponse

PrettyRawHexRender

```
18<script>
19  var wechallinfo = {
20    "level": "natas14", "pass": "qPazSJBmrmU7UQJv17MHk1PGC4Dx2MEP"
21  };
22</script>
23</head>
24<body>
25  <h1>
26    natas14
27  </h1>
28  <div id="content">
29    <br />
30    <b>
31      Warning
32    </b>
33    : mysqli_num_rows() expects parameter 1 to be mysqli_result, bool given in <b>
34      /var/www/natas/natas14/index.php
35    </b>
36    on line <b>
37      24
38    </b>
39    <br />
40    Access denied!<br>
41    <div id="viewsource">
42      <a href="index-source.html">
43        View sourcecode
44      </a>
45    </div>
46  </div>
47</body>
48</html>
```

0 matches

Finished



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>
20 <link href="css/default.css" rel="stylesheet" type="text/css" />
21 </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	Wildcard
5	500	%3A	:	FILE	ADS
6	500	%3E	>	FILE	Redirect
7	500	%3F	?	URL	
8	500	%3C	<	FILE	Redirect
9	404	%2B	+	URL	



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

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


Fuzzing

Name	Date modified	Type	Size
1. compilations	25/06/2023 11:53 pm	File folder	
command-injection	25/06/2023 11:53 pm	File folder	
control-characters	25/06/2023 11:53 pm	File folder	
elasticSearch	25/06/2023 11:53 pm	File folder	
file-upload	25/06/2023 11:53 pm	File folder	
format-strings	25/06/2023 11:53 pm	File folder	
html-javascript	25/06/2023 11:53 pm	File folder	
http	25/06/2023 11:53 pm	File folder	
integer-overflow	25/06/2023 11:53 pm	File folder	
ldap	25/06/2023 11:53 pm	File folder	
lfi	25/06/2023 11:53 pm	File folder	
no-sql	25/06/2023 11:53 pm	File folder	
path-traversal	25/06/2023 11:53 pm	File folder	
polyglot	25/06/2023 11:53 pm	File folder	
programming	25/06/2023 11:53 pm	File folder	
redirects	25/06/2023 11:53 pm	File folder	
server-side-include	25/06/2023 11:53 pm	File folder	
sql-injection	25/06/2023 11:53 pm	File folder	
xml	08/10/2023 12:59 pm	File folder	
xpath	25/06/2023 11:53 pm	File folder	
xss	02/10/2023 6:03 pm	File folder	
README.md	25/06/2023 11:53 pm	Markdown Source ...	1 KB

 file-ul-filter-bypass-microsoft-asp-PH-UE.txt

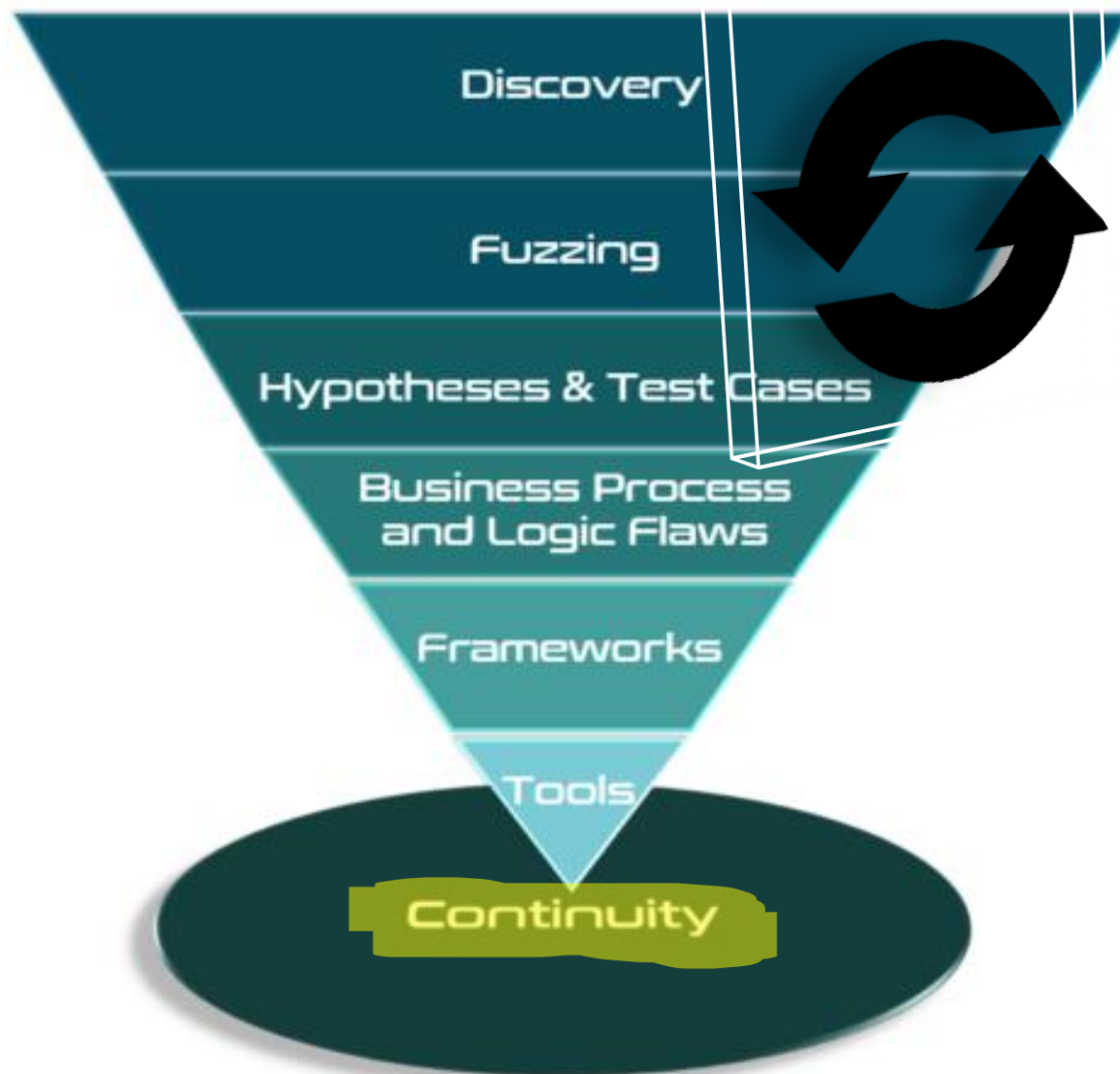
 file-ul-filter-bypass-ms-php.txt

 file-ul-filter-bypass-x-platform-generic-UE.txt

 file-ul-filter-bypass-x-platform-php-PH.txt

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 unturned. 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.

Here Be Dragons





RED VERSUS BLUE

IT IS TIME WE "STOP FIGHTING" AND FORM PURPLE TEAM





Connect with me – <https://into.bio/chrisdale>



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



Fighting Cyber Crime – <https://riversecurity.eu>