**CYBAROGUE**

**Index**

1. Introduction

1.1. Cybarogue Flow diagram

1.2. Cybarogue Key Features

1.3. Basic Steps to connect with Cybarogue

2. Target Audience of the document

2.1. Completed Tool Assessment-Phase 1

2.2. Current Tool Assessment-Phase 2

2.3. Product flow diagram of Cybarogue

2.4. Updated Tabular form

2.5. Milestones Planned- Phase 3

3. Purpose of the document

3.1. Architecture block diagram of Cybarogue

3.2. Steps to add new scan tool into Cybarogue

4.Tools Usability

4.1. Device Firmware

4.1.1. Binwalk

4.1.2. Firmwalker

4.2. Android tools

4.2.1. Static Analyis

4.2.2. Dynamic Analysis

4.2.3. Mara Framework

4.3. Network Application

4.3.1. Nmap

4.3.2. NetCreds/Password Extractor

4.3.3. Pcap Analyser:

4.4. Web Application

4.4.1. SSLScan

4.4.2. Wapiti

4.4.3. Whatweb

4.4.4. Nikto

4.4.5. Dirbuster

4.5. Thick Client Application

4.5.1. Uncompyle6

5. Tools Features

5.1. Device Firmware

5.1.1. Binwalk

5.1.2. Firmwalker

5.2. Android tools

5.2.1. Static Analyis

5.2.2. Dynamic Analysis

5.2.3. Mara Framework

5.3. Network Scan

5.3.1. Nmap

5.3.2. NetCreds/Password Extractor

5.3.3. Pcap Analyser:

5.4. Web Application

5.4.1. SSLScan

5.4.2. Wapiti

5.4.3. Whatweb

5.4.4. Nikto

5.4.5. Dirbuster

5.5. Thick Client Application

5.5.1. Uncompyle6

6. Tools Setup process

6.1. Device Firmware

6.1.1. Binwalk

6.1.2. Firmwalker

6.2. Android tools

6.2.1. Static Analyis

6.2.2. Dynamic Analysis

6.2.3. Mara Framework

6.3. Network Scan Tools

6.3.1. Nmap

6.3.2. NetCreds/Password Extractor

6.3.3. Pcap Analyser:

6.4. Web Application

6.4.1. SSLScan

6.4.2. Wapiti

6.4.3. Whatweb

6.4.4. Nikto

6.4.5. Dirbuster

6.5. Thick Client Application

6.5.1. Uncompyle6

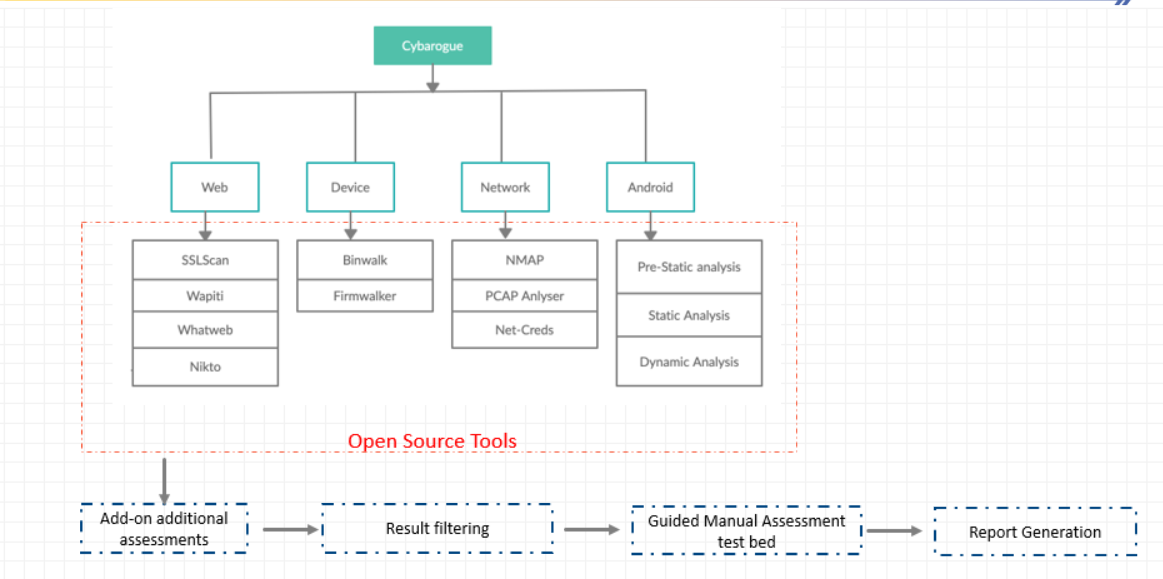
***1. Introduction***

**Cybarogue :**

Cybarogue is a framework designed for security Assessments with at least 30% automation for LTTS security assessment by development/testing teams, under a single framework.

Framework comprises of one click automated assessment based on integrated open tool, result processing, filtering and reporting, along with guided manual pentesting test bed.

It is totally based on Open Source Network Infrastructure Tools (OSNIT) Framework and Cybarogue Flowdiagram is as follows:



In Cybarogue the key features are as follows:

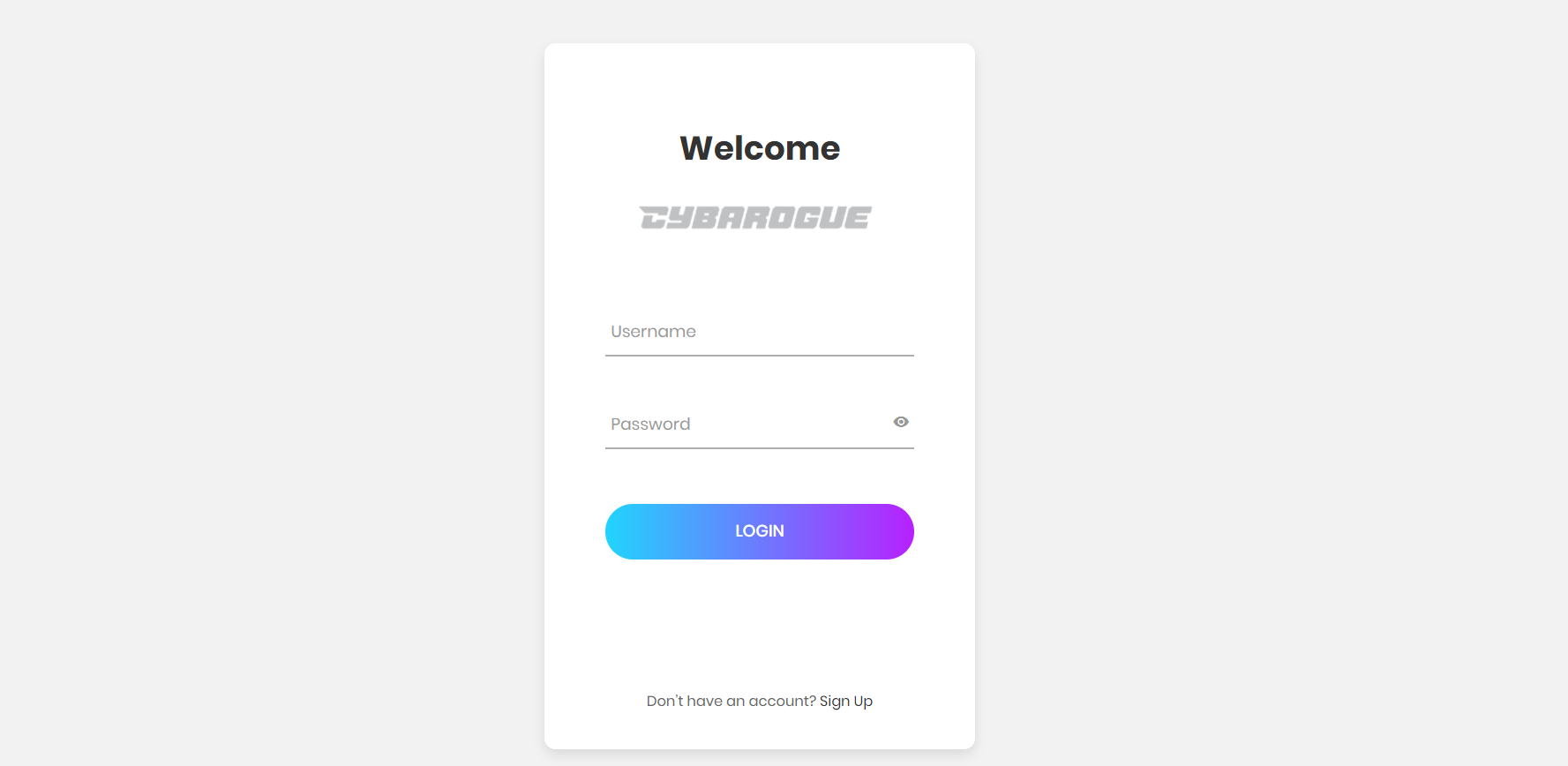
* Automated Static assessment of device firmaware
* Automated dynamic and static assessment of mobile applications
* Automated dynamic assessment of web applications.
* Automated assessment of device traffic dump for radio/Ethernet interfaces[File analysis]
* Automated assessment of network IP address.
* Report generation, false positive analysis.

**Basic Steps to connect with Cybarogue Framework:**

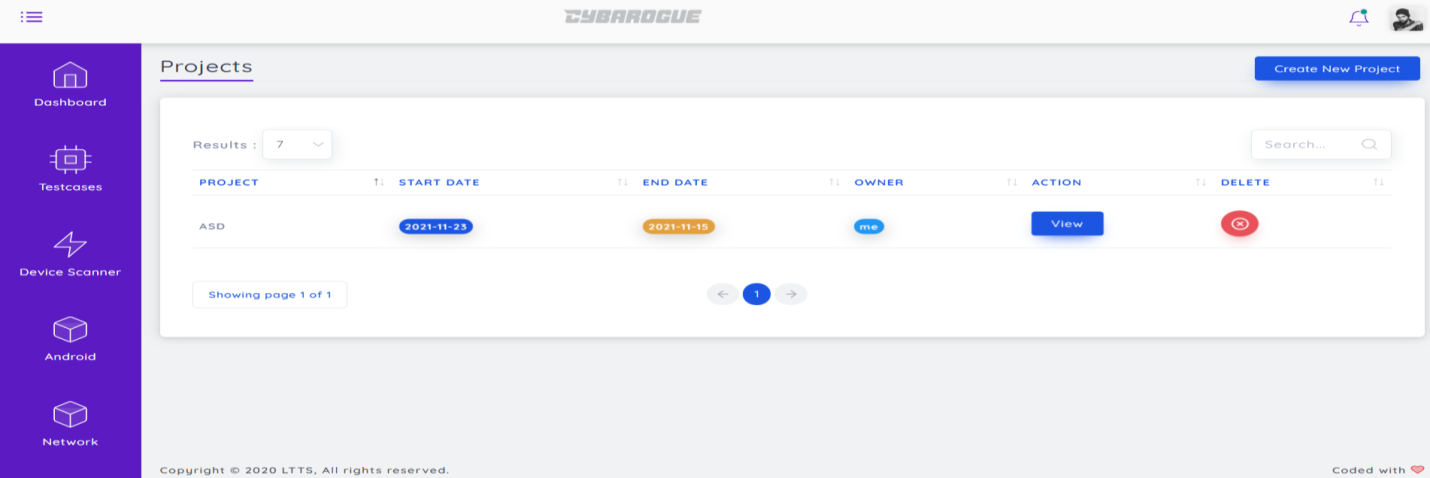
First the user needs the connection active that means the only authorized user can only able to connect to our cybarogue.

Cybarogue Framework overview is as follows

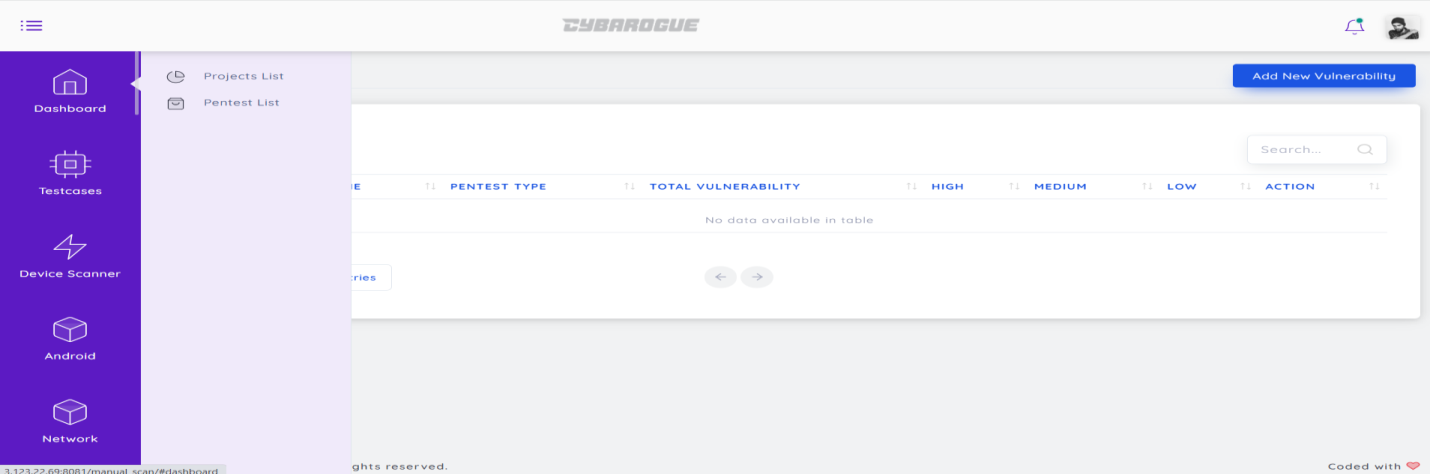
Step 1: By using Login credentials user must login to cybarogue as follows:



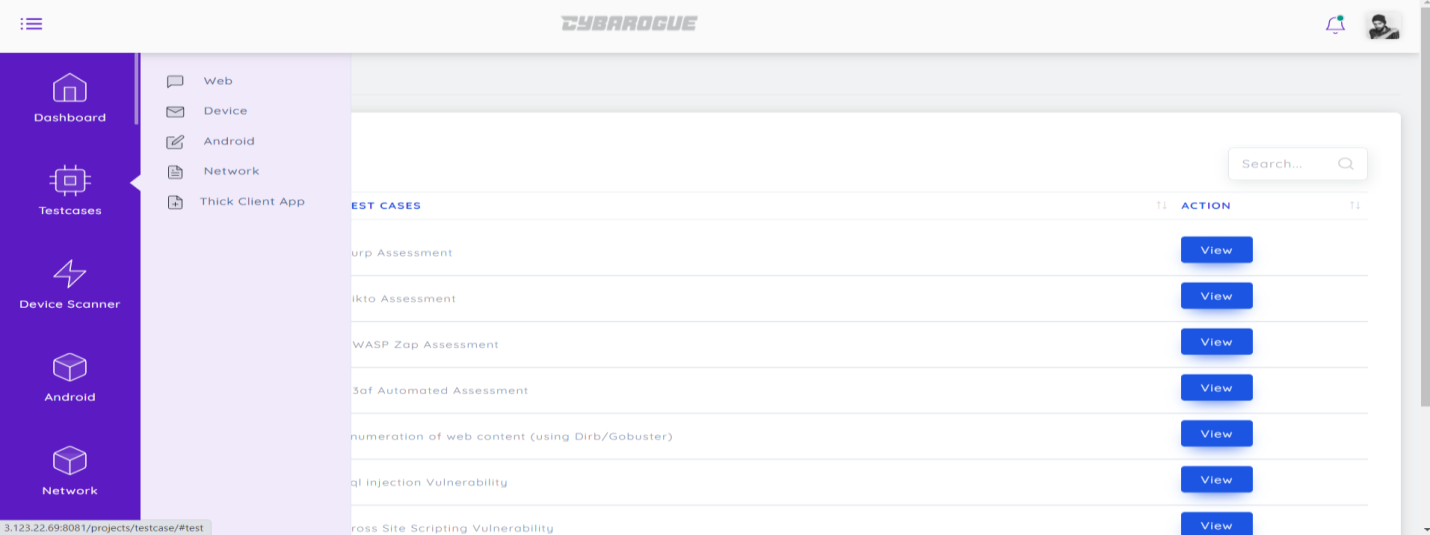
Step 2: After login, the framework looks initially as follows



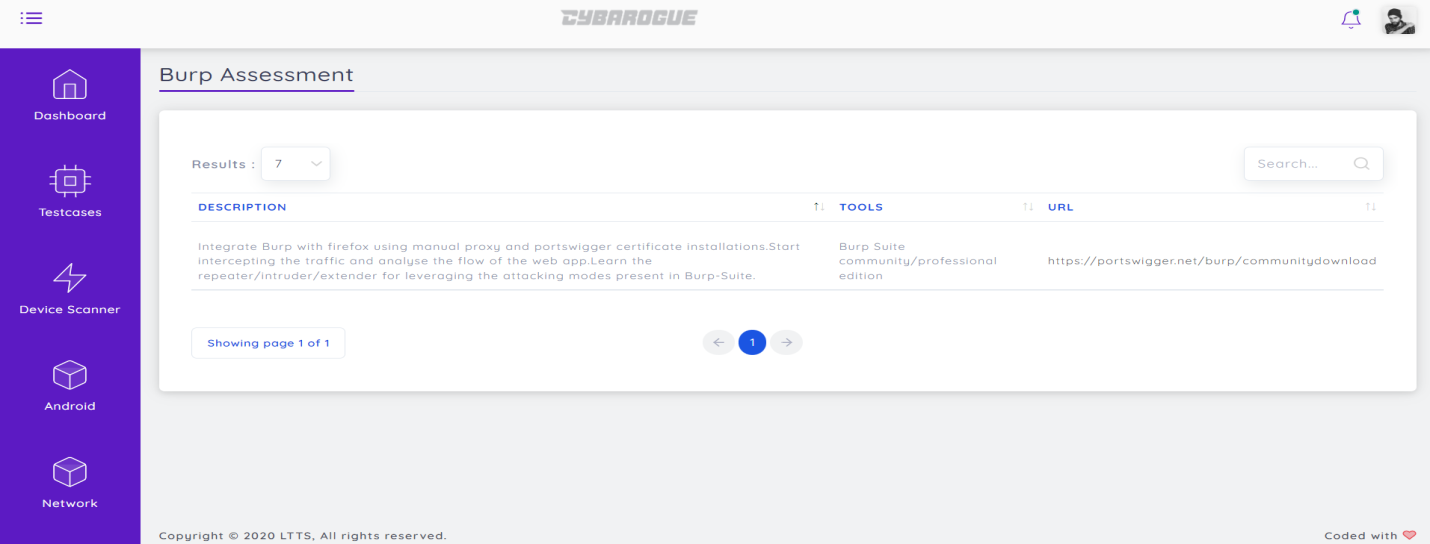
Step 3: In Dashboard there are 2 types of lists project list and pentest list, by selecting what we want it displays the list of details and also right side having an option to add new vulnerability.



Step 4: Here in the below window there is the portal Testcases in that again we can see options Device, Android, Network, Web and Thick Client App. These are the different tool categories in cybarogue.



Step 5: After selecting web in Testcases option, we can get list of testcases while clicking view option we can see as follows:



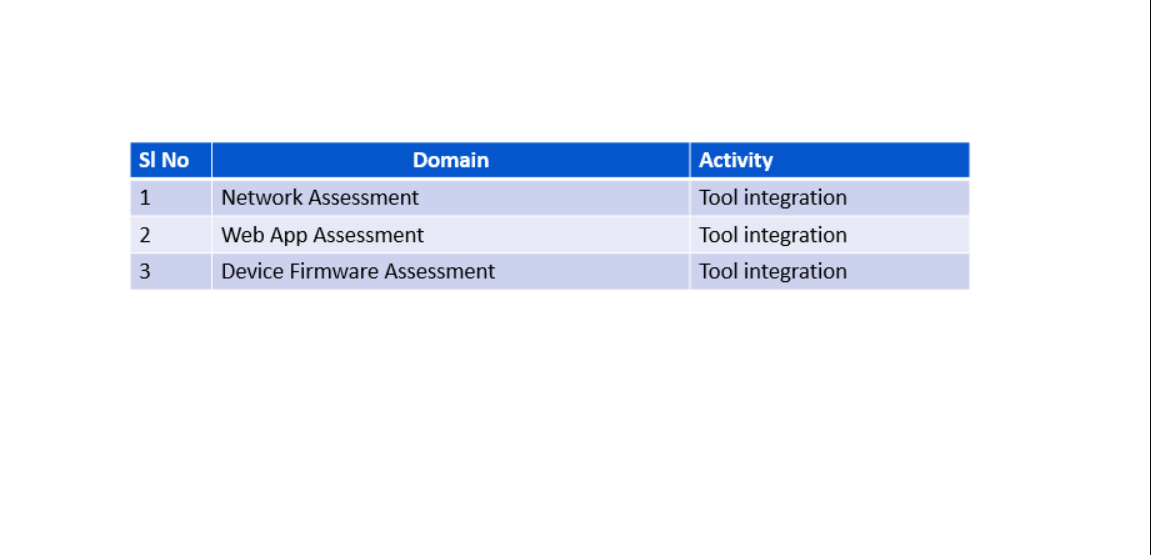
Like What are the actions we want to perform we can do that simply by selecting the respective category and uploading file and then get reports.

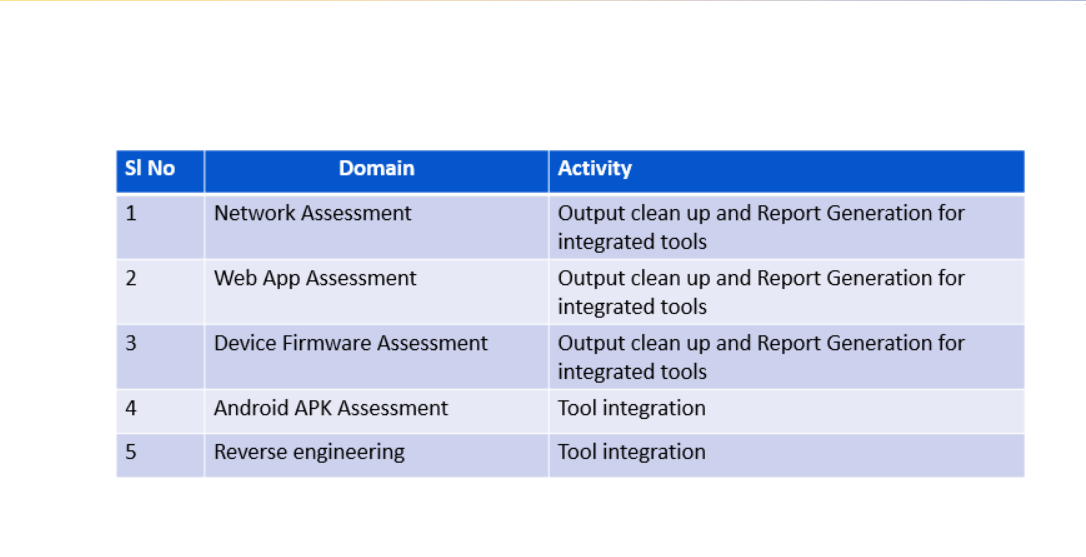
***2.Target Audience of the Document***

Easy to follow. No need to go for any other tools whatever the tools we need for security assessments are comes under this single framework. Our work is just to upload the file/url then it works on result processing next it filters the result and finally generates report and basically Security assessments on Network applications, Web applications, Device firmware, Mobile Applications.

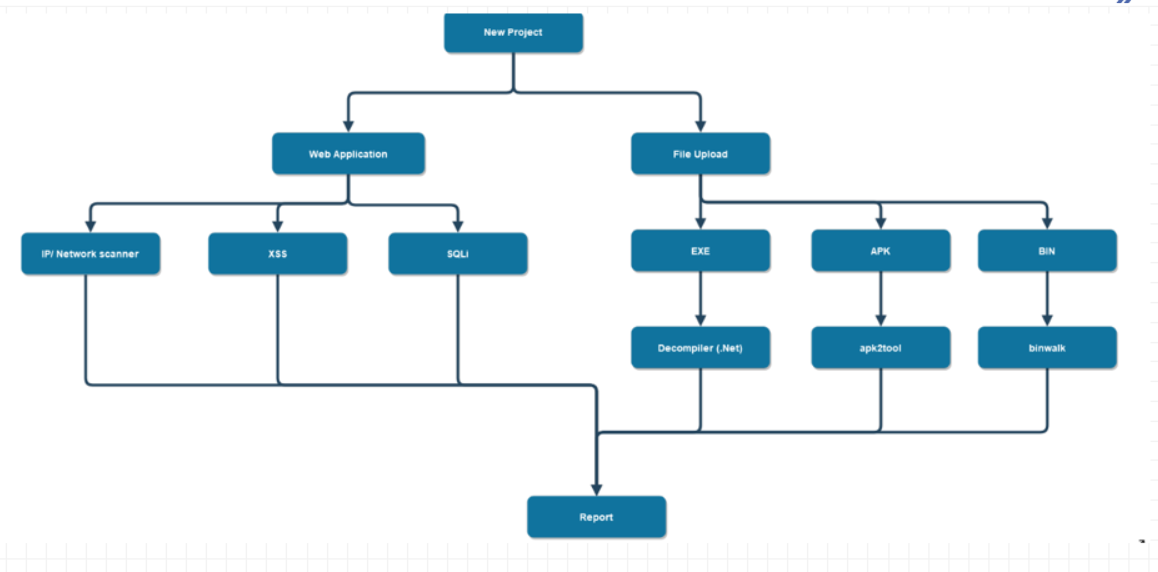
In Cybarogue

Phase-1 tool Assessment Completed

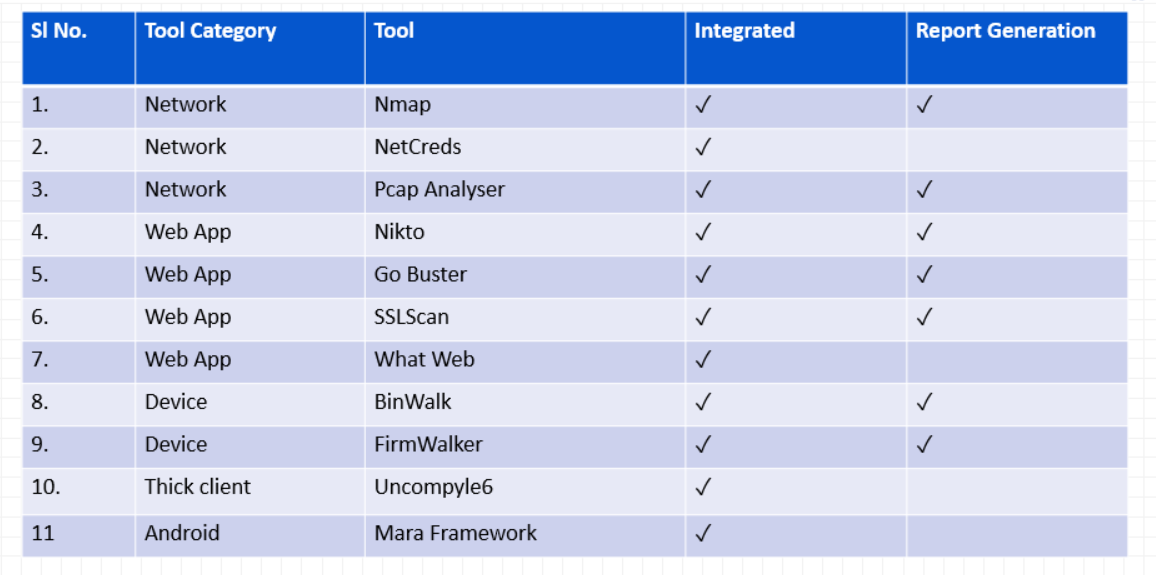


Phase-2 Current Tool Assessment

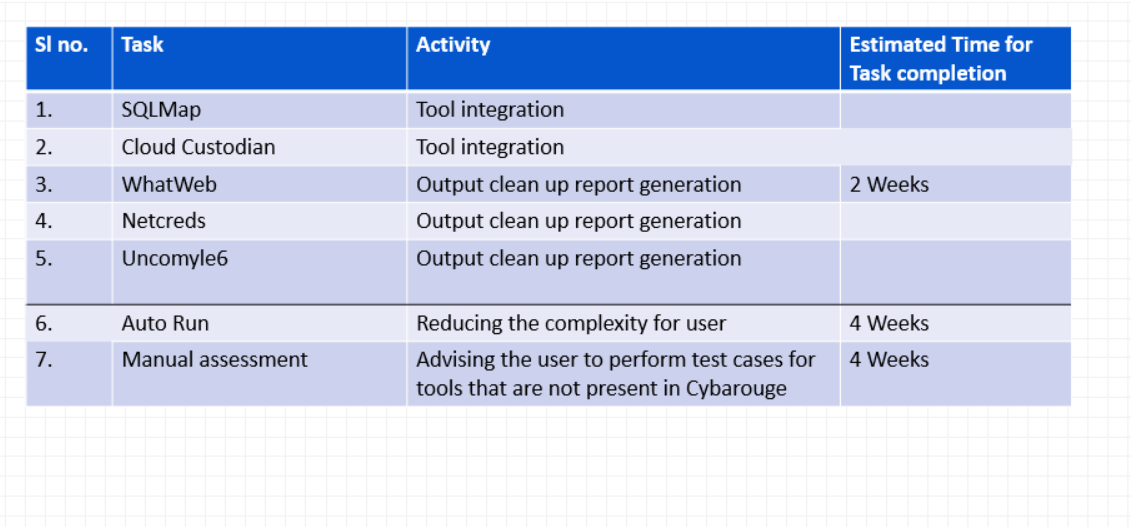
Product Flow Diagram of Cybarogue:



Updated Tabular form of Cybarogue:



The below table is about the Milestones Planned-Phase 3

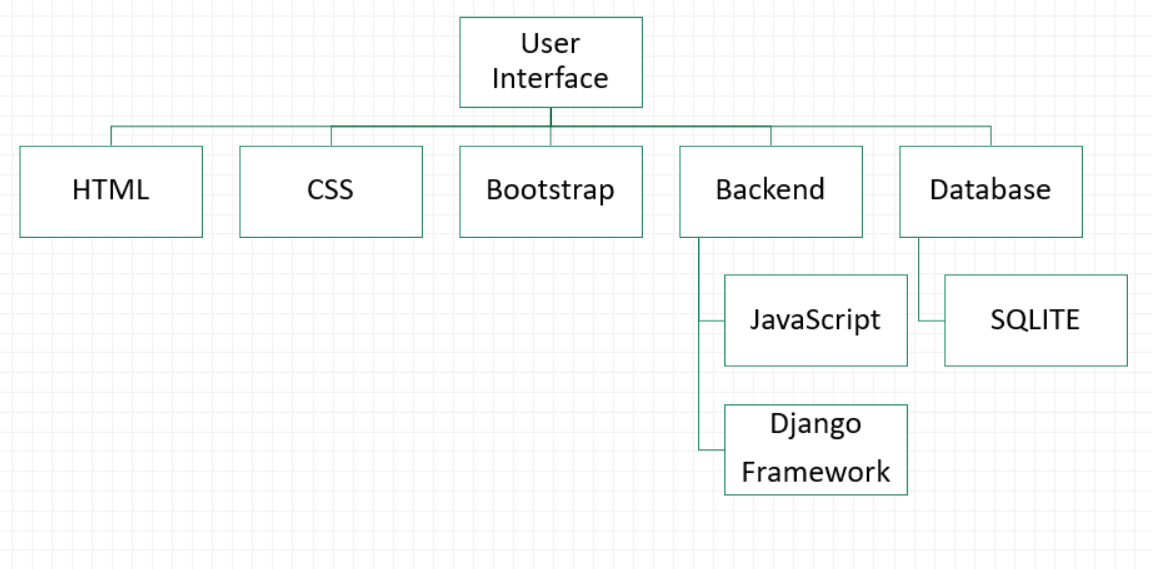


The main Target of the document is to get detail information and clear understandability about Cybarogue.

***3.Purpose of the Document***

The purpose of the document is to get the detail information about cybarogue means what are the tools we used for our framework development and how we are using those tools and how to setup each tool in a very detailed manner.

Architecture Block Diagram of Cybarogue:



**Steps to add new scan tool into cybarogue:**

In cybarogue there is a possibility to add new tool setup. For that we need to follow some steps as follows:

In templates folder

* Create UI HTML scan input page[AJAX used to link UI to function call]
* Create UI HTML scan list and output page.
* Add links to add output of tool to report and generate report.

In urls.py file

Update new tool name and directory

[eg: url(r’^nmap\_scan/’,

views.nmap1\_scan,

name=’nmap1\_scan’),

]

In models.py file

Create required fields to store data

[eg: scan\_id = models.TextField(blank=True, null=True) ]

In views.py file

Import models

Define function for scan [request from UI as argument]

Collect data from request required for tool function [eg: IP, SCAN\_ID etc]

Format data from request

Create subprocess with command line tool and arguments collected from request

[eg:[ ‘bash’ ,’ /root/Desktop/Cybarogue/nmapAutomator/nmapAutomator.sh’ ,scan\_url,scan\_type]]

Store output from tool into model imported.

Redirect to home/ tools page.

In report gen folder

Import models

Clean up output and add to models.

On add to report, add heading of tool, add cleaned output append to text file.

On generate report, Create docx file and add text from created file.

#Django migration and make migration commands to update newly added models with framework.

By following above steps which are nothing but add changes into coding part of cybarogue then setup the tool as 🡪 sudo ./setup.sh and to run use 🡪sudo ./run.sh and then to add the new tool setup is successful and can be able to saw the new tool which was added is appear on cybarogue framework.

***4. Tools Usability***

In Cybarogue we are using open source network Interface tools(OSNIT) the following tools

4.1. Device Firmware

4.2. Android Application tools

4.3. Network Application tools

4.4. Web Application tools

4.5. Thick Client Application

Again in this category what are the tools we are using are the following:

**4.1. Device Firmware Tools :**

Firmware is a software program or set of instructions programmed on a hardware device. It provides the necessary instructions for how the device communicates with the other computer hardware. ... Firmware is typically stored in the flash ROM of a hardware device.

In Cybarogue under Device Firmware tools category we consider binwalk and firmwalker

4.1.1. Binwalk:

Binwalk is a tool for searching a given binary image for embedded files and executable code. Specifically, it is designed for identifying files and code embedded inside of firmware images. Binwalk uses the libmagic library, so it is compatible with magic signatures created for the Unix file utility.

4.1.2. Firmwalker:

Firmwalker is a tool written in a bash script and it is used for analyzing the file system of extracted or mounted firmware.

**4.2. Android Application Tools :**

Tools apps for Android devices include a variety of useful features. With these apps, you can monitor your phone's data usage, manage files and apps, use your device as a broadband modem, back up your device's data, scan new apps for malware, and quickly uninstall apps.

In Cybarogue under Android application tools category we consider Pre-static Analysis, Static Analysis, Dynamic Analysis and MaraFramework.

4.2.1. Static Analyis:

Static analysis (or static code analysis) is an analysis run on the source code, against some set rules, even before the program runs (usually even before the unit tests). This is a kind of debugging that is done without even running the program and this is usually the first step taken towards code-analysis.

4.2.2. Dynamic Analysis:

Dynamic analysis of android apps is all about analyzing apps in real time, for the purpose of detecting application level vulnerabilities and for the sake of manipulating applications while they execute.

4.2.3. MaraFramework:

MARA is a Mobile Application Reverse engineering and Analysis Framework. It is a tool that puts together commonly used mobile application reverse engineering tools, in order to make task or reverse engineering and analysis easier and friendly to mobile application developers and security professionals.

**4.3. Network Application tools:**

Network Application is any application running on one host providing communication to another application running on a different host. It allow network operators to easily manage and monitor network traffic as well as analyze data that can be improve network systems. One of the most important areas in network management is security. The main objectives of network security tools are to maintain network integrity and prevent malicious attacks on servers and systems within the network. These tools usually do this by checking the vulnerability of the network connections and servers.

In Cybarogue under network application tools category we consider some effective tools they are Nmap, NetCreds and Pcap Analyser to design our framework.

4.3.1. Nmap:

Nmap stands for Network Mapper is a free Open source command-line tool. Nmap is an information-gathering tool used for recon reconnaissance. Basically, it scans hosts and services on a computer network means it sends packets and analyzes the response.

4.3.2. NetCreds/Password Extractor:

NetCreds is a free tool that sniffs passwords and hashes from a network interface. The tool launches a MITM attack to capture network packets, and ultimately the login credentials. It is a python based tool that requires scapy and wsgiref library to work. NetCreds runs on Linux as well as Windows OS. Linux(Kali) has built-in Python package, It is capable of sniffing the credentials from an interface as well as pcap file. In order to automatically detect the available interface.

4.3.3. Pcap Analyser:

Packet capture (PCAP) analysis is the process of obtaining and analyzing individual data packets that travel through your network. Because packet analysis (also known as packet capture or packet sniffing) is crucial to network management,

network admins should understand the key concepts of packet capture analysis.

**4.4. Web Application Tools:**

Formally defined as web application performance tools (WAPT), these tools assist in load, stress, and performance testing of sites, web applications, APIs, servers, and other various web interfaces, as well as to find bottlenecks and performance issues.

In Cybarogue under Web Application tools category we consider are SSLScan, Wapiti, Whatweb and Nikto to design our framework.

4.4.1. SSLScan:

SSL stands for Secure Sockets Layer, a security protocol that creates an encrypted link between a web server and a web browser. Companies and organizations need to add SSL certificates to their websites to secure online transactions and keep customer information private and secure. SSLScan is a command-line tool that performs a wide variety of tests overthe specified target and returns a comprehensive list of the protocols and ciphers accepted by an SSL/TLS server along with some other information useful in a security test.

4.4.2. Wapiti:

Wapiti is an open source tool that scans web applications for multiple vulnerabilities including data base injections, file disclosures, cross site scripting, command execution attacks, XXE injection, and CRLF injection. The database injection includes SQL, XPath, PHP, ASP, and JSP injections.

4.4.3. Whatweb:

Whatweb is a scanner written in the Ruby language. This tool can identify and recognize all the web technologies available on the target website. Whatweb works as an information-gathering tool and can identify all the email addresses, SQL errors, technology used in the website.

4.4.4. Nikto:

Nikto is a free software command-line vulnerability scanner that scans webservers for dangerous files/CGIs, outdated server software and other problems.

4.4.5. Dirbuster:

DirBuster is a multi threaded java application designed to brute force directories and files names on web/application servers. Often is the case now of what looks like a web server in a state of default installation is actually not, and has pages and applications hidden within.

**4.5. Thick Client Application:**

Thick Client applications refer to the applications that run on a user's machine. In these applications, the client handles most of the business logic, which includes the validations, view components and occasionally, temporary data. The following are a few examples: Google Talk. Yahoo Messenger.

In Cybarogue under Thick Client Application Tool category we consider Uncompyle6

4.5.1. Uncompyle6:

A native Python cross-version decompiler and fragment decompiler. The successor to decompyle, uncompyle, and uncompyle2.

uncompyle6translates Python bytecode back into equivalent Python source code. It accepts bytecodes from Python version 1.0 to version 3.8, spanning over 24 years of Python releases. We include Dropbox’s Python 2.5 bytecode and some PyPy bytecodes.

The above all tools are Open source tools. Cybarogue follows OSNIT Framework

***5. Tools Features:***

In this category we know briefly about each tools features as follows

**5.1. Device Firmware Tools Features:**

Firmware is a type of software that is etched directly into a piece of hardware.

It operates without going through APIs, the operating system, or device drivers providing the needed instructions and guidance for the device to communicate with other devices or perform a set of basic tasks and functions as intended.

In Device Firmware tools we use binwalk and firmwalker for our framework.

5.1.1. Binwalk Tool Features:

* Scanning Firmware – Binwalk can scan a firmware image for many different embedded file types and file systems
* File Extraction – You can tell binwalk to extract any files that it finds in the firmware image
* Entropy Analysis – Can help identify interesting sections of data inside a firmware image
* String Search – Allows you to search the specified file(s) for a custom string

5.1.2. Firmwalker Tool Features:

* etc/shadow and etc/passwd
* list out the etc/ssl directory
* search for SSL related files such as .pem, .crt, etc.
* search for configuration files
* look for script files
* search for other .bin files
* look for keywords such as admin, password, remote, etc.
* search for common web servers used on IoT devices
* search for common binaries such as ssh, tftp, dropbear, etc.
* search for URLs, email addresses and IP addresses

**5.2. Android Tool Features**:

* User-friendly and simple interface.
* Simple Login or registration page.
* Fast loading speed.
* Social media tools integration.
* User feedback space.
* Multiple language support.

In Android Application tools we use static, Dynamic and Maraframework.

5.2.1. Static Analyis Tool Features:

* To calculate metrics such as cyclomatic complexity or nesting levels (which can help to identify where more testing may be needed due to increased risk).
* To enforce coding standards.
* To analyze structures and dependencies.
* Help in code understanding.

5.2.2. Dynamic Analysis Tool Features:

* It helps detect defects in the software.
* Determines the quality attributes of the code.
* Assists the team in identifying bugs and bottlenecks of the software.
* Testing is done by executing the program.
* Helps the team in validating the consistency of the software product.

5.2.3. MaraFramework Tool Features:

* Reverse engineer apk files to smali, java jar files, java source code and dalvik bytecode (jadx format)
* Reverse engineer dex, jar and class files into java source code and dalvik bytecode (jadx format)
* Statically Analyze java source code and dalvik bytecode
* Scan for apk vulnerabilities via [androbugs](https://github.com/AndroBugs/AndroBugs_Framework" \t "_blank)
* Scan ssl domains found in the app via the standalone SSL scanner that makes use of [pyssltest](https://github.com/moheshmohan/pyssltest" \t "_blank)and [testssl](https://github.com/drwetter/testssl.sh" \t "_blank)

**5.3. Network Tools Features:**

A network monitoring system detects and reports failuresof devices or connections. It measures the CPU utilization of hosts, the network bandwidth utilization of links, and other aspects of the operation

Key Features of Network Application tools are:

* Real-time performance metrics.
* Automatic device discovery.
* Intelligent alerts.
* Issue diagnosis and root cause analysis.
* Network maps.
* Customizable dashboards.
* Natural scalability.

In in Network Application tools we use Nmap, NetCreds and Pcap Analyser to design our framework.

5.3.1. Nmap Tool Features:

* Host discovery – Identifying hosts on a network. For example, listing the hosts that respond to [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) and/or [ICMP](https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol) requests or have a particular port open.
* [Port scanning](https://en.wikipedia.org/wiki/Port_scanner) – Enumerating the open [ports](https://en.wikipedia.org/wiki/TCP_and_UDP_port) on target hosts.
* Version detection – Interrogating network services on remote devices to determine application name and version number.
* [TCP/IP stack fingerprinting](https://en.wikipedia.org/wiki/TCP/IP_stack_fingerprinting) – Determining the [operating system](https://en.wikipedia.org/wiki/Operating_system) and hardware characteristics of network devices based on observations of network activity of said devices.
* Scriptable interaction with the target – using Nmap Scripting Engine (NSE) and [Lua](https://en.wikipedia.org/wiki/Lua_(programming_language)" \o "Lua (programming language)) programming language.

Nmap can provide further information on targets, including reverse [DNS](https://en.wikipedia.org/wiki/Domain_Name_System) names, device types, and [MAC addresses](https://en.wikipedia.org/wiki/MAC_address).

Typical uses of Nmap:

* Auditing the security of a device or [firewall](https://en.wikipedia.org/wiki/Firewall_(computing)) by identifying the network connections which can be made to, or through it.
* Identifying open ports on a target host in preparation for auditing.
* Network inventory, [network mapping](https://en.wikipedia.org/wiki/Network_mapping), maintenance and asset management.
* Auditing the security of a network by identifying new servers.
* Generating traffic to hosts on a network, response analysis and response time measurement.
* Finding and [exploiting](https://en.wikipedia.org/wiki/Exploit_(computer_security)) vulnerabilities in a network
* [DNS](https://en.wikipedia.org/wiki/Domain_Name_System) queries and [subdomain](https://en.wikipedia.org/wiki/Subdomain" \o "Subdomain) search

5.3.2. NetCreds/Password Extractor Tool Features:

It can sniff the following directly from a network interface or from a PCAP file:

* URLs visited
* POST loads sent
* HTTP form logins/passwords
* HTTP basic auth logins/passwords
* HTTP searches
* FTP logins/passwords
* IRC logins/passwords
* POP logins/passwords
* IMAP logins/passwords
* Telnet logins/passwords
* SMTP logins/passwords
* SNMP community string
* NTLMv1/v2 all supported protocols: HTTP, SMB, LDAP, etc.
* Kerberos

5.3.3 Pcap Analyser Tool Features:

* Live capture and offline analysis
* Rich VoIP analysis
* Read/write many different capture file formats
* Capture compressed files (gzip) and decompress them on the fly
* Deep inspection of hundreds of protocols
* Standard three-pane packet browser
* Captured network packets can be browsed via a GUI or TShark utility
* Multi-platform easily ran on Linux, Windows, OS X, and FreeBSD
* Powerful display filters
* Output can be exported to XML, CSV, PostScript, or as a plain text
* Packet list can use coloring rules for quick and intuitive analysis

**5.4. Web Tools Features:**

### HTML and the DOM

[HTML](https://en.wikipedia.org/wiki/HTML) and [DOM](https://en.wikipedia.org/wiki/Document_Object_Model) viewer and editor is commonly included in the built-in web development tools. The difference between the HTML and DOM viewer, and the view source feature in web browsers is that the HTML and DOM viewer allows you to see the DOM as it was rendered in addition to allowing you to make changes to the HTML and DOM and see the change reflected in the page after the change is made.

In addition to selecting and editing, the HTML elements panels will usually also display properties of the DOM object, such as display dimension, and Cascading- style sheet properties.

3D page inspector, a feature of Firefox versions 11 to 46.

Firefox versions 11 to 46 were equipped with a [3D page inspector](https://en.wikipedia.org/wiki/Features_of_Firefox#3D_Page_Inspector) using [WebGL](https://en.wikipedia.org/wiki/WebGL" \o "WebGL), where the nesting of elements was visualized with layers protruding from the page surface.

### Web page assets, resources and network information

Web pages typically load and require additional content in the form of images, scripts, font and other external files. Web development tools also allow developers to inspect resources that are loaded and available on the web page in a tree-structure listing, and the appearance of style sheets can be tested in real time.

Web development tools also allow developers to view information about the network usage, such as viewing what the loading time and bandwidth usage are and which [HTTP headers](https://en.wikipedia.org/wiki/HTTP_headers) are being sent and received.

### Profiling and auditing

[Profiling](https://en.wikipedia.org/wiki/Profiling_(computer_programming)) allows developers to capture information about the performance of a web page or web application. With this information developers can improve the performance of their scripts. Auditing features may provide developers suggestions, after analyzing a page, for optimizations to decrease page load time and increase responsiveness. Web development tools typically also provide a record of the time it takes to render the page, memory usage, and the types of events which are taking place.

These features allow developers to optimize their web page or web application.

#### JavaScript debugging

[JavaScript](https://en.wikipedia.org/wiki/JavaScript) is commonly used in web browsers. Web development tools commonly include a panel to debug scripts by allowing developers to add watch expressions, breakpoints, view the call stack, and pause, step over, step into, and step out of functions while debugging JavaScript.

A JavaScript console is commonly included. The consoles allow developers to type in JavaScript commands and call functions, or view errors that may have been encountered during the execution of a script.

In Web Application tools we use SSLScan, Wapiti, Whatweb and Nikto to design our framework.

5.4.1. SSL Scan Tool Features:

* SSLScan provides Query SSL services
* It will Support cryptographic protocols like SSLv2, SSLv3, TLS 1.0, TLS 1.1 and TLS 1.2 (It depends on the used OpenSSL library)
* Perform an HTTP connect
* STARTTLS support with FTP, IMAP, POP3, SMTP, and XMPP
* Bind to local IP address
* Python extensions
* SSLScan also supports IPv4 and IPv6
* Text and XML output

5.4.2. Wapiti Tool Features:

* Wapiti supports both GET and POST HTTP methods for attacks.
* It also supports multipart forms and can inject payloads in filenames(upload).
* Warnings are raised when an anomaly is found (for example 500 errors and timeouts)
* Wapiti is able to make the difference between permanent and reflected XSS vulnerabilities.

General features :

* Generates vulnerability reports in various formats (HTML, XML, JSON, TXT, CSV)
* Can suspend and resume a scan or an attack (session mechanism using sqlite3 databases)
* Can give you colors in the terminal to highlight vulnerabilities
* Different levels of verbosity
* Fast and easy way to activate/deactivate attack modules
* Adding a payload can be as easy as adding a line to a text file
* Configurable number of concurrent tasks to perform HTTP requests

Browsing features

* Support HTTP, HTTPS and SOCKS5 proxies
* Authentication via several methods : Basic, Digest, NTLM or GET/POST on login forms
* Ability to restrain the scope of the scan (domain, folder, page, url)
* Automatic removal of one are more parameters in URLs
* Multiple safeguards against scan endless-loops (for example, limit of values for a parameter)
* Possibility to set the first URLs to explore (even if not in scope)
* Can exclude some URLs of the scan and attacks (eg: logout URL)
* Import cookies from your Chrome or Firefox browser or using the wapiti-getcookie tool
* Can activate / deactivate SSL certificates verification
* Extract URLs from Flash SWF files
* Try to extract URLs from javascript (very basic JS interpreter)
* HTML5 aware (understand recent HTML tags)
* Several options to control the crawler behavior and limits.
* Skipping some parameter names during attack.
* Setting a maximum time for the scan process.
* Adding some custom HTTP headers or setting a custom User-Agent.

5.4.3. Whatweb Tool Features:

* Over 1700 plugins
* Control the trade off between speed/stealth and reliability
* Plugins include example URLs
* Performance tuning. Control how many websites to scan concurrently.
* Multiple log formats: Brief (greppable), Verbose (human readable), XML, JSON, MagicTree, RubyObject, MongoDB, SQL, and ElasticSearch.
* Proxy support including TOR.
* Custom HTTP headers.
* Basic HTTP authentication
* Control over webpage redirection
* Nmap-style IP ranges
* Fuzzy matching
* Result certainty awareness
* Custom plugins defined on the command line

5.4.4. Nikto Tool Features:

* Full support for SSL
* Finds sub-domain
* Supports full HTTP Proxy
* Outdated component report
* Result saved in multiple format (xml, csv etc)
* Username guessing
* Gives details of installed software
* Takes Nmap file as input to scan port in a web-server.
* Able to perform dictionary attack.
* Updated easily

5.4.5. Dirbuster Tool Features:

The DirBuster tool is a graphical user interface (GUI) application that includes the following features:

* It works over http and https.
* It scans for both files and directories.
* It scans recursively into the directories it identifies, providing more in-depth searches.
* It can be started on any directory.

**5.5. Thick Client Application Tool Features:**

* High-Level Graphics User Interface.
* Efficient Program and Data Processing.
* Ability to Work Offline.
* Highly Efficient Server Performance.

In Thick client Application Tool we use Uncompyle6

5.5.1. Uncompyle6 Tool Features:

Uncompyle6 is more than your normal hacky decompiler. Using [compiler](https://pypi.python.org/pypi/spark_parser) technology, the program creates a parse tree of the program from the instructions; nodes at the upper levels that look a little like what might come from a Python AST. So we can really classify and understand what’s going on in sections of Python bytecode.

Building on this, another thing that makes this different from other CPython bytecode decompilers is the ability to deparse just fragmentsof source code and give source-code information around a given bytecode offset.

***6.Tools Setup Process(Developer):***

In Cybarogue the only setup requires is🡪 **“**sudo ./setup.sh**”**

To Know about each tool in the cybarogue lets get into the following we are using open source network Interface tools(OSNIT) and in this we can briefly get an idea about the setup process of each tool and First we make sure with enivironmental changes.. Here in Cybarogue we set the path as “home/ubuntu/cybarogue-master” in views.py files and if you want to know about reports code the location is “cybarogue-master>ReportGen>views.py” After every scan to reportgen>views.py we can all the reports requests here and automatically generates reports Cybarogue’s every scan reports/results related process is available in ReportGen file.

**6.1. Device Firmware Tool Setup Process:**

Firmware is a software that is embedded into a hardware device. New firmware often fixes bugs, contains new features, and protects you from security vulnerabilities. Some products can regularly check for new firmware and download it, or you can check for new firmware and download it manually.

In Cybarogue under Device Firmware tools category we consider binwalk and firmwalker.

The location of the according tools code is as follows:

HTML Code🡪 cybarogue-master>Devicescanners>templates>binwalk and firmwalker

Python Code🡪cybarogue-master>Devicescanners>views.py>binwalk and firmwalker.

6.1.1. Binwalk Tool Setup Process:

**Install Binwalk**

Binwalk was developed as an open-source program and it is something between software and hardware. To analyze binary files, you can use this simple Linux tool. Binwalk extracts all the files it finds within the firmware image automatically. It is not difficult to install Binwalk. It is pre-installed on the [Kali Linux](https://blog.eldernode.com/tag/kali-linux/) Operating system. So, let’s see how is the process of installation Binwalk on Ubuntu.

**Before Installation Binwalk**

While Binwalk supports Python 2.7 – 3.x., it does run faster in [Python 3](https://blog.eldernode.com/install-python-3-ubuntu-20/). However, the most of systems have Python 2.7 set as their default Python interpreter. So, you will find the installation procedures for both here:

Python 2.7🡪 sudo python setup.py install

Python 3.x🡪 sudo python3 setup.py install

**Binwalk Quick Installation Process:**

First, you need to run the command below to download Binwalk.

Step 1.

Download the binwalk zip file from github

Step 2.

Unzip filename.zip

Then, to install Binwalk type the following commands:

sudo apt update

sudo apt install binwalk

If you have a previously installed version of Binwalk, you are recommended to uninstall it before upgrading. So, run:

cd binwwalk-master

sudo python setup.py uninstall

sudo python setup.py install

In cybarogue-master

6.1.2. Firmwalker Tool Setup Process:

**Installation**

You can install a firmwalker tool by using the one-liner command from github.

**Output of firmwalker**

firmwalker provides you quick initial results to initiate security assessment of firmware.

* may give sensitive details by searching keywords such as admin, remote, password etc.
* may give .bin, script, configuration files
* may give IP addresses, email id, URLs etc.
* may give details for etc/shadow, etc/passwd and etc/ssl directories
* may give SSL related files such as .pem, .crt etc.

**To Run firmwalker**

This tool can be run against the filesystem of extracted or mounted firmware. For testing purposes, clone the repository of IoTGoat, an intentionally vulnerable firmware developed by the OWASP community.

**6.2. Android Tool Setup Process:**

The Android build system compiles app resources and source code, and packages them into APKs or Android App Bundles that you can test, deploy, sign, and distribute. ... The output of the build is the same whether you are building a project from the command line, on a remote machine, or using Android Studio

Basic Methodology is as follows:

1. Strategy. The first phase of the mobile app development process is defining the strategy for evolving your idea into a successful app.
2. Analysis and Planning.
3. UI / UX Design.
4. App Development.
5. Testing.
6. Deployment & Support.

In Cybarogue under Android application tools category we consider Pre-static Analysis, Static Analysis, Dynamic Analysis and MaraFramework. Let’s see the Setup Process of each tool.

The Android Tool related code location in Cybarogue is as follows:

HTML Code🡪cybarogue-master>Android>templates>Androidscan

Python Code🡪cybarogue-master>Android>views.py>Androidscan

6.2.1. Static Analysis Tool Setup Process:

Step 1: Finalize the tool.

Step 2: Create a scanning infrastructure and deploy the tool. Step 3: Customize the tool.

Step 4: Prioritize and on-board.

Step 5: Analyze results.

Step 6: Governance and training.

Summing it up.

6.2.2. Dynamic Analysis Tool Setup Process:

The primary steps in performing a dynamic analysis are summarized as follows:

* Define the dynamic environment (loading).
* Formulate the proper finite element model.
* Select and apply the appropriate analysis approach to determine the behavior of the structure.
* Evaluate the results.

6.2.3. MaraFramework Tool Setup Process:

## Installing MARA on Linux/Nethunter

MARA ships with a script that assists in downloading and installing the dependencies for each of the tools and components it ships with. Simply run the setup.sh script with sudo privileges and it will install them. If you are using a Mac, simply run the setup\_mac.sh script instead.

**6.3. Network Tool Setup Process:**

All you have to do is follow these five steps.

1. Connect your router. The router is the gateway between the Internet and your home network.
2. Access the router's interface and lock it down.
3. Configure security and IP addressing. ...
4. Set up sharing and control.
5. Set up user accounts.

The location of according tools code is as follows:

HTML Code🡪 cybarogue-master>Networkscanners>templates>nmap,Passwordextractorscan and pcap

Python Code🡪cybarogue-master>Networkscanners>Views.py>nmap,passwordextractorscan and pcap

In Network Application tools we use Nmap, NetCreds and Pcap Analyser to design our framework.

6.3.1. Nmap Tool Setup process:

To get started, download and install Nmap from the nmap.org website and then launch a command prompt.

To install nmap

1. Step 1: Update Ubuntu Package List.

Make sure the software packages on your Ubuntu system are up-to-date with the command**:** sudo apt-get update.

1. Step 2: Install Nmap.

To install NMAP on Ubuntu, run the command: sudo apt-get install nmap.

1. Step 3: Verify Nmap Version Installed.

To check whether nmap is install or not. Type on command prompt 🡪 nmap –version

Typing nmap [hostname] or nmap [ip\_address] will initiate a default scan.

* A default scan uses 1000 common TCP ports and has Host Discovery enabled.
* Host Discovery performs a check to see if the host is online.

6.3.2. NetCreds/Password Extractor Tool Setup Process:

Net creds is python based tool that requires scapy and wsgiref library to work. Net creds runs on Linux as well as Windows OS. Linux(kali) has built-in Python package, however, for Windows installation, a python compiler is recommended to run the script. The tool can be downloaded from github repository.

Sniffing via Net Creds

Net Creds is capable of sniffing the credentials from an interface as well as pcap file. In order to automatically detect the available interface, run the following command in the OS terminal.

sudo python net-creds.py

In our case, the tool detected ‘eth0’ as the available network interface. The following command can be used to start sniffing on the available interface i.e ‘eth0’.

sudo python net-creds.py –i eth0

To test the working of the tool, let’s open an example web application running on the HTTp protocol. We have also logged into web application to see if the login credentials are captured by Net Creds. Once the web application is launched in the browser, the tool starts capturing the packets and login credentials.

The packets from a specific Ip address can be ignored using the following command.

sudo python net-creds.py-f

Similarly, the following command can be used to read the

packets from a pcap file.

python net-creds.py -p pcapfile

6.3.3. Pcap Analyser Tool Setup Process:

Because the pcap data source creates an Apache Storm topology that can rapidly ingest raw data directly into HDFS from Apache Kafka, you can store all of your cybersecurity data in its raw form in HDFS and review or query it at a later date.

CCP supports two pcap components:

* The pycapa tool, for low-volume packet capture
* The Fastcapa tool, or high-volume packet capture

Fastcapa is a probe that performs fast network packet capture by leveraging Linux kernel-bypass and user space networking technology. The probe will bind to a network interface, capture network packets, and send the raw packet data to Kafka. This provides a scalable mechanism for ingesting high-volumes of network packet data into a Hadoop cluster.

Fastcapa leverages the Data Plane Development Kit (DPDK). DPDK is a set of libraries and drivers to perform fast packet processing in Linux user space.

* [Set up pycapa](https://docs.cloudera.com/ccp/2.0.1/setting-up-pcap/topics/ccp-setting_up_pycapa.html)  
  You can use the pycapa tool to capture low-volume data flow.
* [Start pcap](https://docs.cloudera.com/ccp/2.0.1/setting-up-pcap/topics/ccp-starting_pcap.html)  
  To start pcap, CCP provides a utility script. This script takes no arguments and is very simple to run.
* [Installing Fastcapa](https://docs.cloudera.com/ccp/2.0.1/setting-up-pcap/topics/ccp-installing_fastcapa.html)  
  You can install Fastcapa either automatically or manually. The automated installation is the simplest but it requires CentOS 7.1. If you are not running CentOS 7.1 or would like more visibility into the installation process, you can manually install Fastcapa.
* [Using Fastcapa](https://docs.cloudera.com/ccp/2.0.1/setting-up-pcap/topics/ccp-using_fastcapa.html)  
  You can use the Fastcapa tool to capture high-volume data flow.
* [Use Fastcapa in a Kerberized Environment](https://docs.cloudera.com/ccp/2.0.1/setting-up-pcap/topics/ccp-using_fastcapa_in_a_kerberized_environment.html)  
  You can use the Fastcapa probe in a Kerberized environment.

**6.4. Web Tool Setup Process:**

* Define the problem you are solving.
* Plan the workflow of your web application.
* Wireframe/prototype your web application.
* Receive Validation.
* Choose your firepower.
* Build your web application.
* Test your web application.
* Host and deploy your web application.

To know the location of web related code go to the location in cybarogue as follows:

HTML Code🡪 cybarogue-master>Tools>templates>sslscan and Nikto

cybarogue-master>Devicescanners>templates>Nikto

cybarogue-master>Webscanners>templates>Whatweb and Dirbuster

Python Code🡪 cybarogue-master>Tools>views.py>sslscan and Nikto

cybarogue-master>Devicescanners>views.py>Nikto

cybarogue-master>Webscanners>views.py>Whatweb and Dirbuster

In Cybarogue under Web Application tools we consider are SSLScan, Wapiti, Whatweb, Nikto and Dirbuster to design our framework. Lets Know about Tool Setup Process of each Tool.

6.4.1. SSL Scan Tool Setup Process:

If you are using Kali Linux operating system then, SSLScan is already be installed as it is a very helpful Security Analysis tool.

You can confirm by firing the terminal and hit the sslscan command in it. If the tool is installed in the system, then the sslscan command’s help page will appear.

If you are using other Linux-based distros like Ubuntu Linux, Linux Mint, Parrot, CentOS, etc then you can get it from the official SSLScan repository.

To get the SSLScan, just go to the Github repo and download it.

Now, to install sslscan you need some pre-requisites to be fulfilled. You need to have some tools or binaries installed on the system to proceed further with the installation of sslscan .

**Requirements:**

* gcc
* make
* cmake
* OpenSSL 1.0.0 or better
* openssl-devel
* python-devel (2.6, 2.7, 3.2 or 3.3 – Python 3.x preferred)

If you have all the requietes installed then, you can proced for the installtion of sslscan :

Makefile build of sslscan :

$ mkdir build

$ cd build

$ cmake

$ make

$ make install (as root user)

After successfull installation, we can start using it for our purposes.

6.4.2. Wapiti Tool Setup Process:

Wapiti installation

You do not have to "install" the software on your system to make it work.

If you have all the requirements on your system, just extract the tarball and launch the "wapiti" command line in

the "bin" folder :

./bin/wapiti (when you are in the extracted tarball)

or

python bin/wapiti (if your system can't execute the script directly)

Installation on Unix-like systems

If you really want to install Wapiti on your system, launch the setup.py script with the following command :

python setup.py install

It will copy the wapiti libraries (wapitiCore) in your Python installation and place the executables in a "bin" system

directory (eg: /usr/local/bin).

Using Wapiti on Windows systems

If you don't want to install all the requirements to use Wapiti on Windows you should look for a standalone package

made with py2exe (see the available downloads on SourceForge).

Then, just download and extract the zip archive and launch wapiti.exe from the Windows command line.

Installing Wapiti requirements on Windows

You can't install Wapiti on a Windows system but if (for some reasons) you really want to install all the

requirements requires

* Download a Python 2.7.5 (or more recent) installer for your platform.
* Install it and change the PATH environment variable to append the Python path
* Download a requests archive
* Extract the archive and call the setup.py script from the archive with "python setup.py install"
* Download BeautifulSoup 3
* Extract the archive and call the setup.py script from the archive with "python setup.py install"

The current version of Wapiti can be installed in 2 ways:

* download the source from the official site and run the installation script with Python3;
* using **pip3 install wapiti3** command.

After that, Wapiti will be ready to go.

# Working with the tool

To demonstrate how Wapiti works, we will use a specially prepared stand **sites.vulns.pentestit.ru** (an internal resource) containing various vulnerabilities (Injection, XSS, LFI / RFI) and other shortcomings of web applications.

# Setting up a virtual environment for Wapiti

Let's create a virtual environment called 'wapiti3'.  
In this example it will be created in the current working directory.

python –m venv wapiti3

Now let's activate it (make it our current working environment) :

. ./wapiti3/bin/activate

Or alternatively on Windows :

wapti3\Scripts\activate.bat

Now you are in the virtual environment you can install Wapiti and its dependencies :

python3 setup.py install

To leave the virtual environnement just call the following command :

deactivate.

Remember that you will need to reactivate the environment each time you want to use Wapiti.

To setup we use the command

python setup.py install –with-html5lib

6.4.3. Whatweb Tool Setup Process:

## Installation

**Step 1:** Open your kali Linux operating system and use the following command to install the tool from GitHub.

cd Desktop

**Step 2:** Now use the following command to move into the directory of the tool.

cd Whatweb

**Step 3:**Now you are in the directory of the tool. Use the following command to run the tool.

./whatweb

The tool is running successfully

6.4.4. Nikto Tool Setup Process:

Installing **Nikto** on Ubuntu system is basically straightforward as the package is already accessible on the default repositories.

First to update the system repositories we will use the following command :sudo apt-get update && sudo apt-get upgrade

After updating your system. Now Run the following command to Install Nikto.

sudo apt-get install nikto -y

After the completion of installation process ,use the following command to check if nikto is installed correctly or not :$nikto**-**Nikto <version>

If the command gives the version number of Nikto it means installation is successful.

**Usage:** Generally Nikto requires just an host to scan which can be specified with -h or -host option

6.4.5. Dirbuster Tool Setup Process:

As the project is not maintained, it takes more than apt install to get it. DirBuster comes by default with Kali Linux, however, if you are using other distribution, such as Debian, you have to install it manually. Make sure you have everything that the installation will need:

* GIT. You can install GIT by sudo apt install git
* Java. You can check your current Java version by executing java -version. If you don’t have it installed, on the terminal you will get the instructions on how it can be installed (spoiler alert: sudo apt install default-jre).

Now when you are ready, let’s proceed.

* Clone the repository of DirBuster: git clone https://gitlab.com/kalilinux/packages/dirbuster.git
* Move the dirbuster to opt directory: sudo mv dirbuster /opt. Opt directory is for installing unbundled packages - the packages that are provided from different sources than the packages that came with the OS distribution. It’s for external, third-party packages.
* Basically, you can stop after cloning the repository. Just run the ./DirBuster-1.0-RC1.sh and you are ready to go. However, it is not convenient to navigate to the DirBuster directory and run the script manually. That’s why we moved the folder to /opt directory - to make our life easier. If you want to finish the process of making your life easier, please proceed.
* Create a new file for aliases (if you don’t have it already): sudo nano ~/.bash\_aliases, and add a new alias to the file: alias dirbuster=’source /opt/dirbuster/DirBuster-1.0-RC1.sh’ After this, restart the terminal/ssh session or execute source ~/.bash\_aliases for the changes to take effect.

**6.5. Thick Client Tool Setup Process:**

A thick client (sometimes called a fat client) is a form of client-server architecture. Specifically, it is a networked computer system with most resources installed locally, rather than distributed over a network. ... Workplaces will commonly provide thick clients to employees so they can work offline.

Typical examples of thick clients are G-Talk, Yahoo Messenger, Microsoft Outlook,online trading portals,etc…

To know the location of Thickclient related code go to the location in cybarogue as follows:

HTML Code🡪 cybarogue-master>Thick>templates>uncompyle

Python Code🡪cybarogue-master>Thick>View.py>uncompyle

6.5.1. Uncompyle6 Tool Setup Process:

### Requirements

The code here can be run on Python versions 2.6 or later, PyPy 3-2.4 and later. Python versions 2.4-2.7 are supported in the python-2.4 branch. The bytecode files it can read have been tested on Python bytecodes from versions 1.4, 2.1-2.7, and 3.0-3.8 and later PyPy versions.

### Installation

This uses setup.py, so it follows the standard Python routine:

$pip install –e . #set up to run from source tree, or…

$python setup.py install #may need sudo

A GNU makefile is also provided so make

Install(possibly as root or sudo) will do the steps above.

### Running Tests

### $make check

A GNU makefile has been added to smooth over setting running the right command, and running tests from fastest to slowest.

If you have [remake](https://bashdb.sf.net/remake) installed, you can see the list of all tasks including tests via remake --tasks

### Usage

Run

$uncompyle6 \*compiled-python-file-pyc-or-pyo\*

For usage help:

$ uncompyle6 –h

**Note:** In Cybarogue for ant tool setup just need to use the command “sudo ./setup.sh” after source venv/bin/activate.