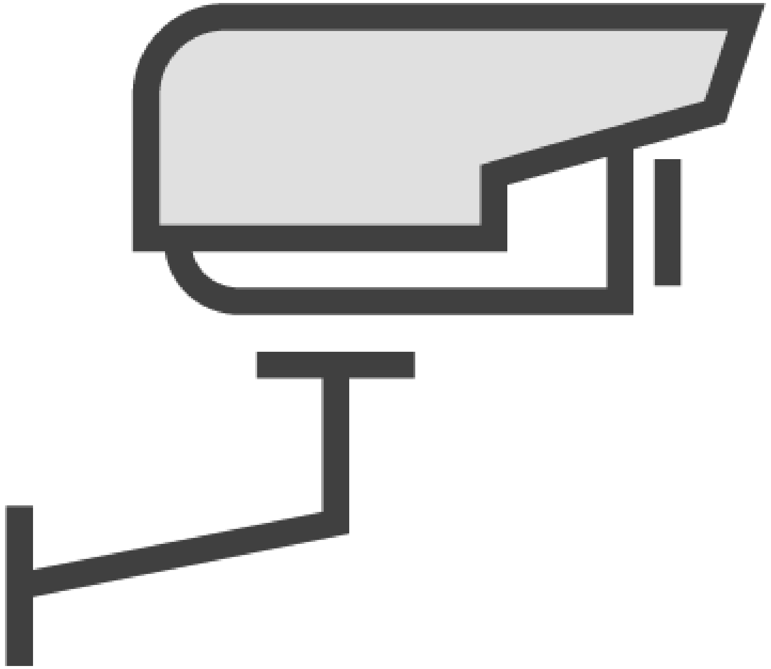
Scanning Your Web Application Functions

Passive vs. Active Scanning

Important Note

By law, you are only permitted to pentest, hack, and assess web applications you own or have permission to test

Passive Scanning



Automatically runs in a background thread against all responses from application

No change to response so safe to run without fear of repercussions

Alerts of known and common vulnerabilities

No logical scan capabilities

Active Scanning

An automated scan that is manually run.

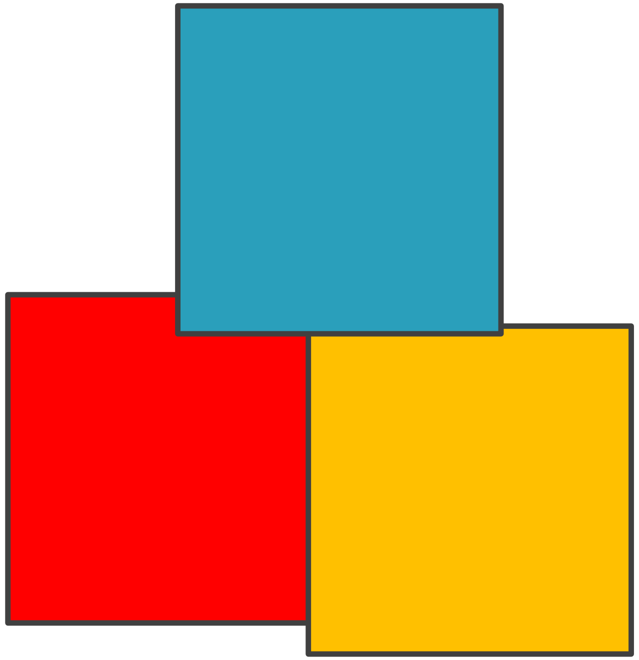
Except when in ATTACK mode

An attack on the application using known techniques to find common vulnerabilities.

No logical scan capabilities.

You should NOT use on an app you do not own or have permission to test .

Marketplace Addons



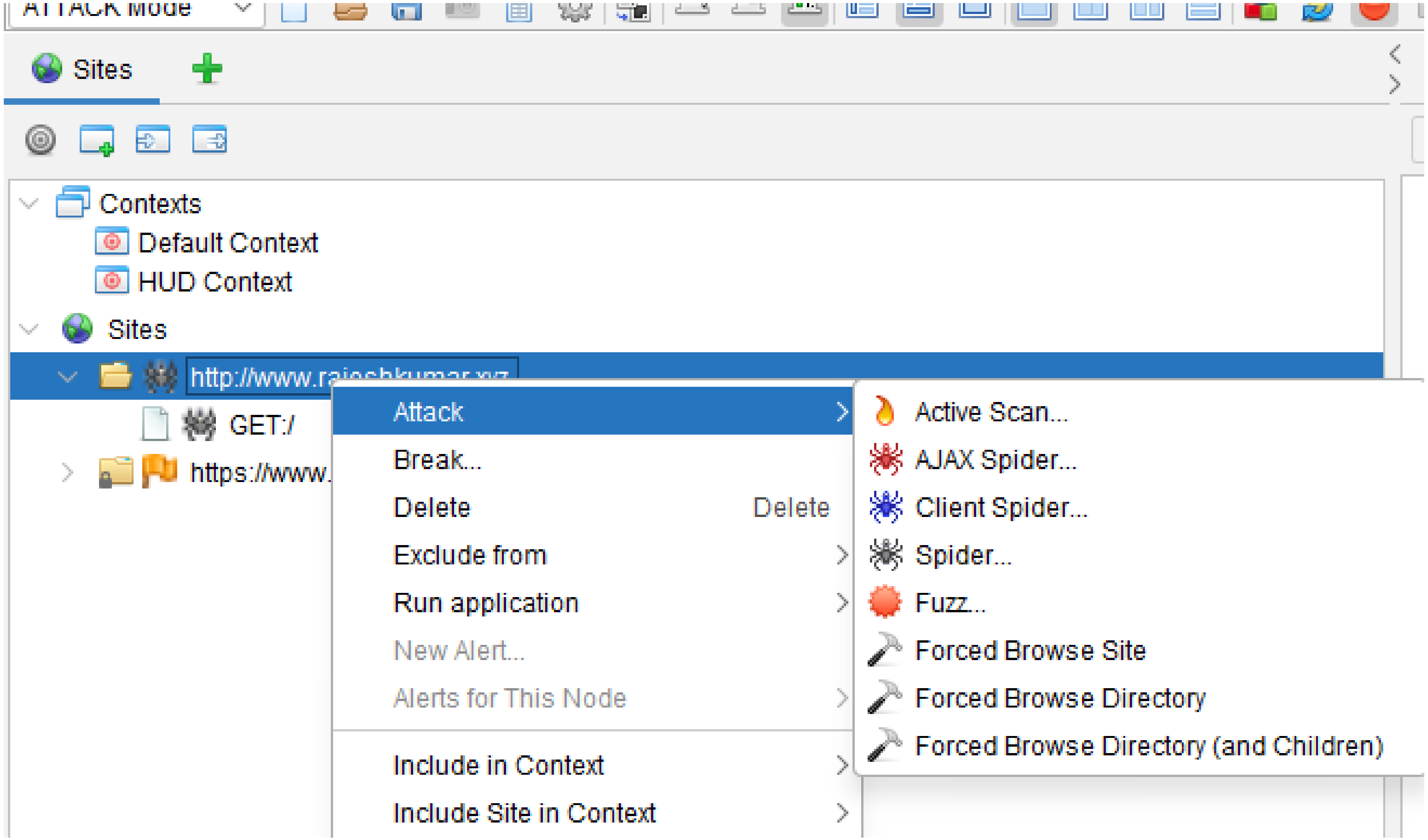
Active scanner rules (alpha)

Active scanner rules (beta)

Passive scanner rules (alpha)

Passive scanner rules (beta)

Mode	Active Scan	Scope Restriction	Risk Level	Typical Use Case
Safe	❌ No	🔒 N/A	🟢 Minimal	Passive monitoring of live systems
Protected	✅ Yes (In Scope)	✅ Enforced	🟡 Controlled	Staging or authorized targets
Standard	✅ Yes	❌ None	🟠 Moderate	Normal development testing
Attack	🔥 Aggressive	⚠️ In Scope (Recommended)	🔴 High	Full pentesting or lab attacks


ATTACK Mode
Sites
Contexts
Default Context
HUD Context
Sites
http://www.rajeshkumar.xyz
GET:/
https://www.
Attack
Break...
Delete
Delete
Exclude from
Run application
New Alert...
Alerts for This Node
Include in Context
Include Site in Context
Active Scan...
AJAX Spider...
Client Spider...
Spider...
Fuzz...
Forced Browse Site
Forced Browse Directory
Forced Browse Directory (and Children)

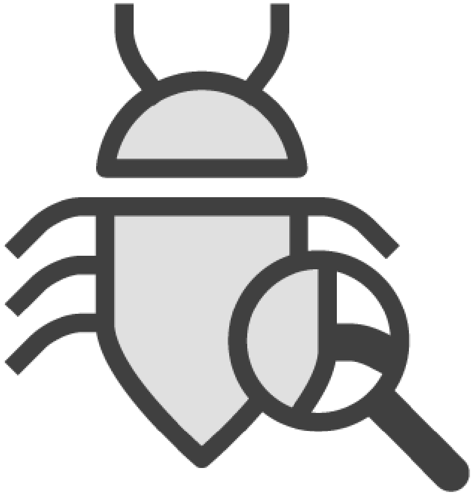
Menu Option	Purpose / Description	When to Use It
🔥 **Active Scan…**	Performs **automated vulnerability scanning** by sending crafted payloads and attacks (SQLi, XSS, path traversal, etc.) to test the target for known weaknesses.	Use after mapping the site (Spider done) to discover actual exploitable issues in a **test or staging** environment. Avoid on production.
🕷️ **AJAX Spider…**	Uses a **headless browser (e.g., Selenium)** to dynamically crawl JavaScript-heavy or SPA (Single Page Application) sites, executing JS to find hidden pages/endpoints.	Use when your app relies heavily on **AJAX / dynamic JS navigation** (React, Angular, Vue).
🕸️ **Client Spider…**	A newer **client-side spider** that runs inside the browser launched by ZAP. It observes what URLs and API calls the app itself triggers while you browse.	Use to capture **client-side navigation and API requests** that normal crawling might miss.
🕷️ **Spider…**	The **traditional crawler** that follows links and forms from a seed URL to map out all reachable pages. Works best on static or server-rendered content.	Run this **before Active Scan** to build the site tree and gather all links for coverage.
💥 **Fuzz…**	Sends **customized or random input values** into parameters, headers, or request bodies to discover unexpected behaviors (e.g., buffer overflow, input validation flaws).	Use for **fine-grained or manual testing** of parameters when you want to craft custom payloads.
🔨 **Forced Browse Site**	Attempts to **discover hidden files or directories** site-wide by brute-forcing common paths.	Use for **content discovery** on smaller or less restricted sites.
🔨 **Forced Browse Directory**	Same as above but limited to a **specific directory** you select.	Use when targeting one **folder path** to look for admin panels, backups, etc.
🔨 **Forced Browse Directory (and Children)**	Brute-forces the chosen directory **and all its sub-directories** recursively.	Use when you want a **deep recursive search** for hidden resources across a full directory tree.

Demo

Scan Policies

Quick Start

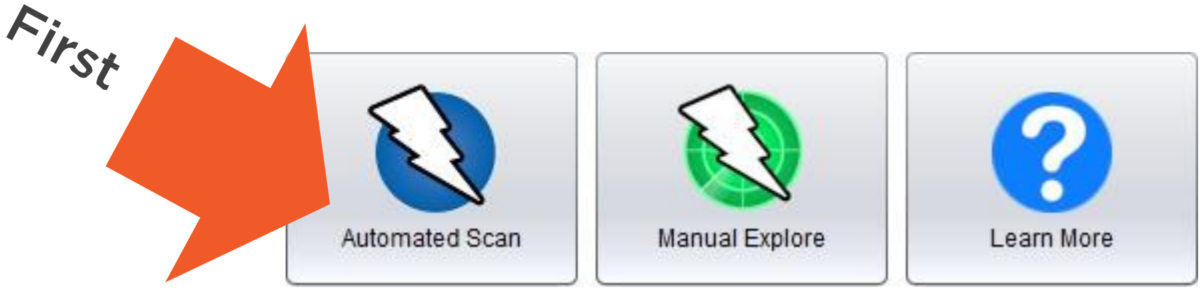
Quick Start

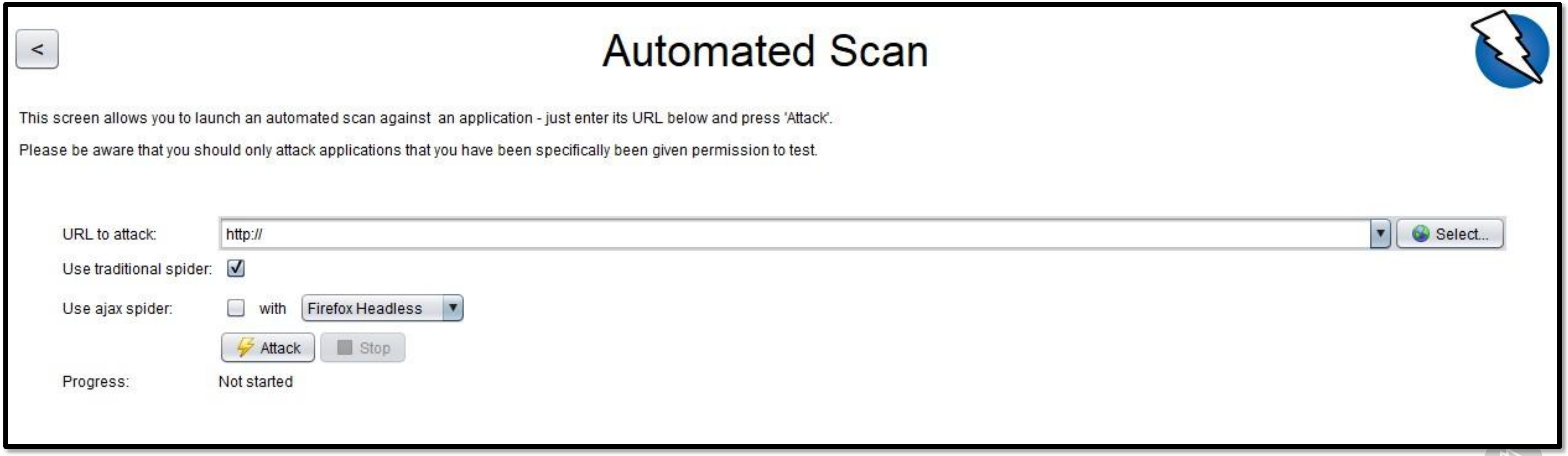


Spider

Active Scan

Quick Start





Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://

Use traditional spider: ☑

Use ajax spider: □ with Firefox Headless

Progress: Not started

Demo

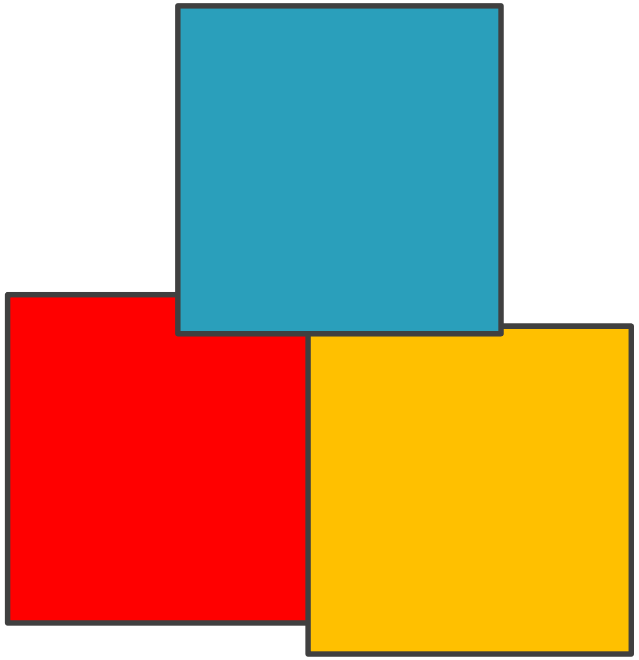
Quick Start

Fuzzer

> “QA Engineer walks into a bar. Orders a beer. Orders 0 beers. Orders 999999999 beers. Orders a lizard. Orders -1 beers. Orders a sfdeljknesv.”

Bill Sempf (@sempf)

Marketplace Addons

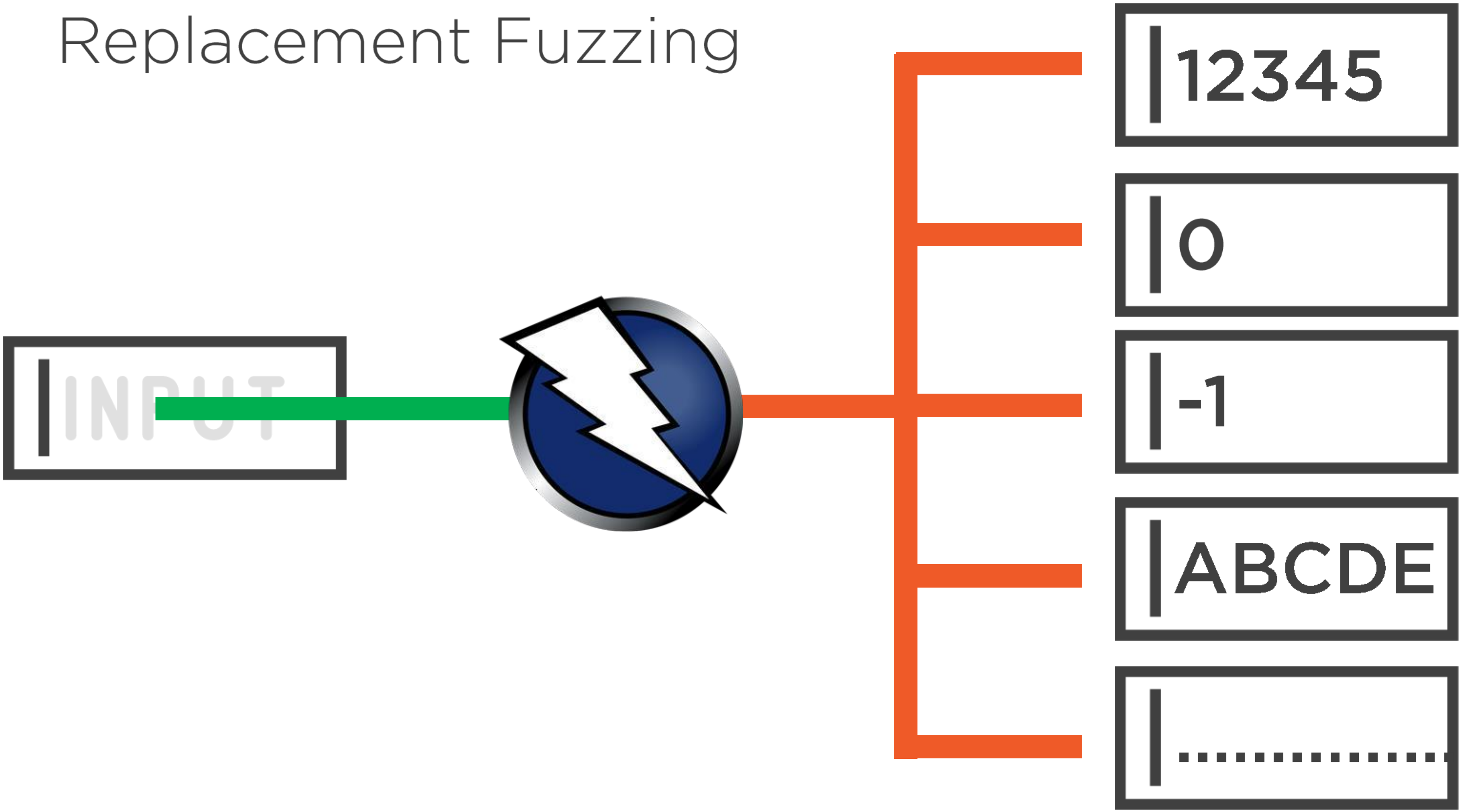


FuzzDB Files which can be used with the ZAP fuzzer

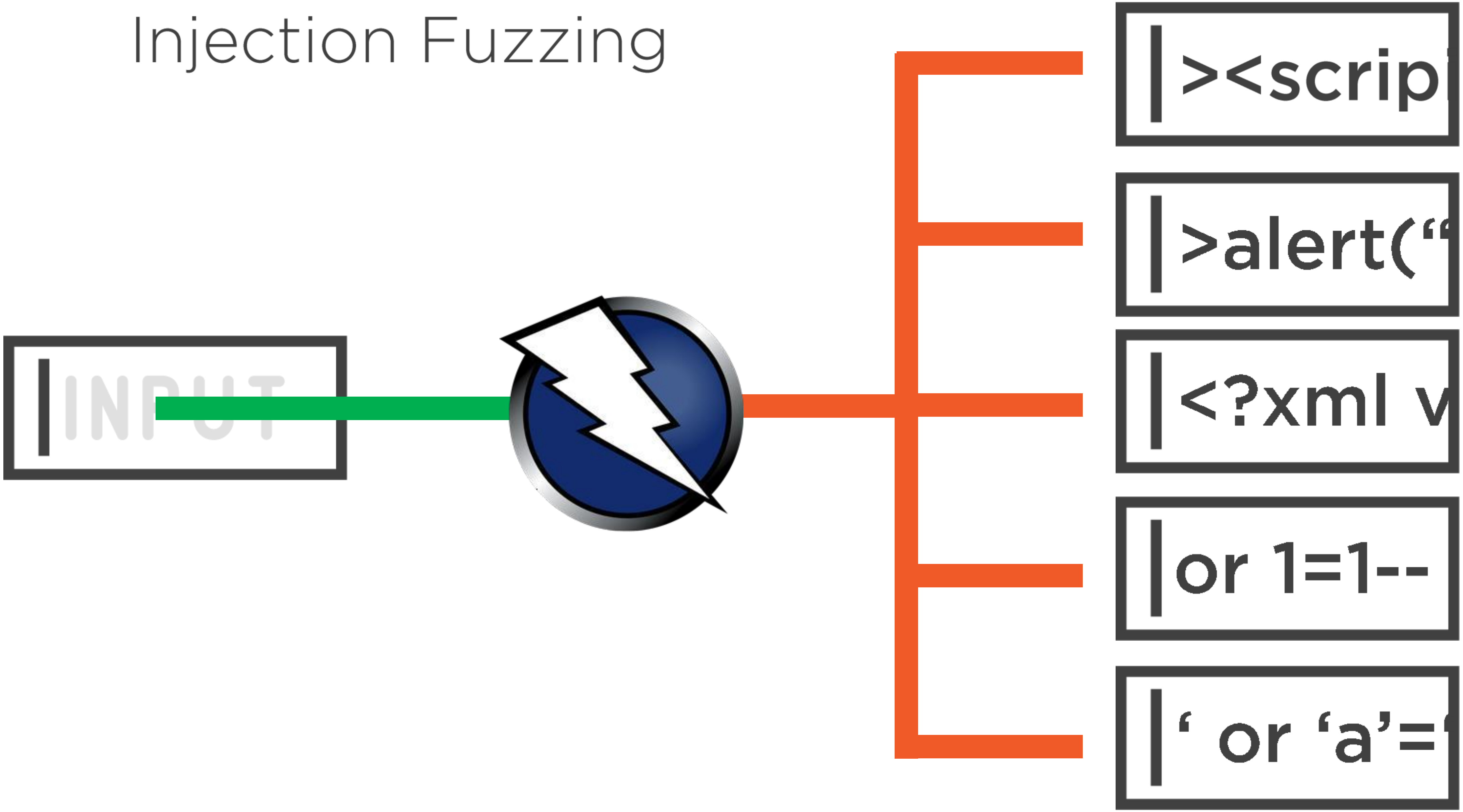
Status: Release

Note: May cause issues with popular antivirus

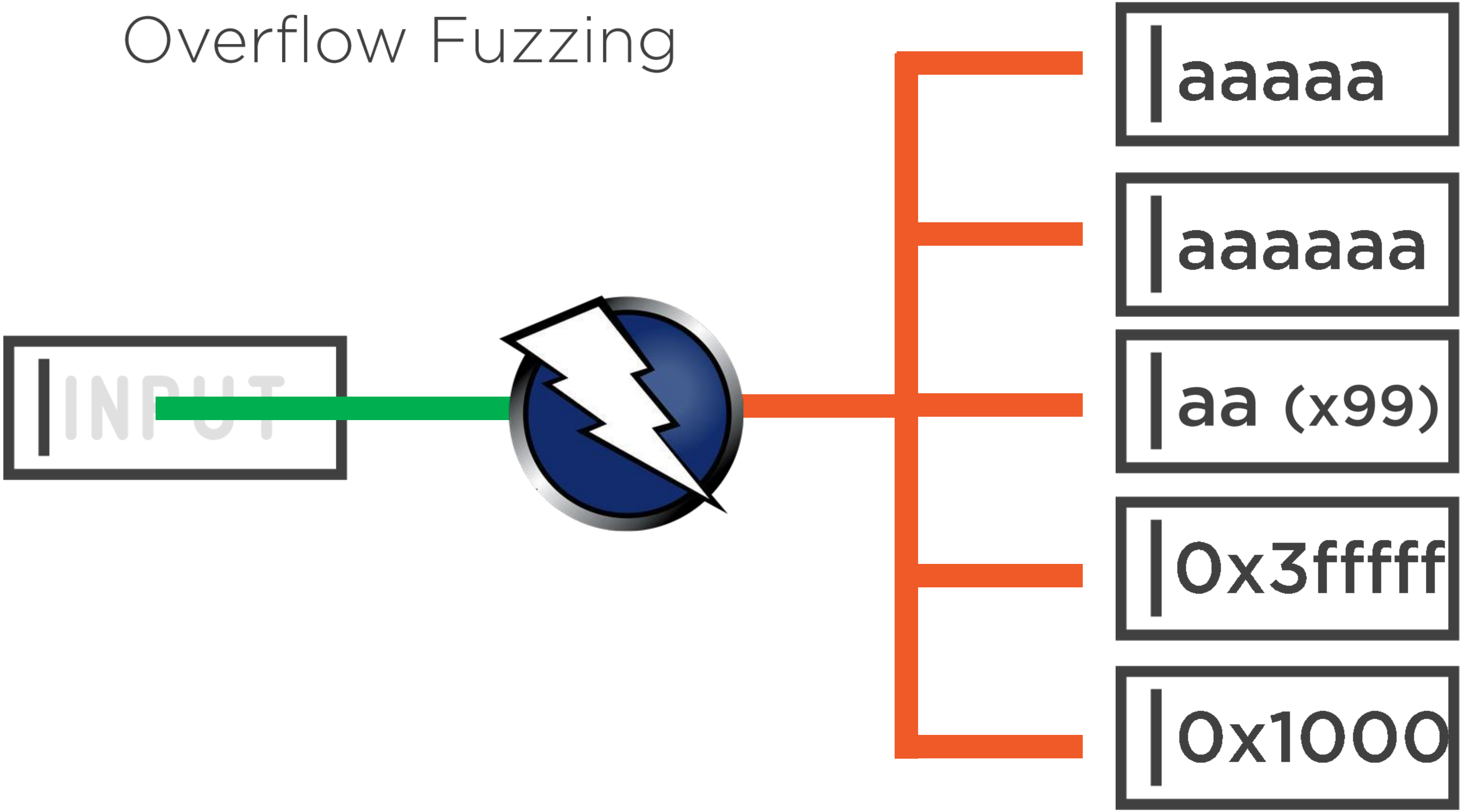
Replacement Fuzzing



Injection Fuzzing



Overflow Fuzzing

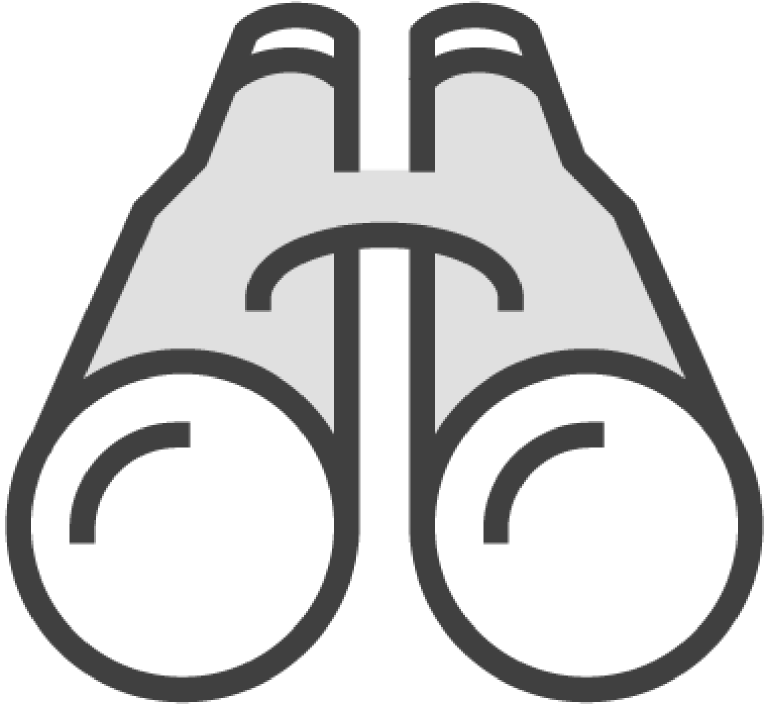


Demo

Fuzzing

Forced Browsing

Function and Purpose



example.com/user/123456
example.com/user/123457
example.com/user/123458

/system/
/password/
/logs/
/admin/
/test

Demo

Forced Browsing

Port Scan Host

Port Scanning

21	22	23	25	79	80	110	113	119	135	139	143
389	443	445	1002	1024	1025	1026	1027	1028	1029	1030	1720

Demo

Port Scanning

Summary

We are now getting into the functions that can get you into legal trouble

Passive scans are still safe. They run automatically in the background

Active scans begin to manipulate pieces of the site which can be construed as hacking

Summary

Quick Start

Fuzzer

Forced Browsing

Port Scanning