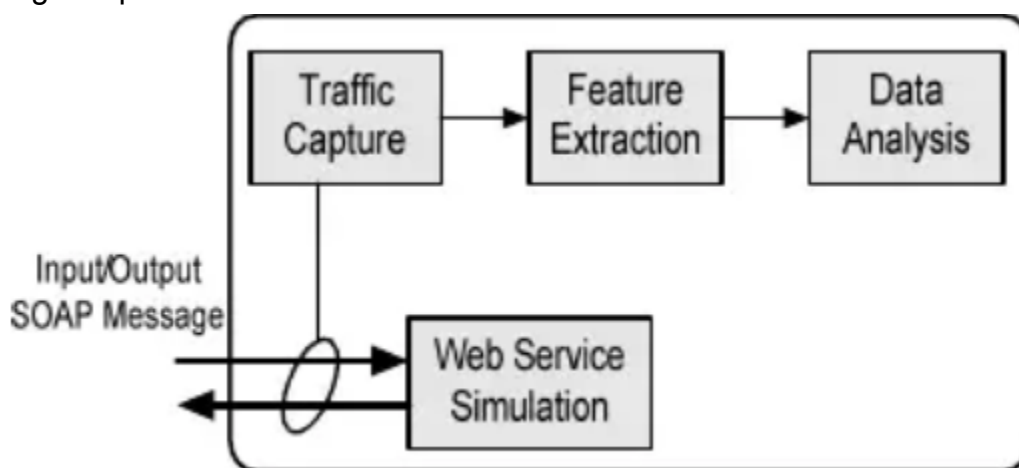


Web Application Honeypots

1 Architectural Design of Web Honeypot

The role of the WS Honeypot is to simulate the behavior of a Web service. It incorporates automated tools to capture and analyze clients activities, especially those issued from attackers. The system architecture is shown in Figure 1. It consists of following components:



1.1 Web Service Simulation

The purpose of this module is to convince attackers that they are interacting with a real Web service. For this reason, we choose to integrate in the Honeypot two Web service servers: Axis and .NET. The administrator can choose which implementation to use. To avoid giving a static appearance to the operations offered by the simulated service, we added an automatic tool that creates from a WSDL (Web Services Description Language) file, a real service that can be deployed in the honeypot. The administrator will be interested in only customizing responses to be returned by the service operations.

1.2 Traffic Capture

Traffic logging is an important task to collect and classify client activities. This component includes traffic capturing mechanisms and monitoring tools to intercept and parse requests and responses for Web services simulated on the WS Honeypot. The Web service requests are formatted in XML language and encapsulated in SOAP

messages which use HTTP as the transport protocol. The content inspection of these messages is necessary to detect attacks.

1.3 Feature Extraction

During this step, we extract features from each SOAP message captured by the WS Honeypot. These parameters will then be useful to classify activities related to these messages. For this purpose, we created three categories of parameters:

SOAP message content: IP source, message length, request preamble, response preamble, invoked operations in each request, input parameters of every operation.

Resource consumption: response time, CPU usage, memory usage.

Operations list:

The Web service offers several operations that can be called differently depending on the user. For each user, we will extract a list of all invoked operations to build a profile and analyze it afterward.

1.4 Data analysis

In honeypots, the analysis is a very difficult and tedious task that requires much effort due to the great amount of data which has to be examined by the human expert. To facilitate this task, we chose to develop an analysis tool based on machine learning techniques to detect any abnormal activities in the honeypot. This tool is very useful for detecting new attacks and speeding up the analysis of normal activities or attacks already seen.

=====

The architectural design of a web honeypot can vary depending on the specific goals and requirements of the deployment. However, here is a generalized architectural design for a web honeypot:

1. Isolation: The honeypot should be isolated from the production environment to prevent any potential compromise from affecting legitimate systems. It is typically deployed in a separate network segment or virtualized environment.

2. Front-End Server: A front-end server acts as the entry point for incoming requests. It can be a real web server or a specialized honeypot server designed to emulate

common web server behaviors. The front-end server handles initial interactions with attackers and forwards requests to the appropriate components.

3. Application Emulation: This component emulates the behavior of a web application or website, including the exposed vulnerabilities. It can be a modified version of an existing open-source application, a custom-built application, or a combination of both. The application emulation component may include a database, file system, and other components typically found in a web application.

4. Logging and Monitoring: Extensive logging and monitoring are crucial in a web honeypot to capture as much information as possible about attacker activities. This includes recording network traffic, requests, responses, user interactions, and any attempted exploits. These logs are analyzed to understand attack techniques and gather intelligence.

5. Intrusion Detection/Prevention System (IDS/IPS): An IDS/IPS system can be deployed to monitor network traffic and detect and block malicious activities. It helps in real-time detection and prevention of attacks against the honeypot.

6. Alerting and Notification: The honeypot should have a mechanism to alert system administrators or security teams when an attack is detected or suspicious activity occurs. This allows for timely response and investigation.

7. Decoy Content: The honeypot can include decoy content to attract attackers. This could include fake user accounts, enticing files, or simulated sensitive data to entice attackers to interact with the honeypot.

8. Honeytokens: Honeytokens are decoy elements strategically placed within the honeypot. These can be fake credentials, hidden links, or other resources that are likely to be accessed or exploited by attackers. Any interaction with honeytokens can be a strong indicator of an attacker's presence.

9. Analysis and Forensics: The captured data and logs are analyzed to understand attack patterns, techniques, and attacker motivations. This information helps in enhancing security measures, updating threat intelligence, and developing countermeasures.

10. Security Controls: Strong security controls, such as firewalls, access controls, and encryption, should be implemented to protect the honeypot from unauthorized access or compromise.

It is important to note that the architecture of a web honeypot can be customized based on specific goals, desired level of interaction, and the expertise of the team managing the honeypot. The key is to create an environment that appears attractive to attackers while ensuring the honeypot is secure, isolated, and monitored effectively.

2. Understanding Frontend and backend technology

Frontend and Backend are the two most popular terms used in web development. The front end is what users see and interact with and the backend is how everything works. Each side needs to communicate and operate effectively with the other as a single unit to improve the website's functionality.

- The front end is the part of the website users can see and interact with such as the graphical user interface (GUI) and the command line including the design, navigating menus, texts, images, videos, etc. The backend, on the contrary, is part of the website users cannot see and interact with.
- The visual aspects of the website that can be seen and experienced by users are frontend. On the other hand, everything that happens in the background can be attributed to the backend.
- Languages used for the front end are HTML, CSS, and JavaScript while those used for the back end include Java, Ruby, Python, and .Net.

Front End Development

The part of a website that the user interacts with directly is termed the front end. It is also referred to as the 'client side of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and a navigation menu. HTML, CSS, and JavaScript are the languages used for Front End development. Responsiveness and performance are the two main objectives of the Front End. The developer must ensure that the site is responsive i.e. it appears correctly on devices of all sizes no part of the website should behave abnormally irrespective of the size of the screen.

Front End Languages

The front-end portion is built by using some languages which are discussed below:

- **[HTML](#)**: HTML stands for Hypertext Markup Language. It is used to design the front-end portion of web pages using a markup language. HTML is a combination of Hypertext and Markup language. Hypertext defines the link between web pages. You can learn this language with Geeksforgeeks [Advanced HTML – Self-Paced course](#) and master the concepts of advanced HTML.
- **[CSS](#)**: Cascading Style Sheets fondly referred to as CSS is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. Also, if you want to enhance your skills then enroll in Geeksforgeeks [CSS Foundation – Self-Paced course](#) and learn all new concepts of CSS.
- **[JavaScript](#)**: JavaScript is a famous scripting language used to create magic on sites to make the site interactive for the user. It is used to enhance the functionality of a website to run cool games and web-based software. Applicable in both front-end and back-end, Javascript is key to becoming a good developer. So start your web-development journey with Geeksforgeeks [JavaScript Foundation – A self-Paced course](#) today.

There are many other languages through which one can do front-end development depending upon the framework for example Flutter uses Dart, React uses JavaScript and Django uses Python, and much more.

Front-End Frameworks and Libraries:

- **[AngularJS](#)**: AngularJs is a JavaScript open-source front-end framework that is mainly used to develop single-page web applications(SPAs). It is a continuously growing and expanding framework which provides better ways for developing web applications. It changes the static HTML to dynamic HTML. It is an open-source project which can be free. It extends HTML attributes with Directives, and data is bound with HTML.

- **[React.js](#)**: React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based front-end library responsible only for the view layer of the application. It is maintained by Facebook. Moreover, React Js makes Front-end development very easy. You can now develop industry-ready Web Applications by enrolling in Geeksforgeeks **[React JS \(Basic to Advanced\) – A self-Paced course](#)**.

Bootstrap: Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

- **[jQuery](#)**: jQuery is an open-source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript. Elaborating on the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.
- **[SASS](#)**: It is the most reliable, mature, and robust CSS extension language. It is used to extend the functionality of an existing CSS of a site including everything from variables, inheritance, and nesting with ease.
- **[Flutter](#)**: Flutter is an open-source UI development SDK managed by google. It is powered by the Dart programming language. It builds performant and good-looking natively compiled applications for mobile (ios, Android), web, and desktop from a single code base. The key selling point of flutter is flat development is made easier, more expressive, and flexible with UI and native performance. In march 2021 flutter announce Flutter 2 which upgrades flutter to build release applications for the web, and the desktop is in beta state.
- Some other libraries and frameworks are Semantic-UI, Foundation, Materialize, Backbone.js, Ember.js, etc.

Back End Development

The backend is the server side of the website. It stores and arranges data, and also makes sure everything on the client side of the website works fine. It is part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend.

Back End Languages

The back-end portion is built by using some languages which are discussed below:

- **[PHP](#)**: PHP is a server-side scripting language designed specifically for web development. Since PHP code is executed on the server side, it is called a server-side scripting language.
- **[C++](#)**: It is a general-purpose programming language and is widely used nowadays for competitive programming. It is also used as a backend language. So if you are curious about learning C++ then you can take help from Geeksforgeeks [C++ Programming Foundation – Self-Paced course](#) and learn all the basics of the language without hassle.
- **[Java](#)**: Java is one of the most popular and widely used programming languages and platforms. It is highly scalable. Java components are easily available and for learning this one of the most popular languages you can check the Geeksforgeeks [Java Programming Foundation – Self-Paced course](#). It will help you understand the proper framework, concepts, functions, and more.
- **[Python](#)**: Python is a programming language that lets you work quickly and integrate systems more efficiently. It is also a very important language for the back end and for mastering it you can take a look at [Python Programming Foundation -Self-Paced course](#). This is a beginner-friendly course and will help you to build a strong foundation for python.

- [Node.js](#): Node.js is an open-source and cross-platform runtime environment for executing JavaScript code outside a browser. You need to remember that NodeJS is not a framework, and it's not a programming language. Most people are confused and understand it's a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It's used in production by large companies such as Paypal, Uber, Netflix, Walmart, and so on.

Back-End Frameworks:

- [Express](#) – Express is a Nodejs framework used for backend/server-side development. It is used to build single-page, multi-page, and hybrid web applications. With its help, you can handle multiple different HTTP requests.
- [Django](#) – Django is a Python web-based framework, following the model-template-views pattern. It is used to build large and complex web applications. Its features include being fast, secure, and scalable.
- [Ruby on Rails](#) – Ruby on Rails is a server-side framework following the model-view-controller architecture pattern. It provides default structures such as web services, web pages, and databases.
- [Laravel](#) – Laravel is a web application framework for PHP and is robust. The feature which makes it perfect is reusing the components of different frameworks for creating a web application.
- [Spring](#) – This server-side framework provides infrastructure support for Java applications. It acts as a support to various frameworks like Hibernate, Struts, EJB, etc. It also has extensions that help in developing Java applications quickly and easily.
- Some more back-end programming/scripting languages are [C#](#), [Ruby](#), [GO](#), etc.

3. Dynamic Web Application Frameworks

A **web framework (WF)** or **web application framework (WAF)** is a [software framework](#) that is designed to support the development of [web applications](#) including web services, [web resources](#), and [web APIs](#). Web frameworks provide a standard way to build and deploy web applications on the [World Wide Web](#). Web frameworks aim to automate the overhead associated with common activities performed in [web development](#). For example, many web frameworks provide [libraries](#) for [database access](#), [templating](#) frameworks, and [session](#) management, and they often promote [code reuse](#).^[1] Although they often target development of [dynamic web sites](#), they are also applicable to [static websites](#).

History

As the design of the [World Wide Web](#) was not inherently dynamic, early [hypertext](#) consisted of hand-coded [HTML](#) text files that were published on [web servers](#). Any modifications to published pages needed to be performed by the pages' author. In 1993, the [Common Gateway Interface](#) (CGI) standard was introduced for interfacing external applications with web servers, to provide a [dynamic web page](#) that reflected user inputs.^[3]

Original implementations of the CGI interface typically had adverse effects on the server load however, because each request started a separate [process](#).^[4] More recent implementations utilize persistent processes amongst other techniques to reduce the footprint in the server's resources and offer a general performance boost.^[citation needed]

In 1995, fully integrated server/language development environments first emerged and new web-specific languages were introduced, such as [ColdFusion](#), [PHP](#), and [Active Server Pages](#).^[citation needed]

Although the vast majority of languages for creating dynamic web pages have [libraries](#) to help with common tasks, web applications often require specific libraries for particular tasks, such as creating [HTML](#) (for example, [Jakarta Server Faces](#)).^[citation needed]

In the late 1990s, mature, "full stack" frameworks began to appear, that often gathered multiple libraries useful for [web development](#) into a single cohesive [software stack](#) for web developers to use. Examples of this include [ASP.NET](#), [Java EE](#), [WebObjects](#), [web2py](#), [OpenACS](#), [Catalyst](#), [Mojolicious](#), [Ruby on Rails](#), [Laravel](#), [Grails](#), [Django](#), [Zend Framework](#), [Sails.js](#), [Yii](#),^[5] [CakePHP](#),^[6] and [Symfony](#).

Types of framework architectures

Most web frameworks are based on the [model-view-controller](#) (MVC) [pattern](#).

Model-view-controller (MVC)

Reference article: [Model-view-controller](#)

Many frameworks follow the MVC [architectural pattern](#) to separate the [data model](#) into [business rules](#) (the "controller") and the [user interface](#) (the "view"). This is generally considered a good practice as it [modularizes code](#), promotes [code reuse](#), and allows multiple interfaces to be applied. In web applications, this permits different views to be presented, for example serving different [web pages](#) for mobile vs. desktop browsers, or providing machine-readable [web service](#) interfaces.

Push-based vs. pull-based

Most MVC frameworks follow a push-based architecture also called "action-based". These frameworks use actions that do the required processing, and then "push" the data to the view layer to render the results.^[7] [Django](#), [Ruby on Rails](#), [Symfony](#), [Spring MVC](#), [Stripes](#), [Sails.js](#), [CodeIgniter](#)^[8] are good examples of this architecture. An alternative to this is pull-based architecture, sometimes also called "component-based". These frameworks start with the view layer, which can then "pull" results from multiple controllers as needed. In this architecture, multiple controllers can be involved with a single view. [Lift](#), [Tapestry](#), [JBoss Seam](#), [Jakarta Server Faces](#), and [Wicket](#) are examples of pull-based architectures. [Play](#), [Struts](#), RIFE, and [ZK](#) have support for both push- and pull-based application controller calls.

Three-tier organization

In a [three-tier organization](#), applications are structured around three physical tiers: client, application, and database. The database is normally an [RDBMS](#). The application contains the business logic, running on a server and communicating with the client using [HTTP](#). The client on web applications is a web browser that runs HTML generated by the application layer. The term should not be confused with MVC, where, unlike in three-tier architecture, it is considered a good practice to keep business logic away from the controller, the "middle layer".

Framework applications

Frameworks are built to support the construction of internet applications based on a single programming language, ranging in focus from general purpose tools such as Zend Framework and Ruby on Rails, which augment the capabilities of a specific language, to native-language programmable packages built around a specific user application, such as [content management systems](#) (CMS), some mobile development tools and some portal tools.

General-purpose website frameworks

Web frameworks must function according to the architectural rules of browsers and [protocols](#) such as [HTTP](#), which is [stateless](#). Webpages are served up by a [server](#) and can then be modified by the browser using [JavaScript](#). Either approach has its advantages and disadvantages.^{[\[citation needed\]](#)}

Server-side page changes typically require that the page be refreshed, but allow any language to be used and more computing power to be utilized. Client-side changes allow the page to be updated in small chunks which feels like a desktop application, but are limited to JavaScript and run in the user's browser, which may have limited computing power. Some mix of the two is typically used.^{[\[19\]](#)} Applications which make heavy use of JavaScript and only refresh parts of the page, are called [single-page applications](#) and typically make use of a client-side JavaScript web framework to organize the code

Server-side

- [Apache Wicket](#)
- [ASP.NET Core](#)
- [CakePHP](#)
- [Catalyst](#)
- [CodeIgniter](#)
- [CppCMS](#)
- [Django](#)
- [Flask](#)
- [Grails](#)
- [Jam.py](#)
- [Laravel](#)
- [Mojolicious](#)
- [Ruby on Rails](#)
- [Sails.js](#)
- [Symfony](#)

- [Spring MVC](#)
- [VIEwoNLY](#)
- [Wt \(web toolkit\)](#)
- [Yii](#)
- [Zend Framework](#)

Client-side

Examples include [Backbone.js](#), [AngularJS](#), [Angular](#), [EmberJS](#), [ReactJS](#), [jQuery UI](#), [Svelte](#), and [Vue.js](#)

Top 10 Frameworks for Web Applications

1. Ruby on Rails

Ruby on Rails is an extremely productive web application framework written by David Heinemeier Hansson. One can develop an application at least ten times faster with Rails than a typical Java framework. Moreover, Rails includes everything needed to create a database-driven web application, using the Model-View-Controller pattern.

- **Language:** [Ruby](#)
- **Latest Version:** Rails 5.0.0.beta2
- **Framework Link:** <http://rubyonrails.org>
- **Github Link:** <https://github.com/rails/rails>

Websites using Ruby on Rails are GroupOn, UrbanDictionary, AirBnb, Shopify, Github

2. [Django](#)

Django is another framework that helps in building quality web applications. It was invented to meet fast-moving newsroom deadlines while satisfying the tough requirements of *experienced Web developers*. Django developers say the applications are ridiculously fast, secure, scalable, and versatile.

- **Language:** Python
- **Latest Version:** Django 1.9.2
- **Framework Link:** <https://www.djangoproject.com>
- **Github Link:** <https://github.com/django/django>

Websites using Django are Disqus, Pinterest, Instagram, Quora, etc.

3. Angular(Also, know as Angular JS)

Angular is a framework by Google (originally developed by Misko Hevery and Adam Abrons) which helps us in building powerful Web Apps. It is a framework to build large scale and high-performance web applications while keeping them as easy-to-maintain. There are a huge number of web apps that are built with Angular.

- **Language:** JavaScript
- **Latest Version:** Angular 7.1.5
- **Framework Link:** <https://angular.io/>
- **Github Link:** <https://github.com/angular/angular>

Websites using Angular are Youtube on PS3, Weather, Netflix, etc.

4. ASP.NET

ASP.NET is a framework developed by Microsoft, which helps us to build robust web applications for PC, as well as mobile devices. It is a high performance and lightweight framework for building Web Applications using .NET. All in all, a framework with Power, Productivity, and Speed.

- **Language:** [C#](#)
- **Latest Version:** ASP.NET 5 (ASP.NET Core 1.0)
- **Framework Link:** <http://www.asp.net/>

Websites using ASP.NET are GettyImages, TacoBell, StackOverflow, etc.

5. METEOR

Meteor or MeteorJS is another framework that gives one a radically simpler way to build real time mobile and web apps. It allows for rapid prototyping and produces cross-platform (Web, Android, iOS) code. Its cloud platform, Galaxy, greatly simplifies deployment, scaling, and monitoring.

- **Language:** JavaScript
- **Latest Version:** Meteor 1.2.1
- **Framework Link:** <https://www.meteor.com/>
- **Github Link:** <https://github.com/meteor/meteor>

Websites using Meteor are HagggleMate, WishPool, Telescope, etc.

6. [Laravel](#)

Laravel is a framework created by Taylor Otwell in 2011 and like all other modern frameworks, it also follows the MVC architectural pattern. Laravel values Elegance, Simplicity, and Readability. One can right away start learning and developing Laravel with Laracasts which has hundreds of tutorials in it.

- **Language:** PHP
- **Latest Version:** Laravel 5.2
- **Framework Link:** <https://laravel.com/>
- **Github Link:** <https://github.com/laravel/laravel>

Websites using Laravel are Deltanet Travel, Neighbourhood Lender, etc.

7. [Express](#)

Express or Expressjs is a minimal and flexible framework that provides a robust set of features for web and mobile applications. It is relatively minimal meaning many features are available as plugins. Express facilitates the rapid development of Node.js based Web applications. Express is also one major component of the MEAN software bundle.

- **Language:** JavaScript
- **Framework Link:** <http://expressjs.com/>
- **Github Link:** <https://github.com/strongloop/express>

Websites using Express are Storify, Myspace, LearnBoost, etc.

8. [Spring](#)

Spring, developed by Pivotal Software, is the most popular application development framework for enterprise Java. Myriads of developers around the globe use Spring to create high performance and robust Web apps. Spring helps in creating simple, portable, fast, and flexible JVM-based systems and applications.

- **Language:** Java
- **Latest Version:** Spring 4.3.0
- **Framework Link:** <http://projects.spring.io/spring-framework/>
- **Github Link:** <https://github.com/spring-projects/spring-framework>

Websites using spring are Mascus, Allocine, etc.

9. PLAY

Play is one of the modern web application framework written in Java and Scala. It follows the MVC architecture and aims to optimize developer productivity by using convention over configuration, hot code reloading, and display of errors in the browser. Play quotes itself as “The High-Velocity Web Framework”.

- **Language:** Scala and Java
- **Latest Version:** Play 2.4.6
- **Framework Link:** <https://www.playframework.com/>
- **Github Link:** <https://github.com/playframework/playframework>

Websites using PLAY are LinkedIn, Coursera, LendUp, etc.

10. [CodeIgniter](#)

CodeIgniter, developed by EllisLab, is a famous web application framework to build dynamic websites. It is loosely based on MVC architecture since Controller classes are necessary but models and views are optional. CodeIgniter promises with exceptional performance, nearly zero-configuration, and no large-scale monolithic libraries.

- **Language:** PHP
- **Latest Version:** CodeIgniter 3.0.4
- **Framework Link:** <https://codeigniter.com/>
- **Github Link:** <https://github.com/EllisLab/CodeIgniter>

Websites using CodeIgniter are Bufferapp, The Mail and Guardian, etc. Apart from these 10 frameworks, others like [Symfony](#) , [Ember.js](#), [Sails.js](#), [React.js](#) are also worth mentioning.

4.Set Up and Configuration Of The Honeypot To Model A Vulnerable System

NOTE: you can use windows also . but i'm using Linux mint.you can use any ubuntu/debian os

There are Lots Of Web Honeypot Available,

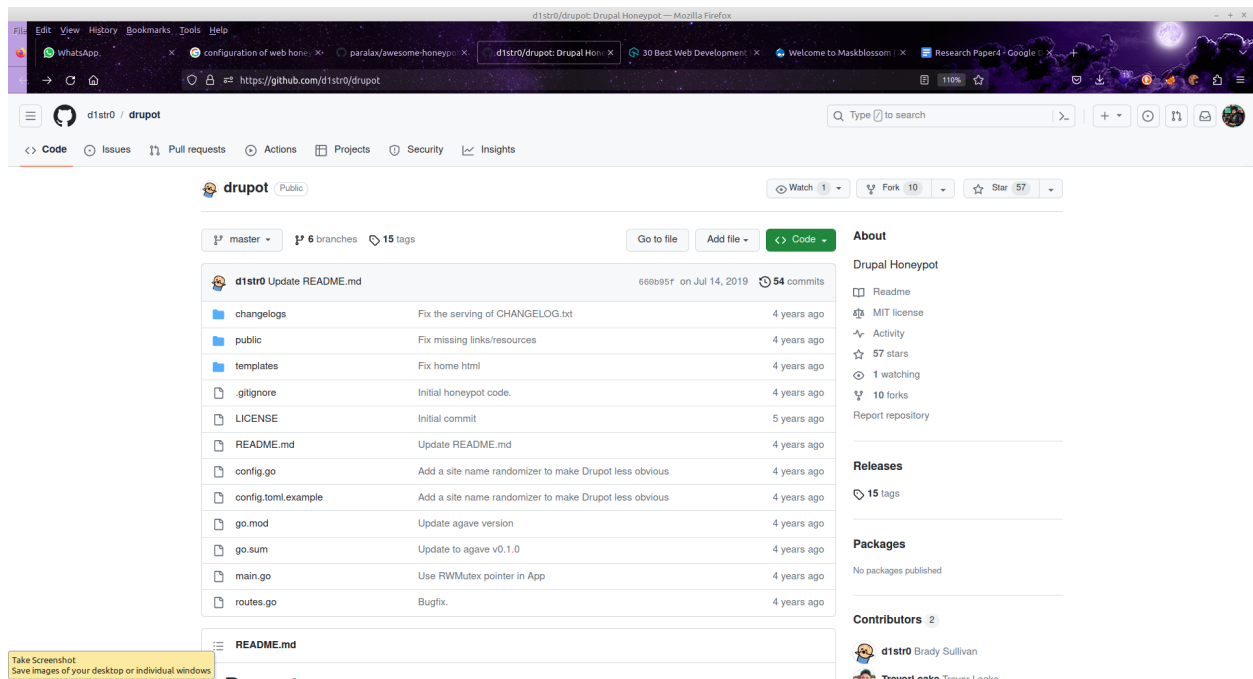
Go And Visit This Github Repo [Honeypot Tools](#)

Visit Web Honeypot Section

- Web honeypots

- [Express honeypot](#) - RFI & LFI honeypot using nodeJS and express.
- [EoHoneypotBundle](#) - Honeypot type for Symfony2 forms.
- [Glastopf](#) - Web Application Honeypot.
- [Google Hack Honeypot](#) - Designed to provide reconnaissance against attackers that use search engines as a hacking tool against your resources.
- [HellPot](#) - Honeypot that tries to crash the bots and clients that visit it's location.
- [Laravel Application Honeypot](#) - Simple spam prevention package for Laravel applications.
- [Nodepot](#) - NodeJS web application honeypot.
- [PasitheaHoneypot](#) - RestAPI honeypot.
- [Servletpot](#) - Web application Honeypot.
- [Shadow Daemon](#) - Modular Web Application Firewall / High-Interaction Honeypot for PHP, Perl, and Python apps.
- [StrutsHoneypot](#) - Struts Apache 2 based honeypot as well as a detection module for Apache 2 servers.
- [WebTrap](#) - Designed to create deceptive webpages to deceive and redirect attackers away from real websites.
- [basic-auth-pot \(bap\)](#) - HTTP Basic Authentication honeypot.
- [bwpot](#) - Breakable Web applications honeyPot.
- [django-admin-honeypot](#) - Fake Django admin login screen to notify admins of attempted unauthorized access.
- [drupo](#) - Drupal Honeypot.
- [honeyhttpd](#) - Python-based web server honeypot builder.
- [honeyup](#) - An uploader honeypot designed to look like poor website security.
- [owa-honeypot](#) - A basic flask based Outlook Web Honey pot.
- [phpmyadmin_honeypot](#) - Simple and effective phpMyAdmin honeypot.
- [shockpot](#) - WebApp Honeypot for detecting Shell Shock exploit attempts.
- [smart-honeypot](#) - PHP Script demonstrating a smart honey pot.
- [Snare/Tanner](#) - successors to Glastopf
 - [Snare](#) - Super Next generation Advanced Reactive honeypot.
 - [Tanner](#) - Evaluating SNARE events.
- [stack-honeypot](#) - Inserts a trap for spam bots into responses.
- [tomcat-manager-honeypot](#) - Honeypot that mimics Tomcat manager endpoints. Logs requests and saves attacker's WAR file for later study
- WordPress honeypots
 - [HonnyPotter](#) - WordPress login honeypot for collection and analysis of failed login attempts.
 - [HoneyPress](#) - Python based WordPress honeypot in a Docker container.
 - [wp-smart-honeypot](#) - WordPress plugin to reduce comment spam with a smarter honeypot.
 - [wordpot](#) - WordPress Honeypot.
- [Python-Honeypot](#) - OWASP Honeypot, Automated Deception Framework.

I'm Going To Use [Drupot Tool](#)



Install Using Git Clone Command or Install Using Given Instructions

Installation

Drupot supports go modules.

```
go get github.com/d1str0/drupal-honeypot
```

```
go build
```

Running Drupot

```
./drupot -c config.toml
```


Configuration

`config.toml.example` contains an example of *all* currently available configuration options.

Method For Latest Version , Click On Release Section

About

Drupal Honeypot

 Readme

 MIT license

 Activity

 57 stars

 1 watching

 10 forks

Report repository





















Releases

 15 tags

Install Latest Package For Updated Honeypot

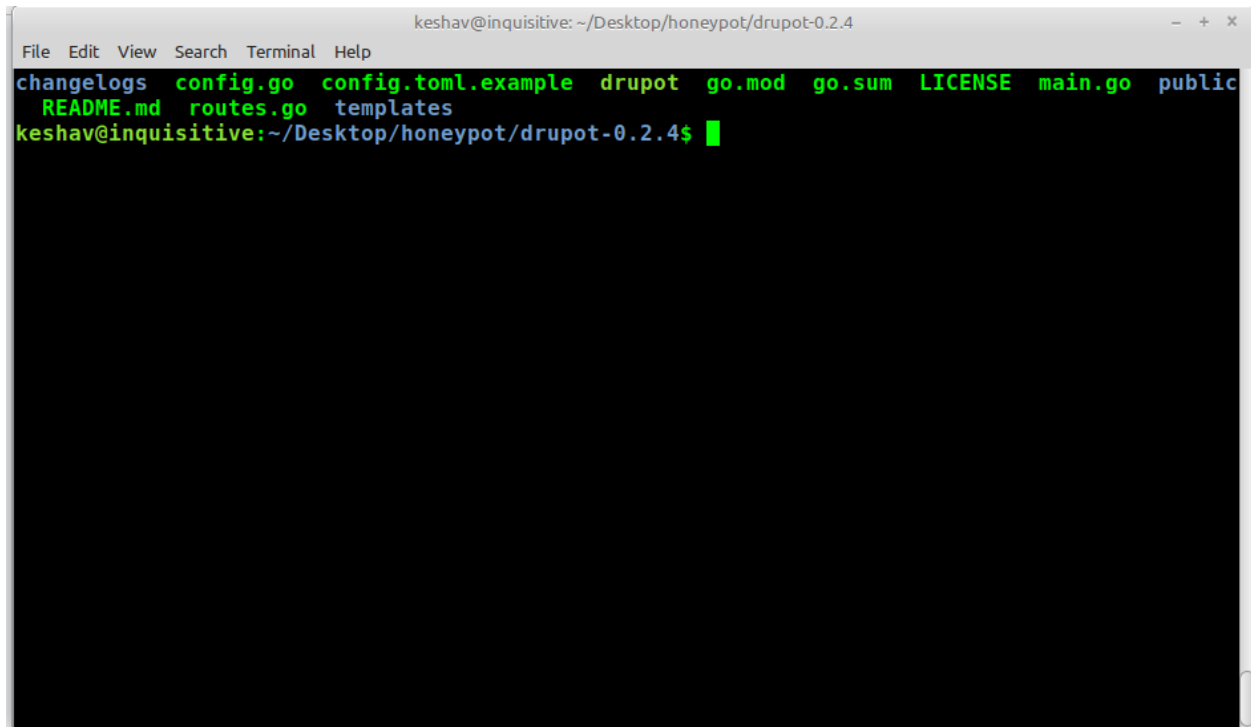
OR

Install Any Outdated Package To Build Vulnerable Honeypot

Tags	
v0.2.4 ...	 on Jun 19, 2019  c1c046f  zip  tar.gz
v0.2.3 ...	 on Jun 13, 2019  e2f5a67  zip  tar.gz
v0.2.2 ...	 on Jun 12, 2019  e5cfb03  zip  tar.gz
v0.2.1 ...	 on Jun 12, 2019  703ef95  zip  tar.gz
v0.2.0 ...	 on Jun 12, 2019  a7159a8  zip  tar.gz

Open Downloaded Folder In Linux Terminal Terminal.

And type ls in your terminal



```
keshav@inquisitive: ~/Desktop/honeypot/drupot-0.2.4
File Edit View Search Terminal Help
changelogs config.go config.toml.example drupot go.mod go.sum LICENSE main.go public
README.md routes.go templates
keshav@inquisitive:~/Desktop/honeypot/drupot-0.2.4$
```

(If you installed Drupal using go command , type go build before running drupal honeypot)

Installation

Drupot supports go modules.

```
go get github.com/d1str0/drupot
```

```
go build
```

Type Given Command On GitHub

Running Drupot

```
./drupot -c config.toml
```

Configuration

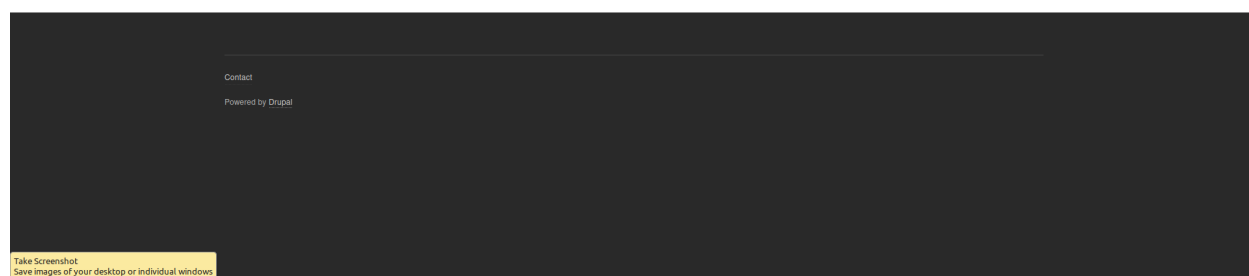
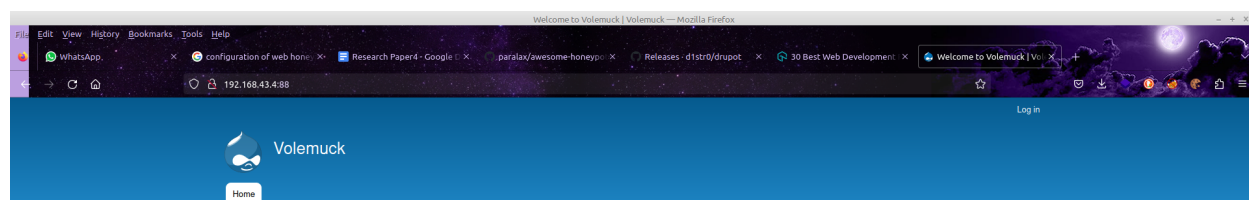
`config.toml.example` contains an example of *all* currently available configuration options.

example

```
4$ sudo ./drupot -c config.toml.example
```

Don't forget to use sudo.

After Running Lets Visit our WebSite, type your machine ip with port 80, in my case i changed it to port 88



Check your terminal also

```
keshav@inquisitive:~/Desktop/honeypot/drupot-0.2.4$ sudo ./drupot -c config.toml.example
[sudo] password for keshav:
Sorry, try again.
[sudo] password for keshav:
///- Running Drupot
///- v0.2.4
///- Loading config file: config.toml.example
2023/06/28 18:25:32 {"protocol":"HTTP/1.1","app":"agave","agave_app":"Drupot","channel":"drupot.ev
ents","sensor":"ffe250aa-15b2-11ee-8706-646c80a9b795","dest_port":88,"dest_ip":"106.193.150.173","
src_port":57616,"src_ip":"192.168.43.4","signature":"","prev_seen":false,"request_json":{"Method":
"GET","URL":{"Scheme":"","Opaque":"","User":null,"Host":"","Path":"/","RawPath":"","ForceQuery":fa
lse,"RawQuery":"","Fragment":""},"Proto":"HTTP/1.1","ProtoMajor":1,"ProtoMinor":1,"Header":{"Accep
t":["text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8"],"Acce
pt-Encoding":["gzip, deflate"],"Accept-Language":["en-US,en;q=0.5"],"Connection":["keep-alive"],"C
ookie":["PHPSESSID=kn3450jjh5tvebqbt49743oe53; security=low"],"Sec-Gpc":["1"],"Upgrade-Insecure-Re
quests":["1"],"User-Agent":["Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/114.
0"]},"Body":"","TransferEncoding":null,"Host":"192.168.43.4:88","PostForm":{}},"agave_client_versi
on":"v0.1.2"}
```

Traffic captured

5. Tools For Web Application Security Testing and Analysis

What is Security Testing?

Security testing is a process to determine whether the system protects data and maintains functionality as intended. Penetration testing or pen testing is also a type of Security testing which is performed to evaluate the security of the system (hardware, software, networks or an information system environment).

We can do security testing using both manual and automated security testing tools and techniques. Security testing reviews the existing system to find vulnerabilities.

Most of the companies perform security testing on newly deployed or developed software, hardware, and network or information system environments. But it's highly recommended by experts to make security testing as a part of the information system audit process of an existing information system environment.

To find the flaws and vulnerabilities in a web application, there are many free, paid, and open source security testing tools available in the market. We know that the advantage of open source tools are we can easily customize it to match our requirements. We are here to showcase some of the top __ open source security testing tools.

We use security testing tools for checking how secure a website or web application is.

Security tests include testing for vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), Session Management, Broken Authentication, Cross-Site Request Forgery (CSRF), Security Misconfiguration, Failure to Restrict URL Access etc.,

Website hacking is quite common nowadays. Every now and then there is some news regarding a website being hacked or data breach. [Infosec](#) (information security) has come a long way and in the same way, hacking too. To keep a website safe from hackers we need to build secure websites to stay away from hackers. Web Security Testing Tools act proactively in detecting web application vulnerabilities and safeguarding websites against attacks. There are many paid and free web application testing tools available in the market. Here, we discuss the top 14 open source security testing tools for web applications.

Here are some of the Commercial and Open Source Security Testing Tools which are popular among Security Testers.

- [#1. Invicti](#)
- [#2. Acunetix](#)
- [#3. Zed Attack Proxy \(ZAP\)](#)
- [#4. Wfuzz](#)
- [#5. Wapiti](#)
- [#6. W3af](#)

- [#7. Vega](#)
- [#8. SQLMap](#)
- [#9. SonarQube](#)
- [#10. Nogotofail](#)
- [#11. Grabber](#)
- [#12. Arachni](#)
- [#13. Skipfish](#)
- [#14. Ratproxy](#)

#1. Invicti

Invicti is a web vulnerability management system. It is an automatic, deadly accurate, and easy-to-use web application security scanner. It is used to automatically identify security issues such as Cross-Site Scripting (XSS) and in websites, web applications, and web services.

Its Proof-based Scanning technology doesn't just report vulnerabilities, it also produces a Proof of Concept to confirm they are not false positives. So there is no point in wasting your time manually verifying the identified vulnerabilities after a scan is finished.

Some of the features of Invicti are as follows

- Vulnerability assessment
- Advanced web scanning
- Proof-based scanning technology for dead-accurate vulnerability detection and scan results
- Full HTML5 support
- Web services scanning
- HTTP request builder
- SDLC integration
- Reporting
- Exploitation
- Manual testing
- Anti-CSRF (Cross-site Request Forgery) token support
- Automatic detection of custom 404 error pages
- REST API support
- Anti-CSRF token support

#2. Acunetix

Acunetix is an easy yet powerful solution to secure your website, web applications and APIs. It detects over 4500 web vulnerabilities such as Cross Site Scripting (XSS), SQL injection, etc.,

Don't miss our detailed review on [Acunetix](#)

Its DeepScan Crawler scans HTML5 websites and AJAX-heavy client-side SPAs. It allows users to export discovered vulnerabilities to issue trackers such as Atlassian JIRA, GitHub. It runs on Windows, Linux, and Online.

Some of the features of Acunetix are as follows

- In-depth crawl and analysis – automatically scans all websites
- The highest detection rate of vulnerabilities with low false positives
- Integrated vulnerability management – prioritize and control threats
- It can be integrated with defect trackers such as JIRA, Bugzilla, or Mantis.
- Free network security scanning and Manual Testing tools
- Supported platforms Windows, Linux, and macOS

#3. Zed Attack Proxy (ZAP)

Zed Attack Proxy popularly known as ZAP is an open source security testing tool for a web application which was developed by OWASP (Open Web Application Security Project). It runs on all operating systems that support Java 8. It is one of the world's most popular free security tools and is actively maintained by volunteers. It is an easy to use integrated penetration testing tool for finding a number of security vulnerabilities in a web application while we are developing and testing an application. It is also a great tool for experienced pentesters to use for manual security testing. It is designed to be used by people with a wide range of security experience and as such is ideal for developers and functional testers who are new to penetration testing as well as experienced security professionals. It comes with a friendly GUI which helps newbies as well as experts. It gives command line access for advanced users.

ZAP has a huge reputation amongst Security Testing Tools as being easy to use, and powerful.

Highlights:

- Easy to use
- Easy to install
- Free, Open source
- Cross-platform
- Internationalized

Key features of ZAP are:

- Automatic scanning

- Rest-based API
- Intercepting proxy
- Authentication Support
- Ajax Spider
- Dynamic SSL Certificates
- SQL Injection
- XSS Injection
- Forced Browsing
- Fuzzing
- Web Socket Support
- Active and Passive scanners
- Cookie-based and HTTP authentication session management
- Anti CSRF token handling

#4. Wfuzz

Wfuzz is a web application security fuzzer tool which is developed in Python. It doesn't come with GUI Interface, so security testers who want to use this tool have to work on command line interface. This tool is designed for brute forcing web applications.

Key features of Wfuzz are:

- Multiple injection points with multiple dictionaries
- Post, headers and authentication data brute forcing
- Output to HTML
- Cookies fuzzing
- Multithreading
- Proxy Support
- SOCK Support
- Time delays between requests
- Authentication Support (NTLM, Basic)
- All parameters brute forcing (POST and GET)
- Multiple encoders per payload
- Baseline request (to filter results against)
- Brute force HTTP methods
- Multiple proxy support (each request through a different proxy)
- HEAD scan (faster for resource discovery)

Website Link: <http://www.edge-security.com/wfuzz.php>

#5. Wapiti

Wapiti is a web application vulnerability scanner. It allows us to audit the security of websites or web applications. It performs black box scans of the web application by crawling the web pages of the deployed webapp, looking for scripts and forms where it can inject data. Once it gets the list of URLs, forms and their inputs, Wapiti acts like fuzzer, injecting payloads to see if a script is vulnerable. This open source security testing tool supports both GET and POST HTTP attack methods. It is a command line application. It doesn't come with GUI. So it is important to have a knowledge of various commands of Wapiti. There is detailed documentation on Wapiti official site.

It detects vulnerabilities like

- File disclosure
- Data injection
- XSS (Cross Site Scripting) injection
- XXE (XML External Entity) injection
- CRLF injection
- SSRF(Server Side Request Forgery)
- Bypass weak .htaccess configurations
- Shell shock (aka Bash Bug)

Key features of Wapiti web vulnerability scanner are:

- Supports both GET and POST HTTP methods for attacks
- Acts like a fuzzer

Website Link: <http://wapiti.sourceforge.net/>

#6. W3af

W3af is a web application attack and audit framework that is developed using python. It is one of the most popular web application security testing frameworks in the market. It comes with both GUI and console interface. It helps developers and penetration testers identify and exploit vulnerabilities in web applications. It supports authentication types such as HTTP basic authentication, NTLM authentication, Form authentication, Cookie authentication. It is able to identify more than 200 types of security issues in web applications, including

- Cross-Site Scripting
- SQL Injection

- Guessable credentials
- Unhandled application errors
- PHP misconfigurations
- Blind SQL injections
- Buffer overflow vulnerability
- CORS (Cross-Origin Resource Sharing)
- CSRF (Cross Site Request Forgeries) vulnerabilities
- OS Commanding
- Authentication support

Website Link: <http://w3af.org/>

#7. Vega

Vega is a free and open source web security scanner and web security testing platform to test the security of web applications. It is written in Java and has a well designed graphical user interface (GUI) that runs on Linux, OS X, and Windows.

It exposes vulnerabilities including

- Find and validate SQL injection
- Cross-Site Scripting (XSS) injection
- Blind SQL injection
- Header injection
- Remote file include
- Shell injection

Website Link: <https://subgraph.com/vega/>

#8. SQLMap

SQLMap is an open source penetration testing tool. It allows us to automate the process of detecting and exploiting SQL injection vulnerabilities in a website's database. It comes with a powerful detection engine and many features to detect vulnerabilities.

It supports 6 types of SQL Injection techniques:

- Boolean-based blind
- Time-based blind
- Error-based
- Union query-based

- Stacked queries
- Out-of-band

It supports a large number of database services such as

- MySQL
- Oracle
- PostgreSQL
- Microsoft SQL Server
- Microsoft Access
- IBM DB2
- SQLite
- Firebird
- Sybase
- SAP
- MaxDB
- Informix
- HSQLDB
- H2

Website Link: <http://sqlmap.org/>

#9. SonarQube

SonarQube is an open source security testing tool developed by SonarSource. It is an automatic code review tool to detect bugs, vulnerabilities and code smells in your code.

Key features of SonarQube are

- Continuous inspection
- Detect Tricky issues
- Multi-Language support
- DevOps Integration
- Centralize Quality

Website Link: <https://www.sonarqube.org/>

#10. Nogotofail

Nogotofail is a network security testing tool (network vulnerability scanner tool) designed to help developers and penetration testers. As a network security scanner, it

includes testing for common SSL certificate verification issues, HTTPS and TLS/SSL library bugs, SSL and STARTTLS stripping issues, cleartext issues, and more.

Vulnerabilities exposed by Nogotofail network testing tool are

- SSL Injection
- TLS Injection
- SSL Certificate verification issues
- SSL and STARTTLS stripping issues
- Cleartext issues

Website

Link:

<https://security.googleblog.com/2014/11/introducing-nogotofaila-network-traffic.html>

[Download Nogotofail](#)

#11. Grabber

Grabber is an open source web application scanner that detects some kind of vulnerabilities in a website or web applications. It is designed to scan small websites such as forums and personal websites. It is absolutely not for big application. It will take a too long time and flood your network when you use it for a big application. It doesn't come with GUI interface. It was developed in Python.

Grabber can identify the following issues:

- Cross-site scripting
- SQL injection
- File inclusion
- Backup files check
- Simple AJAX check
- Hybrid analysis or Crystal ball testing for PHP application using PHP-SAT

Website Link: <https://tools.kali.org/web-applications/grabber>

[Download Grabber](#)

#12. Arachni

Arachni is an open source security testing tool aimed towards helping penetration testers and administrators evaluate the security of web applications. It is a feature-full, modular, high-performance Ruby framework. It supports all major operating systems such as MS Windows, Mac OS X, and Linux. It is designed to identify security issues within a web application and make it hacker proof.

Arachni can identify the following issues:

- Local file inclusion
- Remote file inclusion
- Invalidated redirects
- Invalidated DOM redirects
- XPath injection
- SQL injection
- XSS injection

Website Link: <http://www.arachni-scanner.com/>

[Download Arachni](#)

#13. Skipfish

Skipfish is an active web application security testing tool. It prepares an interactive sitemap for the targeted site by carrying out a recursive crawl and dictionary-based probes. It is available for Linux, Mac OS X, and Windows.

Some of the security checks offered by Skipfish are:

- Server-side query injection
- Explicit SQL-like syntax in GET or POST parameters
- Server-side shell command injection
- Server-side XML/XPath injection
- Password forms submitting from or to non-SSL pages
- Incorrect or missing MIME types on renderable

Website Link: <https://tools.kali.org/web-applications/skipfish>

[Download Skipfish](#)

#14. Ratproxy

Ratproxy is an open source security testing tool. It is a semi-automated, largely passive web application security audit tool. Ratproxy assessments take little bandwidth or time to run and proceed in an intuitive, distraction-free manner. It affords a consistent and predictable coverage of user-accessible features. It is supported by all popular operating systems such as Mac OS X, Windows, and Linux.

Website Link: <https://sectools.org/tool/ratproxy/>

Reference

Design:

https://www.academia.edu/33775156/Design_and_Implementation_of_Web_Service_Honeypot

Frontend or Backend:

<https://www.geeksforgeeks.org/frontend-vs-backend/> ,

https://en.wikipedia.org/wiki/Frontend_and_backend

Web framework:

https://en.wikipedia.org/wiki/Web_framework#Client-side ,

<https://www.lambdatest.com/blog/best-web-development-frameworks/>

Top 10 Web

Framework: <https://www.geeksforgeeks.org/top-10-frameworks-for-web-applications/>

Web Application Testing Tools:

<https://www.softwaretestingmaterial.com/open-source-security-testing-tools/> ,

<https://www.cigniti.com/blog/10-open-source-web-security-testing-tools/> ,

<https://www.softwaretestinghelp.com/open-source-security-testing-tools/>