

HEIMDALL: A DIGITAL THREAT MANAGEMENT
PLATFORM

Research Manual

by

Killian O'Connor

For IT Carlow

1/11/2019

Abstract

Heimdall is a platform through which users can identify potential threats, information disclosure, and vulnerabilities on their externally facing network. The Heimdall platform will be extensible through the use of APIs, scalable through the usage of microservices and on-demand cloud hosting, and customisable due to user defined key indicators. The current project is being built as a proof of concept and as such will only contain the core web console and framework, as well as five key modules:

1. Darknet monitoring
2. Open source intelligence and information disclosure gathering
3. AWS S3 bucket monitoring
4. CVE scanning and notifications
5. Report generation

Table of Contents

Abstract	2
1 Introduction.....	5
1.1 Problem.....	5
1.2 Solution.....	5
2 Similar Applications.....	6
3 DevOps	6
3.1 Source Control Management	7
3.2 Continuous Integration/Continuous Deployment	7
3.2.1 GitLab	7
3.2.2 TravisCI	7
3.2.3 Jenkins.....	7
3.2.4 CodeShip.....	7
3.3 Code Testing and Code Quality	8
3.3.1 Hypothesis.....	8
3.3.2 Tox.....	8
3.3.3 Flake8	8
3.3.4 Django-testing	8
4 Container Orchestrator	8
5 Hosting	9
5.1 Blacknight Solutions	9
5.2 Amazon Web Services.....	9
5.3 Digital Ocean	9
6 Web Application	10
6.1 Django.....	10
6.2 Flask	10
7 Web Server	11

7.1 Nginx	11
7.2 Apache	11
8 Database	12
9 Django REST Framework	12
10 Amazon S3 Buckets	13
10.1 S3Scanner	13
10.2 S3-Inspector	14
10.3 AWSBucketDump	14
10.4 GrayHatWarefare	14
11 OSINT APIs.....	14
11.1 GitHub	15
11.2 Stack Exchange	15
11.3 Have I Been Pwned	15
12 Darknet Monitoring.....	16
12.1 Webhose.io Dark Web Data Feed	16
12.2 Target Monitoring/TorBot	17
13 CVEs	17
14 Nmap	17
15 Report Generation	18
16 Conclusions.....	18
16.1 Tech Stack.....	19
References	20

1 Introduction

1.1 Problem

Most companies have some form of online presence in current times. This online presence increases their attack surface and can make them potential targets to online attackers. Organisations also find it difficult to discover and act on relevant threat intelligence from open-source intelligence resources. Both of these problems are dependent on the lack of visibility, the visibility of an organisation's vulnerabilities to the internet, and the visibility of targeted threats and indicators of compromise regarding organisations.

1.2 Solution

The Heimdall project is envisioned to act as a unified threat intelligence platform and vulnerability discovery. It will be extensible through the use of a RESTful Django API, allowing for future modules to be easily integrated into the platform, as well as enable third parties to integrate their services. This aggregation of services in a single place should allow analysts to work more efficiently and develop a more contextualised vision of their company's attack surface, and potential threats.

The development of the project will utilise DevOps to ease the burden of both development and operations on the single developer responsible for implementing this project. DevOps will also speed up time to deployment, and through the usage of microservices, provide clear deliverables to indicate the progress of the projects development.

The Darknet module seeks to bring to light the possible indicators of compromise, or target threats present on darknet and hacker forums. Due to ongoing discussions with a data provider this module, this module will be one of two possibilities.

The open source intelligence module will pull data from sites commonly associated with accidental or purposeful leaking of propriety data. This will help identify any possible leakage from a company, and potentially uncover unnoticed breaches.

The Amazon Web Server S3 bucket scanner will attempt to identify bucket exposing their contents publically to the internet by scanning and attempting to access client specified addresses.

The CVE module has a dual function. Firstly, on the main dashboard there will be a ticker displaying the most recently released CVEs. This is to help analysts stay on top of the ever increasing number of vulnerabilities. Secondly, a simple nmap scanner will attempt to enumerate services from a client's IP range and correlate the results with the NIST NVD, thereby identifying any publically facing vulnerabilities and readying them for remediation.

The reporting module will seek to dynamically create reports on the activity registered by other modules. There will be two standardised report templates, a technical and executive. The technical report template will provide low level detail of the threats and vulnerabilities detected by the platform. This report is intended to provide as much context as possible to analysts, and ensure that they have enough information about a threat or vulnerability to choose the right course of remediation. The executive report is intended for managerial and C-level employees. It will be light on technical details, but focus heavily on trends and alert statistics to give manager the information they need to make effective strategic decisions about the security posture of their organisation.

2 Similar Applications

The Heimdall project is the composition of many different threat intelligence and vulnerability assessment modules. While individual modules have similar applications, and some even integrate said applications into Heimdall, there is no product on the market directly comparable to Heimdall. Heimdall is not a traditional security event and incident management (SIEM) console.

The closest application to Heimdall would be Fireeye's iSIGHT Threat Intelligence service.

3 DevOps

Given the current industry trend towards the increased adoption of DevOps [1], as well as this being an individual project, and experience with the methodology in a previous course module, I felt it would be beneficial to implement a DevOps production and deployment pipeline. The DevOps pipeline, while requiring more work for initial configuration, allows for an increased development and delivery speed [2]. This development lifecycle also allowed me to approach

my project using a microservice development strategy, allowing me to demonstrate my progress and create functional deliverables [3].

I then moved on to examining the specific pipeline and tools I will use to implement the methodology. The pipeline consists of several stages, researched as such:

3.1 Source Control Management

GitHub: The platform is proprietary. Provides an unlimited amount of private repositories, hosting up to 1GB per project. One of the most common SCM for open-source projects. Provides inbuilt issue tracking boards [4].

GitLab: The platform itself is open source and available for code review. Provides an unlimited amount of private repositories, hosting up to 10GB per project. Provides a full SDLC suite of tools. Provides inbuilt issue tracking boards. I have previous experience using GitLab for recreational project [5].

BitBucket: The platform is proprietary. Provides an unlimited amount of private repositories, hosting up to 1GB per project. Has Jira integration for issue tracking [6].

3.2 Continuous Integration/Continuous Deployment

3.2.1 GitLab

Provides integrated CI/CD functionality alongside SCM. Allows for custom Docker images to be used as a part of testing, bringing it closer to the production environment. A leading cloud CI tool by Forrester [7]. The platform itself is open source and available for code review.

3.2.2 TravisCI

The platform is proprietary. Integrates with GitLab code repository. Cloud hosted CI as a service. Unlimited build for open-source projects. \$63 per month for private repositories [8].

3.2.3 Jenkins

The leading continuous-integration server. The application itself is open source and available for code review. Runs as application on host machine [9].

3.2.4 CodeShip

The platform is proprietary. Integrates with GitLab code repository. Cloud hosted CI as a service. Up to 100 builds a month for free [10].

3.3 Code Testing and Code Quality

3.3.1 Hypothesis

Family of testing libraries that provides simple property-based testing¹.

3.3.2 Tox

Allows for automating tests against multiple interpreter configurations².

3.3.3 Flake8

Flake8 is a wrapper around several tools that perform static code analysis to verify code quality. These tools include PyFlakes, pycodestyle, and Ned Batchelder's McCabe script³.

3.3.4 Django-testing

In built testing framework. Tests are written using the unittest module in the Python standard library [11].

GitLab appeared to tick many of the boxes for DevOps development in a single environment, combined with my previous experience with the platform, and with a simple Flake8 test during the build stage, it seemed the best candidate for my DevOps needs. In conclusion, my DevOps pipeline is as such:

- SCM: GitLab (Private repository)
- CI/CD: GitLab
- Code Testing and Quality: Flake8

4 Container Orchestrator

As this project will be being developed on a single system, complex multi-system container orchestration systems like Kubernetes [12] or Docker Swarm [13] will not be necessary. However, the app itself will function with several microservices all running within their own

¹ <https://hypothesis.readthedocs.io/en/latest/>

² <https://tox.readthedocs.io/en/latest/>

³ <https://gitlab.com/pycqa/flake8>

container. This requires some level of host based container orchestration. Docker offer the Docker Compose utility to achieve this goal [14].

5 Hosting

The Heimdall project will require at least one server, preferably two, to manage the workload of both the console, and the modules. The scanning functionality of the majority of modules will require a significant amount of bandwidth in whichever solution is chosen. To maintain the integrity of the database, a dedicated solution would be preferable.

5.1 Blacknight Solutions

Blacknight Solutions is a hosting company based in Carlow Town, Co. Carlow, Ireland. They provide both cloud solutions and dedicated server hosting in two data centres located in Carlow Town, and Dublin. Both solutions are managed through a CPanel interface. The dedicated servers, while a preferred option for a product in production, are far above the specifications needed for the Heimdall project. The “Business” cloud hosting package would be suitable for the project, it is priced at €39.95 per month [15].

5.2 Amazon Web Services

Amazon Web Services (AWS) is the dominant cloud service provider, having 32% market share in Q4 of 2018 [16]. AWS provides on-demand virtual server hosting in the cloud or dedicated cloud hosting with their Amazon EC2 solution. The dedicated solution requires a minimum 1-year contract at the cost of \$14.38 [17].

5.3 Digital Ocean

Digital Ocean is an on-demand and dedicated cloud hosting provider. The Standard3 package suits the specifications of this project and is priced at \$15 per month, with 3TB bandwidth limit [18]. They also provided server images preconfigured with popular technology stacks and application, as well as managed dedicated databases.

For this project, a Digital Ocean cluster will be utilised. This solution gave a balance between the technical specification of the server, the price, and the bandwidth available. The availability of a managed Kubernetes system would also allow for a rapid expansion of the service capabilities should it be needed in the future.

6 Web Application

For this project I wanted to utilise a web application framework that would allow for a quick development timeframe, and some form of design templates to account for my lack of experience with interface and user experience design. I have experience with Python and as such researched web app frameworks implemented in Python. During this research I came upon the following frameworks:

6.1 Django

Django is a full-stack web framework for Python, this means that it comes with many necessary functionalities inbuilt into the framework [19]. Django has an inbuilt template engine to ease the design of user interfaces, and allows developers to customise these templates using the Django template language.

6.2 Flask

Flask is a lightweight and extensible Python web framework [20]. Out of the box it comes with very few functionalities inbuilt, instead relying on the use of extension to fulfil the functionality needed in a project. This then allows developers to be less reliant on the maintenance and security of third party libraries, of which they have no control over. As with Django, Flask utilises a template engine to ease the design of user interfaces, in this case Flask makes use of Jinja2, a system directly inspired by Django's template system.

Given the dynamic nature of the web console, the inbuilt functionalities present, as well as the large development community from which to derive support, I chose to utilise Django for the

web application framework in Django. This should help to speed up development time, and allow more focus on the development of the modules themselves.

7 Web Server

A web server is necessary to serve the pages and content provided by the web application framework. Due to the nature of microservices, the design methodology being applied to this project, a reverse proxy is necessary to act as a single gateway for requests calling different microservices [21]. To effectively route these requests, the proxy will utilise the service registry of the container orchestration product. Python web applications also require a web server gateway interface (WSGI) to correctly forward request [22]. Gunicorn will be the implemented WSGI in this project [23].

7.1 Nginx

Nginx is an open source web server originally created by Igor Sysoev and released in 2004, however currently it has expanded its capabilities and can also be used as a reverse proxy, load balancer, cache server, and more. According to statistics from Netcraft, in 2019 Nginx has become the number one used web server, holding a nearly 28% market share [24]. Nginx uses event-driven processes when dealing with requests, therefore allowing or better utilisation of hardware resources and for more requests to be served per second [25].

7.2 Apache

The Apache HTTP server is an open source web server developed by the Apache Software Foundation and released in 1995. Unlike Nginx, Apache uses a process driven model to handle requests. In this model, the server will pause the execution of a process until the previous step has been completed, this holds the processes assigned place in RAM and can lead to a large usage of the RAM resource [26].

For the project, I decided to utilise Nginx with Gunicorn, for one reason due to Nginx's event-driven design it has a smaller hardware requirement allowing me to cut down on the initial costs of this project.

8 Database

The database recommended by the Django Software Foundation is PostgreSQL. Django has functionality developed specifically to interact with PostgreSQL databases, as well as a wide variety of tutorials and documentation on its utilisation in the Django framework.

PostgreSQL is a free and open source object-relational database management system (ORDBMS) released in 1996. The object-relational design, in place of the traditional MySQL purely relational database design, allows for features such as table inheritance, the storage of JSON objects, and more. PostgreSQL also adheres very closely to the MySQL syntax [27].

The database for this project will be hosted on a separate VPS specifically designed to manage database systems.

9 Django REST Framework

The design methodology of microservices makes it necessary for the Heimdall console to utilise an API framework to both handle and create requests between services. The development on an API should also allow for easier development and integration of new modules into Heimdall in the future. The Django REST framework provides an answer to this problem.

Firstly, a RESTful API stands for a “representational state transfer application programming interface” [28]. An API is a set of defined functions that allow one program to call upon the functionality of another. A RESTful API is one which uses HTTP requests to call the API functionality. Generally, the method used, such as GET or PUT, and the contents of the parameters on the URL indicate the functionality and values of the function being called.

The Django REST Framework builds on top of an existing Django project and allows for the development of APIs following OpenAPI schemas [29]. The responses from such APIs are

typically, but not always, returned in the JSON (JavaScript Object Notation) format allowing for simplified parsing of the data. This project will utilise JSON formatting for API responses.

10 Amazon S3 Buckets

Due to the rising notoriety of exposed Amazon Web Service (AWS) S3 Bucket exposures following the Capital One breach which effected over 100 million people in the United States and Canada [30], I felt that a scanner capable of flagging any of said buckets relevant to the customer would work well as a module within the Heimdall platform.

Firstly I examined the insecurities in S3 buckets that give rise to these vulnerabilities. Within this research I discovered that “By default, all Amazon S3 resources—buckets, objects, and related sub resources (for example, lifecycle configuration and website configuration)—are private: only the resource owner, an AWS account that created it, can access the resource” [31]. This entails that the exposed S3 buckets have been improperly configured from their secure default. The security controls of a bucket however, may be changed at the user level, not just at the administrator level. This possibly help explain the misconfiguration.

There are several open-source AWS bucket scanners already available to the public. While in a real production environment the Heimdall product would utilise a proprietarily developed tool, the integration of an already existing open source tool will allow for a speedier development time of the project and help serve as a proof of concept for possible future expansions. The tool researched and most suited to the project were as such:

10.1 S3Scanner

This is an open source tool capable of scanning for buckets given specific information, either a bucket name, domain name, full S3 URL, or the bucket and region. Provides the ability to both identify open buckets and list their contents, in a targeted manner. This tool is also capable of being built and run using Docker, this is in line with the current vision of the project infrastructure [32].

10.2 S3-Inspector

Checks buckets for public access using valid user credentials. Requires IAM (Identity & Access Management) account configured for target buckets. This tool will indicate if a bucket is publically accessible or not, as well as display the permissions and URLs associated with the bucket [33].

10.3 AWSBucketDump

This tool attempts to enumerate open S3 buckets and their contents using predefined wordlists. This tool is more akin to a bruteforcer searching for open buckets and dumping their contents, than a targeted solution aiming to identify exposed buckets of a specific user [34].

10.4 GrayHatWarefare

This is not a scanner tool, but a database of open buckets compiled and managed by the owner of the site, accessible through API calls [35]. This is would require a dragnet approach of parsing through all the discovered buckets to identify those that relate to the business. This database also indexes the contents of exposed buckets, giving the ability to identify the exposed data to users. Some search functionality, and access to the full listing of 951 million files at time of writing, is restricted to paid premium accounts. However, registered free accounts do still have access to all listed buckets and 357 million files.

I chose to utilise S3Scanner due to its compatibility with the Docker framework I intend to utilise in the project infrastructures. The functionality provided also matched the needs of the project, as it simply requires a wordlist of buckets provided by clients, in comparison to a more in-depth tool such as S3-Inspector. I also felt that this more targeted approach was desirable over the large scale parse that utilising GrayHatWarefare.

11 OSINT APIs

There is a great many sources of open source intelligence (OSINT) available on the internet, too many to create logic for in the timeframe of this project. As such, the project will focus on

gathering information from a small number of popular sites. Most of these sites provide a public API to query content, this will ease the information gathering process. Some sources considered for this project were:

11.1 GitHub

GitHub is the most popular remote code repository, with over 1.1 billion contributions in 2018. Unfortunately, due to poor developer hygiene, proprietary code can be published publicly. To find these public repositories, Heimdall will make use of the GitHub API [36]. This is a REST (Representational state transfer) API that will allow Heimdall to search the contents of public repositories and commits for any indicators of information leakage. This API will be used in the project.

11.2 Stack Exchange

Stack Exchange is a network of question and answer websites covering a wide variety of topics, both technical and non-technical. The most popular of the technical sites is StackOverflow, a site dedicated to programming and related questions. There are also many more niche sites in the computing field, such as information security or database administration. The Stack Exchange umbrella site provides a public API to query the multi-site contents in a single request [37]. This API will be used in the project.

11.3 Have I Been Pwned

This is a resource where individuals can determine if their accounts have been compromised in any publically disclosed breaches or dumps, it is owned and operated by a Troy Hunt, a notable figure in the security community.

The site allows searches relating to email address and domain, however to make use of a domain search the user must authenticate ownership over the domain. There are four methods of verification involving either an email to the webmaster of the domain, insertion of a metatag in the head of the page at web root, uploading a file to web root, or creating a TXT record on the domain. All of these form of authentication are, in the current rendition of the project, incompatible with the service model. Therefore, domain search functionality will not be

implemented in this project. The email searching functionality however, is accessible to all, and for \$3.50 a month also accessible through an API [38]. This can be utilised to actively monitor possible account breaches for clients.

12 Darknet Monitoring

“Darknet” is a term used to identify sections of the internet not readily accessible to average users using conventional protocols. There are several different darknets that can be accessed through the use of special browsers or the use of proxies configured to use the appropriate network protocols. Some common darknet are:

- The Onion Router (Tor) [39]
- Invisible Internet Project (I2P) [40]
- Freenet [41]

Of these darknets, Tor is by far the most popular, hosting regular content such as Facebook, as well as more concerning content such as hacker forums or illegal marketplaces. A common occurrence on hacker forums is the sale of stolen information or accounts, or the discussion of vulnerable companies [42]. Therefore, this raw intelligence is a valuable resource that can be used to warn companies of impending or previously undetected attacks. Given the decentralised nature of Tor, gathering this data presented a challenge to which two solutions have been identified below:

12.1 Webhose.io Dark Web Data Feed

Webhose provide a feed of data gathered by their vast darknet scraping infrastructure. This data feed is utilised by entities from law enforcement agencies to threat intelligence providers [43]. Due to the potential of this data, access is partially restricted and must be approved following negotiation between Webhose and the requesting party. Following an inquiry with Webhose however, the cost of access to the service was deemed too high for the project.

12.2 Target Monitoring/TorBot

If Webhose access was to be denied, and again due to the size of the darknet, the next feasible monitoring solution for this project would be a targeted web scraper. One such web scraper developed for accessing Tor and clearnet (the surface internet) sites is TorBot [44].

TorBot is an open source web scraper that will access a predefined list of sites through the use of a Tor proxy, and then proceed to scrape all data from the site and store it locally, as well as producing a few more specialised lists of results such as emails and links. TorBot can be leveraged, in conjunction with a list of legitimate hacker forums both on the darknet and clearnet, to acquire the raw data equivalent to Webhose's solution, albeit in a much more limited scope.

13 CVEs

Common Vulnerabilities and Exposures (CVEs) are public entries of security issues that affect the systems we all use. Most large products have CVEs associated with them, and these CVEs are almost always remediated in software patches or updates. CVEs are archived in a standardised format which makes it easier to search through, and to identify applications or services running vulnerable versions. The American National Institute of Standards and Technology (NIST) maintain a National Vulnerability Database (NVD), containing all public CVEs. NIST also provide data feeds on this database in multiple formats, such as JSON, CPE (Common Platform Enumeration), or RSS [45]. This feed can be used to determine if a service or a specific version of a service is vulnerable.

14 Nmap

Nmap, or Network Mapper, is a free an open source tool for network and service discovery [46]. Nmap can be used to discover active services on a device and even numerate the version in some cases. It can achieve this by sending a wide variety of specific packets and interpreting the responses. Nmap can be configured to output results in a wide variety of formats, including CPE, a format particularly helpful when assessing service versions against NVD entries.

Nmap also includes the Nmap Scripting Engine (NSE) [47]. This allows the automation of simple lua scripts for a variety of network tasks. Nmap comes preinstalled with a wide variety

of scripts covering functionalities from vulnerability detection to host discovery. The NSE is a key component in the extensibility of nmap.

The automation of nmap scans is easily achieved with the python-nmap module of Python [48].

15 Report Generation

A key module within this project is the ability to generate reports on demand. There should be two main report templates, a technical and executive report. LaTeX is a document preparation system that will translate the contents of .tex file into a fully formatted document, for example a pdf, effectively automating document design [49]. Within a .tex file content sections are simply labelled according to the design format which will be applied to them, this disregard for the actual contents will allow me to populate .tex files with the relevant information and produce the final, formatted report in a pdf to clients. This process can be easily achieved using the PyLaTeX library for Python [50].

16 Conclusions

In conclusion, every module planned for project Heimdall appears to be technically feasible. Some modules will makes work of already existing projects, but his is to illustrate a working proof of concept for the proposed Heimdall framework. If Heimdall was to progress further, proprietary implementations of these open source would have to be developed. The proposed hosting and development pipeline should accelerate development time for a single developer. The cost of running the project should not be an issue, as most of the software utilised within the project is free and/or open source

16.1 Tech Stack











Area	Solution	
SCM	GitLab	
CI/CD	GitLab	
Code Testing Framework	Tox	
Hosting Platform	Digital Ocean, Standard3 package	
Container Orchestrator	Docker Compose	
Web Server	Nginx	
WSGI	Gunicorn	
Web App. Framework	Django	
API Framework	Django REST Framework	
Database	PostgreSQL	

Table 1, Project Technology Stack

References

- [1] Amazon Web Services, Inc., “What is DevOps?,” 2018. [Online]. Available: <https://aws.amazon.com/devops/what-is/devops/>. [Accessed 27 October 2019].
- [2] V. Farcic, *The Devops 2.0 Toolkit*, Learnpub, 2019.
- [3] V. Farcic, “Microservices,” in *The Devops 2.0 Toolkit*, Learnpub, 2016, p. 23.
- [4] GitHub.com, “Pricing . Plans for every developer,” [Online]. Available: <https://github.com/pricing>. [Accessed 31 October 2019].
- [5] GitLab.com, “GitLab.com Feature Comparision,” [Online]. Available: <https://about.gitlab.com/pricing/gitlab-com/feature-comparison>. [Accessed 31 October 2019].
- [6] bitbucket.com, “Pricing | BitBucket,” [Online]. Available: <https://bitbucket.org/product/pricing>. [Accessed 31 October 2019].
- [7] GitLab.com, “GutLab Analyst Report | GitLab,” [Online]. Available: <https://about.gitlab.com/analysts/forrester-cloudci19/>. [Accessed 31 October 2019].
- [8] Tracis CI, “Travis CI - Test and Deploy with Confidence,” [Online]. Available: <https://travis-ci.com/plans>. [Accessed 31 October 2019].
- [9] Jenkins Project, “Jenkins,” [Online]. Available: <https://jenkins.io/>. [Accessed 31 October 2019].
- [10] Cloud Bees, “Continous Integration Service Costs | Codeship Pricing,” [Online]. Available: <https://codeship.com/pricing>. [Accessed 31 October 2019].
- [11] Django Foundation, “Testing in Django,” [Online]. Available: <https://docs.djangoproject.com/en/2.2/topics/testing/>. [Accessed 31 October 2019].
- [12] The Kubernetes Authors, “Production-Grade container Orchestration | Kubernetes,” [Online]. Available: <https://kubernetes.io/>. [Accessed 1 November 2019].

- [13] Docker Inc., “Swarm mode overview | Docker Documentation,” [Online]. Available: <https://docs.docker.com/engine/swarm/>. [Accessed 1 November 2019].
- [14] Docker Inc., “Overview of Docker Compose | Docker Documentation,” [Online]. Available: <https://docs.docker.com/compose/>. [Accessed 1 November 2019].
- [15] Blacknight, “Cloud Hosting Solutions - Cloud VM - VPS Servers,” [Online]. Available: <https://www.blacknight.com/cloud-hosting/>. [Accessed 1 November 2019].
- [16] Canalys, “Cloud market share Q4 2018 and full year 2018,” 5 February 2019. [Online]. Available: https://www.canalys.com/static/press_release/2019/pr20190204.pdf. [Accessed 1 November 2019].
- [17] Amazon Web Services, “EC2 Instance Pricing - Amazon Web Services (AWS),” [Online]. Available: <https://aws.amazon.com/ec2/pricing/reserved-instances/pricing/>. [Accessed 1 November 2019].
- [18] DigitalOcean LLC., “Pricing on Digital Ocean - Cloud virtual machine & storage pricing,” [Online]. Available: <https://www.digitalocean.com/pricing/#Compute>. [Accessed 1 November 2019].
- [19] Django Software Foundation, “The web framework for perfectionists with deadlines,” [Online]. Available: <https://www.djangoproject.com/>. [Accessed 2 November 2019].
- [20] The Pallets Projects, “Flask | The Pallets Project,” [Online]. Available: <https://www.palletsprojects.com/p/flask/>. [Accessed 2 November 2019].
- [21] NGINX Inc., “Building Microservices Using an API Gateway,” 15 June 2015. [Online]. Available: <https://www.nginx.com/blog/building-microservices-using-an-api-gateway/>. [Accessed 2 November 2019].
- [22] Python Software Foundation, “PEP 3333 -- Python Web Server Gateway Interface v1.0.1,” 4 October 2010. [Online]. Available: <https://www.python.org/dev/peps/pep-3333/>. [Accessed 2 November 2019].
- [23] Gunicorn, “Gunicorn - WSGI HTTP Server for UNIX,” [Online]. Available: <https://gunicorn.org/>. [Accessed 2 November 2019].

- [24] Netcraft, “Netcraft | April 2019 Web Server Survey,” 22 April 2019. [Online]. Available: <https://news.netcraft.com/archives/2019/04/22/april-2019-web-server-survey.html>. [Accessed 1 November 2019].
- [25] NGINX Inc., “Inside NGINX: Designed for performance & Scalability,” 10 June 2015. [Online]. Available: <https://www.nginx.com/blog/inside-nginx-how-we-designed-for-performance-scale/>. [Accessed 2 November 2019].
- [26] B. R. M. K. Prakash P, “Performance analysis of process driven and event driven web servers,” in *Institute of Electrical and Electronic Engineers*, Coimbatore, India, 2015.
- [27] The PostgreSQL Global Development Group, “PostgreSQL,” [Online]. Available: <https://www.postgresql.org/>. [Accessed 2 November 2019].
- [28] M. Wittmann, “What is a RESTful API? | PROAD,” PROAD Software, 17 May 2019. [Online]. Available: <https://www.proadsoftware.com/en/blog/entries/2019/05/what-is-a-restful-api.php>. [Accessed 2 November 2019].
- [29] Django REST Framework Organisation, “Documenting your API - Django REST framework,” [Online]. Available: <https://www.django-rest-framework.org/topics/documenting-your-api/#generating-documentation-from-openapi-schemas>. [Accessed 2 November 2019].
- [30] Reuters, “Capital One customer data breach rattles investors,” Reuters, 30 July 2019. [Online]. Available: <https://www.reuters.com/article/us-capital-one-fin-cyber-amazon-com/capital-one-customer-data-breach-rattles-investors-idUSKCN1UP1LD>. [Accessed 1 November 2019].
- [31] Amazon Web Services, “Identity and Access Management in Amazon S3 - Amazon Simple Storage Service,” [Online]. Available: “By default, all Amazon S3 resources—buckets, objects, and related sub resources (for example, lifecycle configuration and website configuration)—are private: only the resource owner, an AWS account that created it, can access the resource”. [Accessed 1 November 2019].

- [32] sa7mon, “Sa7mon/S3Scanner: Scan for open AWS S3 buckets and dump contents,” GitHub, [Online]. Available: <https://github.com/sa7mon/S3Scanner#using-docker>. [Accessed 2 November 2019].
- [33] kromtech, “kromtech/s3-inspector: Tool to check AWS S3 bucket permissions,” GitHub, 24 April 2018. [Online]. Available: <https://github.com/kromtech/s3-inspector>. [Accessed 2 November 2019].
- [34] jordanpotti, “jordanpotti/AWSBucketDump: Security Tool to Look For Interesting Files in S3 Buckets,” GitHub, 19 June 2019. [Online]. Available: <https://github.com/jordanpotti/AWSBucketDump>. [Accessed 2 November 2019].
- [35] GrayHatHarfare, “<https://buckets.grayhatwarfare.com/packages>,” [Online]. Available: <https://buckets.grayhatwarfare.com/packages>. [Accessed 31 October 2019].
- [36] GitHub Inc., “GitHub API v3 | GitHub Developer Guide,” [Online]. Available: <https://developer.github.com/v3/>. [Accessed 2 November 2019].
- [37] Stack Exchange, Inc, “Stack ExchangeAPI,” [Online]. Available: <https://api.stackexchange.com/>. [Accessed 2 November 2019].
- [38] T. Hunt, “Have I Been Pwned: API v3,” [Online]. Available: <https://haveibeenpwned.com/API/v3>. [Accessed 2 November 2019].
- [39] The Tor Project, Inc, “Tor Project | Anonymity Project,” [Online]. Available: <https://www.torproject.org/>. [Accessed 3 November 2019].
- [40] I2P Team, “I2P Anonymous Network,” [Online]. Available: <https://geti2p.net/en/>. [Accessed 3 November 2019].
- [41] Freenet project, “Freenet,” [Online]. Available: <https://freenetproject.org/index.html>. [Accessed 3 November 2019].
- [42] A. Cuthbertson, “Dark web data dump sees 620 million accounts from hacked websites go on sale,” The Independent, 13 February 2019. [Online]. Available: <https://www.independent.co.uk/life-style/gadgets-and-tech/news/dark-web-data->

hackers-dubsmash-myfitnesspal-myheritage-cyber-security-a8775666.html. [Accessed 2 November 2019].

- [43] Webhose, “Dark Web Data API,” [Online]. Available: <https://webhose.io/products/darkweb-monitoring/>. [Accessed 3 November 2019].
- [44] DedSecInside, “Dark Web OSINT Tool,” [Online]. Available: <https://github.com/DedSecInside/TorBot>. [Accessed 3 November 2019].
- [45] National Institute of Standards and Technology, “NVD - Data Feeds,” [Online]. Available: <https://nvd.nist.gov/vuln/data-feeds>. [Accessed 3 November 2019].
- [46] G. Lyon, “Nmap: the Network Mapper,” [Online]. Available: <https://nmap.org/>. [Accessed 3 November 2019].
- [47] Nmap.org, “NSEDoc Reference Portal,” [Online]. Available: <https://nmap.org/nsedoc/>. [Accessed 3 November 2019].
- [48] norman, “Python-nmap . PyPI,” Python Package Index, 26 July 2016. [Online]. Available: <https://pypi.org/project/python-nmap/>. [Accessed 3 November 2019].
- [49] LaTeX3 Team, “LaTeX - A document preparation system,” [Online]. Available: <https://www.latex-project.org/>. [Accessed 3 November 2019].
- [50] JelteF, “PyLaTeX . PyPI,” Python Package Index, [Online]. Available: <https://pypi.org/project/PyLaTeX/0.6.1/>. [Accessed 1 November 2019].