SMART CITY OPEN DATA NETWORK SYSTEM: OPENNESS, SECURITY, AND PRIVACY

A Thesis
IN
Computer Science

Presented to the Faculty of the University
of Missouri Kansas City in partial fulfillment of
the requirements for the degree

MASTER OF SCIENCE

by
MOHAMMED ALMANSOORI

B. S., University of Garyounis, Benghazi, 2008

Kansas City, Missouri
2017

SMART CITY OPEN DATA NETWORK SYSTEM: OPENNESS, SECURITY, AND PRIVACY

Mohammed Almansoori, Candidate for the Master of Science Degree

University of Missouri Kansas City, 2017

ABSTRACT

The increasing concentration of population around the cities poses challenges in their operation and services. On the other hand, the current technological revolution allows scalable and innovative means to better serve the public. Many city governments are collecting, publishing and analyzing more data from diverse sources including IoT sensors. City government's open data provides multiple values such as improving transparency of the government, enhancing the efficiency of its operations and services and attracting more businesses to the region. However, the resulting data systems, called Open Data Portals (ODPs) become more complicated and create the issues of accessibility, security, and privacy.

Extensive analyses of ODPs of many cities around the world using diverse methodologies have been performed. We find that the extent of openness of data and popularity, and the level of security of ODPs are highly diverse across the cities. We then provide

the recommendations for improving security measures of ODPs. Considering the privacy issues of data in ODPs, we provide a tool to automatically filtering Personally Identifiable Information (PII).

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering, have examined a thesis titled "Smart City Open Data Network System: Openness, Security, and Privacy," presented by Mohammed Almansoori, candidate for the Master of Science degree, and hereby certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Baek-Young Choi, Ph.D., Committee Chair
Department of Computer Science & Electrical Engineering

Sejun Song, Ph.D.
Department of Computer Science & Electrical Engineering

Deep Medhi, Ph.D.
Department of Computer Science & Electrical Engineering

CONTENTS

ILLUSTRATIONS

TABLES

ACKNOWLEDGEMENTS

I would like to thank my advisors especially Dr. Baek-Young Choi for her advice during my Master's study. I worked under Dr. Choi's guidance through a two years journey of a continuous gain of new skills which shaped my researches insight. I would like to thank my committee members: Dr. Dr. Deep Medhi and Dr. Sejun Song for their help and insightful comments in my thesis.

I also sincerely thank the Professor Tony Luppino, the Law professor at UMKC, for proving a wide and diverse space to our work direction in terms of meetings with Kansas City officials and several important meetings with Smart Kansas City vendors. I would like to thank Eric Roche, Chief Data Officer of Kansas City for giving us a great opportunity to understand and analyze an important real-life study case about my thesis which is smart city system internetworking of Kansas City, and Bill Mullins at Better Choices Consulting for his insightful discussions. Thanks to these comments, this thesis work was improved significantly by them.

My work and interaction with Dr. Song reflected a great progress towards my thesis, through exposing our lab team to several practical research topics that were related to my topic especially IoT applications. I thank my lab-mates as I had a great experience with them such as teamwork projects and our weekly meeting and our rich lab resources too. I would like to thank and appreciate the great support of my friends Ahmed Albishri, Khalid Almalki, and Mustafa Maswadi who proved the real meaning of the friendship.

I appreciate my parents as they were the hope that encourages me along my study

journey and especially in the tough times, and their prays were enlightening my way. My father in law Ibrahim, who was the friend, the father, and the supporter in my toughest times.

I really hoped that I would share this special feeling in my life with my best friend Mahir as we started this journey together three years ago but he passed away a month before my graduation.

My wife Sarah, who supported and encouraged me without hesitation along all my study years sacrificing her graduate study to give me the chance. My little smart daughter Sama who makes me smile whatever the situation is. My little son Jad, who always insisting calling me while I was working in the lab. The total thank is to God, who always lead and guide my way, and know better than me about my way even though I cannot see what life holds in store for me.

Mohammed Almansoori,

December 2017

CHAPTER 1

INTRODUCTION

As the population in different cities and the general state of technology are correlated, it is becoming more challenging to manage and control these cities to provide the acceptable level of services. Including traffic and parking issue many other strategic problems that still need effective solutions such as the utilities (water, electricity, waste management etc.). These needs raised the idea of finding an automated and effective way to capture and measure the real usage for long periods of time by using sensors, such as camera, data traffic sensor, pollution sensor, light sensor, vibration sensor, temperature sensor, etc. Data from different sources are published in ODP (Open Data Portal) and are faster and easier to access, even across national boundaries.

There are multiple benefits of making city data available publicly. The government transparency greatly enhances the ability of policymakers and scientists to complex development problems. It can also help finding effective solutions for cities problem as this data can be analyzed by the services vendors to provide better service quality, as they can use these real measurements to find effective solutions for their future planning and load balancing. In addition, the open data is an effective means to attract business and development investors as they can realize how they can spot specific city issues and provide a service (or solution) in an efficient way. It also shows how a particular city is developed and transparent [11]. The captured data usually contains sensitive and private information

such as people locations, faces, cars license plates, addresses, phone numbers, etc. Therefore, the data needs to be filtered before releasing it to the public to maintain the people information privacy and security. Examples of the captured data and what sensitive information may contain are provided in the table 1.

Table 1: Data captured, sensor type, and possible sensitive information

Data Source	**Captured Data**	**Sensitive Info**
Traffic camera	Video record	Faces, Car license plate, Location
DHCP server	Connection info	Device type, MAC, Location, OS, Carrier
Police reports	Crime records	Addresses, Phone numbers, Names, PII

In this work, an analysis of 10 ODPs for 10 cities around the world is conducted to identify the strength and weakness areas for each and what are the common system environments that are being used. Our selection for these cities based in seeking to include local and international diversity as possible, including the cities of reliable and complete log records, and selecting the cities which their comparison metrics are available (classification) such as datasets. The findings can help any new smart city or open data initiative, or even any already established smart cities to see what are the trends and how they can improve their weaknesses comparing to other cities considering the city size and population to find the best city match.

The Open Data Portals (ODP) are web applications that are directly connected to the governmental data servers where the data stored, filtered, and posted to the public.

That means the traffic between the ODP and the server contains very important information about the server side, further this traffic represents an access point to these servers. Therefore, any traffic from/to/through the ODP must be secure and monitored. Even though the data content that is published on these portals is not confidential, but at least its integrity (correctness) is crucial. Further, this data is used for a high level of strategic planning or in the research field that will be relied on, so the data integrity must be protected. Protecting the security of smart city and the economic value is becoming more critical [26].

In this work, first, the analysis is carried out for the smart city system and provide detailed explanations for its subsystems in terms of topology, interconnecting, and functionality. It is to provide the in-depth understanding of ODPs and smart city systems starting from the data source (i.e., sensors) up to the data portal interfaces. The analysis of the ODPs for 10 cities around the world is by identifying the essential metrics to evaluate the quality and the performance of any ODP. Then, the measures of the most important metrics for the cities such as the number of visits and the population has been taken. Further, apply an in-depth analysis to explain these results in a meaningful representation. For example, the number of visits can represent indications for how popular an ODP is, how relevant contents it has, how much interests the city draws from various entities such as individuals, companies, businesses. Finally identifying the issues of privacy and security of the cities ODPs and propose the best possible technical solutions based on the ODPs system types.

CHAPTER 2

RELATED WORK

2.1 Background

The quality and value of ODPs significantly depend on the datasets. Not only the types and duration of data but also their organization and the possibility of classification or filtering are integral to serve its purpose. Data organized in a way that is easy to access, search, and classify would increase its utility. Quality metrics for ODPs are discussed in [12], that includes the presence of metadata to enable users to search, filter, classify the data in an efficient way. The issue becomes more significant, as the volume of data gets increased. ODP administrators may exclude any unorganized data form ODP, as open datasets would be difficult to manage if the metadata is missing or wrong.

There are studies that compare and rank the city ODPs by designing quantifying indices, such as [12, 17, 27, 32]. However, these indices do not consistently present fair comparisons across for all the ODPs, because the cities practice diverse ways in how they aggregate the datasets classification, population, and the audience interests and the level of the city development vary.

2.2 State of Art

Several data platforms (PaaS) used for governments and public sector institutions have been compared in [2]. Most of them use cloud-based services, and they include

Ckan [14], Dkan [22], Junar [23], Socrata [29], Prognoz [24], and Opendatasoft [19]. During the study, these platforms were identified to provide a very much similar type of services in terms of functionality, access and control levels. While those cloud-based data platforms may be similar, the quality of ODPs depend on the ODP administrators in making their service interface (ODP) efficient, powerful, rich in content, and regularly updated. It is similar to the analogy between a web hosting service (i.e data platform provider such as Socrata) and a website developer (i.e. ODP administrator).

In [27], the authors discuss the need for standard features and requirements of ODPs and present their effort to provide a generic, effective and unified framework for ODPs in HOMER Project [8].

As the data is being collected from different types of sources and sensors, then it will be transferred through a networking media to the main data center (sometimes a storage cloud). The captured raw data often contain personal and private information of the citizens, thus it is very important to filter or even anonymize the information in order to prevent leaking the citizens' PII [34]. The administrators have the control of access to the stored raw data, so they format, filter, and then publish the filtered data on an ODP [10]. As ensuring the private information is a must, an effective and automatic filtration method needs to be used [34].

A thorough analysis was performed by testing the actual traffic for all the ODPs and considering the functionality, the structure, and the content of the ODP. This consideration allowed us to identify a list of the most dangerous and common vulnerabilities that any ODP. Actual multi-method tests were applied to all the targeted ODPs to address

the weaknesses and the strengths as well. A quantifying evaluation is provided for all the cities based on the security practices detected by our tests.

CHAPTER 3

SMART CITY OPEN DATA SYSTEM INTERNETWORKING

3.1 Smart City Open Data Ecosystem

Smart city open data ecosystem: In conventional open data portal (shown in Figure 1), government and different departments collect data from citizens and upload collected data to the portal directly, so there are fewer security vulnerabilities and filtration of personal information is easier. However, smart cities open data portal is quite complex as an infrastructure as well as the data flow. Figure 2 is an illustration of the smart city open data ecosystem. In the smart city, ODP sensors, citizens, and different city departments (police department, fire department, emergency medical service, 311 call, housing and others) are the basic data source. Some are directly connected to ODP as a source also as a user, and some are (like sensors) using network infrastructure to upload data to the portal. 3rd party data analytic organization does the analysis of data from ODP for the business decision, policymakers also for the government. Even in most of the cases, 3rd

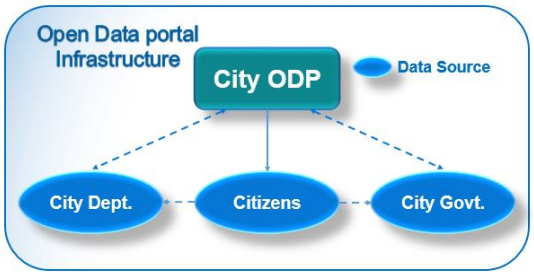


Figure 1: Conventional open data system

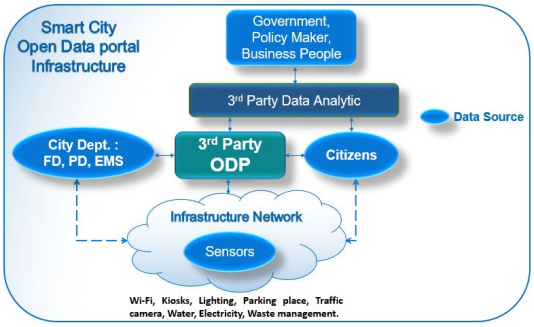


Figure 2: Smart city open data ecosystem

party handles the open data portal itself (like Socrata is one of the popular data handlers in US ODPs). Involvement of different network provider and 3rd party data analytic organization makes the smart city open data ecosystem more vulnerable in different aspects. Therefore, smart city open data ecosystem requires more and complex security system to prevent unauthorized data leakage and requires privacy insurance at different levels.

3.2 Smart City System Internetworking: Revenue Sharing

Smart city system usually controls many of the critical services in the city such as utilities, traffic flow, and sometimes not critical services such as weather forecast, guidance services for tourists, and so on. Kansas Smart City officials and their technical staffs gave us a clear depiction of the smart city System-of-Systems. This system includes a mix for both of the public and private sectors as they integrate their resources to provide several specific intelligent services such as smart lighting, open Wi-Fi, information Kiosks, and so on. The vendors participating in Kansas Smart City are Cisco, Google, Sprint, Sensity (Verizon), ThinkBig and others. Cisco is providing the controlling nodes for the networking such as switches, routers, APs, Datacenter cloud (i.e. CDP). Google is providing the networking links infrastructure through Google Fiber. Sprint provides the open Wi-Fi in Kansas City, using the infrastructure of Cisco and Google. Sensity is providing the smart lightening, using the infrastructure of Cisco and Google. ThinkBig is accessing the raw data captured by the smart city system (in Cisco CDP cloud storage) to filter, process, analyze, and publish it. Cisco is providing the Networking infrastructure and Sprint is operating part of it to provide the open Wi-Fi in Kansas City downtown

area. Sprint is collecting statistical data from the network user's devices to improve their services.

Data flow between the data storage and each of smart services sensors is not bidirectional, as a result, raw data are only flowing from sensors to storage cloud but they are not able to pull it back. Other smart services can access a filtered version of these data to provide their smart services. Smart city components are using the cloud resources which are very efficient usually. However, in some cases private organization keeping a copy of the captured data from sensors for their analysis, which can be a security issue.

CHAPTER 4

SMART CITY OPEN DATA SYSTEM SECURITY

4.1 Raised Issues and Background

The OPDs are web-based applications that are serving as an interface for visitors to send requests to the database engine on the open data. These portals are becoming more popular and their content is growing rapidly, so more types of visitors are accessing them using different connection types such as HTTP and API. The ODPs are containing a wide variety of information sets, such as crime incidents, utility outages, traffic information, and so on. As the ODP publishes the records of a city size, therefore it is important to use efficient automated methods to collect the data, filter, and format it to make it ready for publishing on the ODP. The collected data usually contain confidential information such as the personal identification and contact information of the people who are included in any record that the smart city or the open data database is collecting. Therefore, a filtration step is required to be done in an efficient and smart way. Let's take an example of someone who is filling an online form to report a violation or incident close to their home or their neighbor. It happened that a person to mention a private information in the More-Details box such as names, phone numbers, birth dates, car records, full addresses, and so on. This report will be saved to some database that the open data system has access to. If the open data system filters the columns of the private data and not scanning every single column such as the More-Details column, then all that column content will be published

with its private information. This is a privacy issue that we will be discussing in a later chapter.

As the ODP is an interactive interface between the visitors and the database engine of the open data system which is connected to the row and unfiltered data, further, the ODP is a part of the smart city system which is connected normally to very sensitive governmental systems such as the police records, the traffic control system, etc. Therefore, the ODP can be considered as a possible access point to many sensitive systems, so it needs to be secured. Further, as the ODP visitors numbers are in a rapid increase, then ODPs can be used to compromise or attack these visitors if these ODPs have not been secured.

The ODP content is used in analyzing city problems and needs for the future planning and big size of possible projects of investments and development. Therefore, the integrity of the ODPs is important.

The ODPs availability and accessibility are important for whoever is connected to them to pull data and especially the API connections that are used in real time applications and websites that provide real-time services such as the traffic status, the weather, the crime information and so no. So, the ODPs availability is important and need to be maintained and protected from any possible attacks that affect their availability such as DDoS attacks.

Research Scope: Based on the ODP classification as web-based application, we started to study the web-based application security issues and focusing on the most common ten web applications vulnerabilities that are; Injection, Broken Authentication

and Session Management, Cross-Site Scripting (XSS), Broken Access Control, Security Misconfiguration, Sensitive Data Exposure, Insufficient Attack Protection, Cross-Site Request Forgery (CSRF), Using Components with Known Vulnerabilities, and Underprotected APIs. Next, we narrow our scope to only six of these ten security issues that are related to the ODPs functionalists. For example, the Broken Access Control is a security issue that can affect the web-based applications such as email web applications and e-commerce and online shopping but it is not important for the ODPs. The six security issues that are specific to ODPs are; Injection, Sensitive Data Exposure, Broken Authentication and Session Management, Cross-Site Scripting (XSS), Using Components with Known Vulnerabilities, and Underprotected APIs.

Investigation area: To investigate the above six security issues we have two investigation areas. First, is looking at a higher level of the application which is the browser level which includes the client side only. Second, is the deeper level of the communication protocol which includes all of the server side, networking phase, and the client side. Securing the higher level applications such as the browsers is not ultimate solution as this thing depends on many factors that are significantly varying such as, the browser type and version, the operating system, the security applications used, and the client awareness for security issues. So securing at high levels is not a stable and rigid solution. Instead, we prefer to investigate and find a solution in a deeper level such as the communication protocol (i.e. HTTP) and make the server enforce the solution so it will include all of the server side, the networking medium, and the client side. Therefore, we investigated how we can secure the communication protocol which is the HTTP so that we can secure

these six ODPs related security issues. We found that there are many security settings and practices that can be applied to avoid these security issues, mainly including; setting the related security HTTP settings (flags), and utilizing and integrating these settings with the programming side of the ODP. The related HTTP security settings are; XSS injection protection and Defense-In-Depth with Content Security Policy, Broken Authentication and Session, Management with Cookies Security and privacy and X-XSS-Protection, Cross-origin Resource Sharing, HTTP Public Key Pinning, HTTP Strict Transport Security, Redirection, Referrer Policy, Subresource Integrity, X-Content-Type-Options, and X-Frame-Options.

Evaluating the security settings for ten ODPs: We used several methods to test the security settings for each ODP of the ten ODPs, and then we quantified these settings to be able to compare the ten ODPs.

4.2 Security Analysis Methodology

We analyzed the HTTP traffic interactions (requests and responses) between the client and the ODP server to find out the setting values of these flags. We are listing and explaining in this chapter all the methods that we used in our investigation. The most important, frequent, and up-to-the-date security vulnerabilities that can the ODP's have were targeted. Later in this section, these vulnerabilities will be explained with the possible risks behind them and the most successful solutions for these security gaps. By considering two concepts essentially in our analysis; the most dangerous, and frequent vulnerabilities, and the HTTP specific vulnerabilities.

- **Running security tests:** Several security tests such as SUCURI [31], SSL Labs [30], Quttera [25], Site Guarding [7], and Web Inspector [9], that are specialized for web applications to test the web traffic security related information.

- **HTTP Header analysis:**

- **cURL analysis with C/C++:**

- **Sniffing the HTTP traffic with raw web sockets:** Raw sockets programming is the deepest and accurate method to capture the network traffic for all the network layers. it can be considered the lowest levels method. Because they provide the direct access to the network traffic data units (frames, packets, segments) by capturing the incoming ones and generating the outgoing ones. It is a huge number of data units that can be captured by the raw sockets so it is a difficult task. Further, analyzing the incoming packets is much more difficult, so the well-known software Wireshark is used to capture the incoming traffic that is originating from the ODP's. the main interest is in a specific traffic that belongs to the HTTP protocol and especially the HTTP traffic that manages the connection between the client and the ODP server. To do that I used very precise filters to the traffic captured by the raw sockets to focus and analyze that traffic we are interested in.

  With this method, it is possible to see the traffic in the bit level up to the original information such as the HTTP flags that set the connection specification. Based on the ODP server setup that is used we can identify the security strengths and weaknesses and even predict the possible risks as it will be discussed in detail later

in this section.

- **Remote Access:** The normal way for any client to access a file or page on any web server is by using any web browser that all using the HTTP protocol. However, to manage the server remotely it is very common to use an SSH connection with the proper credentials to change the configuration or updating any content based on the privilege level of the user credentials. The security issue in this connection can raise in two common usages. First, when SSHing any server with the wrong credentials, the default reaction for the server is by replying an access error but with accurate and up to date information about the server such as the server's operating system type, distribution, and the version. This is the first successful step of any hacking operation which can lie under reconnaissance phase, which can lead to identifying the security vulnerabilities for that server which leads to 'Using Components with Known Vulnerabilities' problem that is discussed later in more detail. The second security issue is that when accessing any file or page on the server using SSH connection it will show the content source code and the protocol header information which includes the connection setup including the security setup, such as the HTTP security flags.

- **Fetching the HTTP traffic by the browser-level tools and addons:** As the most important communication that OPD's using is HTTP that is established by the web browsers. the web browsers at the client end are applying very deep analysis for the HTTP traffic to present the web content in an appropriate way that the web server specifying. therefore, web browsers are able to provide a great level of the traffic

of the ODP's. Most of the major web browsers (such as Firefox, Chrome, Internet Explorer, etc) are providing a collection of sophisticated development tools that are very helpful for web developers. this collection usually includes a traffic analysis tool that focuses mainly on the HTTP traffic and the bandwidth consumption for the active sessions.

- Browsers built-in tools: Firefox is equipped with a very useful tool that provides a comprehensive collection of real-time measures and traffic information such as a detailed and organized view for the HTTP request and responds headers and their flags with a briefed hints about the interesting values as figure 23 showing a screenshot for it. Chrome is providing less sophisticated developer tools set especially in the networking part that provide only a very limited information about the HTTP traffic as figure 4 is showing a screenshot of it. Microsoft Edge is providing lightweight set of tools that are well organized and providing very useful networking tool that presents real-time HTTP information including some of the HTTP flags values as figure 5 showing is a screenshot of it.

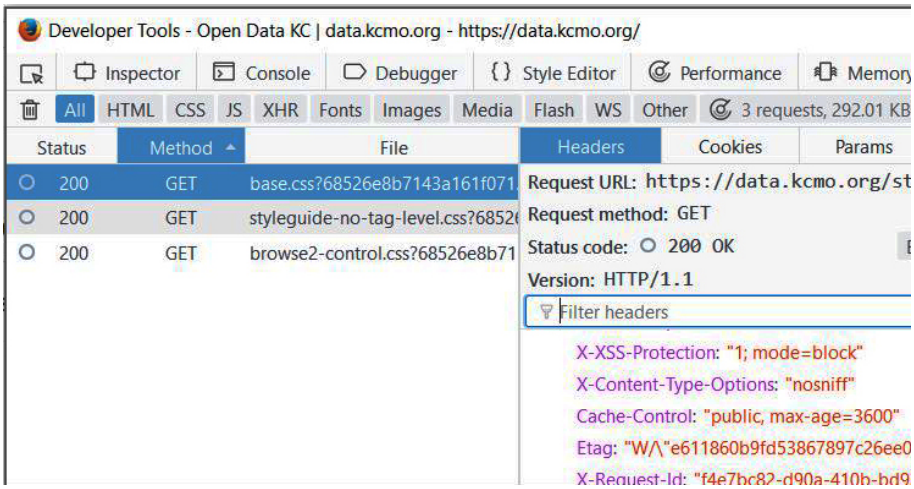


Figure 3: Firefox Network Analyzer

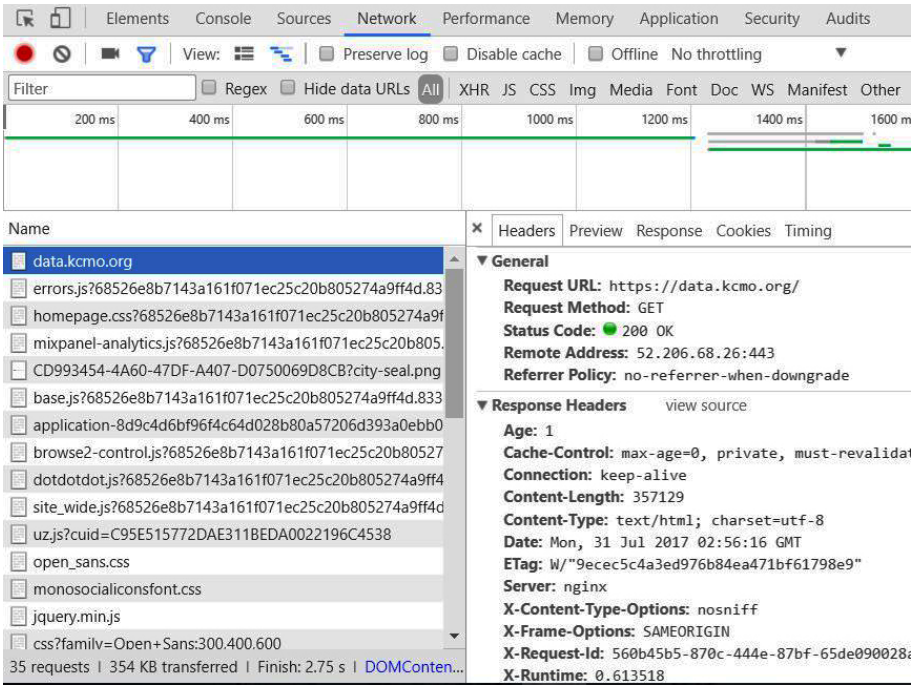


Figure 4: Chrome Network Analyzer

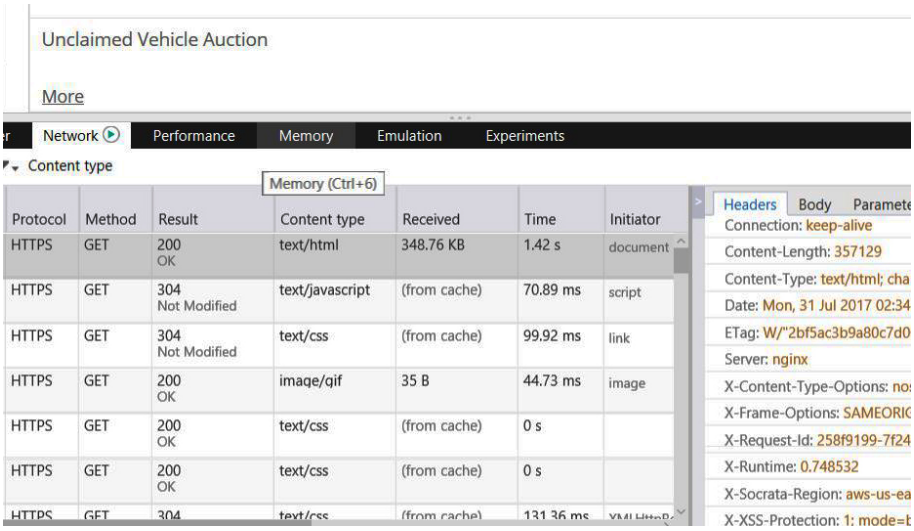
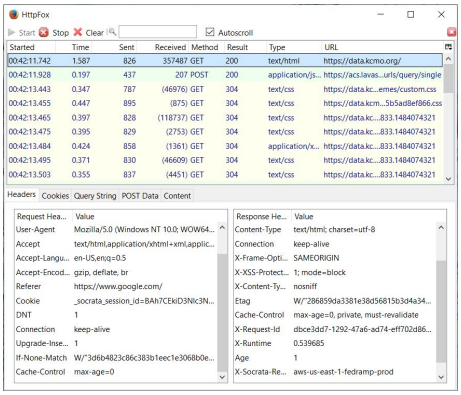
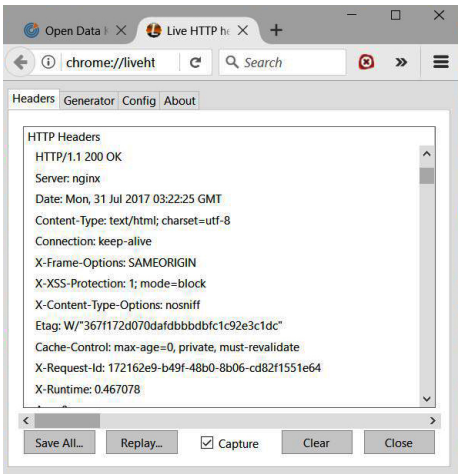


Figure 5: Microsoft Edge Network Analyzer

- Third party specialized add-ons: Live HTTP is one of the add-ons that can be used effectively in both Firefox and Chrome web browsers which provides most of the important HTTP traffic setup details in an organized and easy to read way. HttpFox is another web browser add-on that is compatible with Firefox only but it provides great details about the HTTP traffic and especially the headers flags with a very easy way to read and understand the flags values in addition to other features such as pausing and filtering the captured traffic. Figure 6 is showing screenshots of both add-ons



(a) HttpFox add-on



(b) Live HTTP add-on

Figure 6: Third party web browsers add-ons

4.3 Top Ranked Web Security Vulnerabilities

Since the ODP's are considered as database web applications, then a key security factor is the communication protocol which is HTTP (Hyper Text Transfer Protocol). This protocol plays the main role about how the web application behaves in; the server side (the web and database server(s)), client-side (the browser or API requests), and the communications in between them (the network). We did a research to identify the most dangerous and frequent web security vulnerabilities, so we found that The Open Web Application Security Project [20] (OWASP) 2017 report is containing the major part of them:

- Injection
- Broken Authentication and Session Management

- Cross-Site Scripting (XSS)
- Broken Access Control
- Security Misconfiguration
- Sensitive Data Exposure
- Insufficient Attack Protection
- Cross-Site Request Forgery (CSRF)
- Using Components with Known Vulnerabilities
- Underprotected APIs

4.4 HTTP Security Vulnerabilities

HTTP is the application layer protocol that web traffic uses and it is very exposed to the users' end and they are familiar with it. that makes it easy to read (or sniff), and easy to modify it or control it with high-level programming languages and no deep knowledge needed usually. In fact, HTTP has two types of important information; the data transferred by the HTTP protocol and the HTTP connection information. Both of these categories need to be secure. Simply, the data is being transferred by the HTTP protocol usually can be confidential or at least its integrity is important, so encrypting this type is important. the encrypted version of HTTP is HTTPS that encrypt the data and matches a signature between the connection ends. The other part of the HTTP information is the connection

information, that part is called the HTTP header. The HTTP header contains very important information such as the configuration and setup used of the connection established between the client and the server. each part of this information is separated in a specific field inside the HTTP header. in this analysis, we will describe in detail the HTTP header fields that are directly related to security settings, and how we already used these fields in our analyses to find out what web security vulnerabilities can exist in the open data portals that we targeted in our study.

4.5 Content Security Policy (CSP)

Among all the flags mentioned above, many of them that are security related HTTP parameters or flags. We investigated to find out the most relevant and effective parameters.

XSS and Defense-In-Depth: To make sure that the client that is requesting a web page content from a web server will receive the right contents (page, files, etc) and all its dependencies (fonts, styling files, scripts libraries/files, etc). sometimes, the dependencies are loaded from a third-party server(s)/website(s). for example, requesting a page from a tutorials website that contains a video from youtube.com, a bootstrap styling file from w3schools.com, and a font file from one of Google servers. In this example, the client side will receive files from four different sources, so how clients make sure that all these files are from these sources only? If a client is not sure then hackers are using this vulnerability to inject a malicious script (JavaScript mainly, VBScript, ActiveX, and Flash) that will load into clientâs browser and execute at their end, and usually, it will send a sensitive

information back to the hacker such as a cookie session information. Such scripts can be injected into the clientâs browser within (but not limited to); script, body, img, iframe, input, link, table, div, and object tags. For example, the img-src attribute of the element img can be changed to any malicious value or file address. To make sure that the web session interaction happening between the right ends a strict list (or whitelist) of all the servers that are needed to be involved in that session must be identified in advance by the web server. These set of policies are represented by the CSP HTTP parameter. This vulnerability can include any of cross-site scripting, injection, sensitive data exposure, and broken authentication and session management of the ODP specific web vulnerabilities listed in the next.

4.6 XSS and Defense-In-Depth Prevention

The strict white list method: BY adding the nonce attribute to any of both script and style elements. It will hold a nonce unique authorization value. The nonce value can be generated by any programming language function that supports cryptographically strong random function such as Python. Now, each of these elements has its own unique authenticator that must match the ones that are listed in the CSP policy that is set on the web server. In this case, any unauthenticated element or attribute will not pass to the client. This method is more accurate than just identifying a whitelist of ULRâs for the servers that are involved in the web request as if the client uses lax CSS parsing algorithm it still possible that a malicious style sheet can be grabbed by the user agent parameter.

XSS Prevention: The strict white list method: BY adding the nonce attribute to

any of both script and style elements. It will hold a nonce unique authorization value. The nonce value can be generated by any programming language function that supports cryptographically strong random function such as Python. Now, each of these elements has its own unique authenticator that must match the ones that are listed in the CSP policy that is set at the web server. In this case, any unauthenticated element or attribute will not pass to the client. This method is more accurate than just identifying a whitelist of ULRâs for the servers that are involved in the web request as if the client uses lax CSS parsing algorithm it still possible that a malicious style sheet can be grabbed by the user agent parameter.

This vulnerability can include any of Cross-Site Scripting, Injection, Sensitive Data Exposure, and Broken Authentication and Session Management of the ODP specific web vulnerabilities listed in the next.

As these tags can be found in any part of the web page, very useful recommendations were delivered by Open Web Application Security Project (OWASP).

- Never Insert Untrusted Data Except in Allowed Locations
- HTML Escape Before Inserting Untrusted Data into HTML Element Content
- Attribute Escape Before Inserting Untrusted Data into HTML Common Attributes
- JavaScript Escape Before Inserting Untrusted Data into JavaScript Data Values
- CSS Escape And Strictly Validate Before Inserting Untrusted Data into HTML Style Property Values.

- URL Escape Before Inserting Untrusted Data into HTML URL Parameter Values
- Sanitize HTML Markup with a Library Designed for the Job
- Prevent DOM-based XSS
- Use HTTPOnly cookie flag
- Implement Content Security Policy
- Use an Auto-Escaping Template System
- Use the X-XSS-Protection Response Header

Web server configuration: we are providing important steps for setting this parameter based on what the web server is using to set the security policies: Note: these steps may change slightly when any of the frameworks updates. Also, the web server administrator can add their own customized CSP rules based on their need. The selected web servers are the top three that are being used globally according to [16].

Apache:

Access the file httpd.conf as root user and add the following rule setting:

Header set Content-Security-Policy ”default-src ’self’;”

IIS:

Add the following to the HTTP response Headers for any specific site:

Name: Content-Security-Policy

Value: default-src ’self’

Nginx:

In the nginx.conf file, add:

add_header Content-Security-Policy ”default-src ’self’;”;

Second, Defense-In-Depth, User-Agent, and Content Injection: The concept of Defense-In-Depth is to build multiple layers security barriers to increase the protection level for the whole system. This principle came from the fact that the totally secure system is a system that has zero functionality (totally disconnected and powered off) [28] Information Security, Managing Risk with Defense in Depth. Therefore, an acceptable security level for a system at several stages is the best solution to provide diverse protection framework that provides integrity, confidentiality, and accessibility of the system and the data. As mentioned before, the whole idea of the CSP is to identify the safe and right source(s) that can be used in any web traffic session. That means the source identification and the infection detection can be effectively applied to the element, file, and directory level that is needed to run a web application file through the session. Based on that, we can say that CSP authentication and detection scopes are element, file, and directory. So it can tell if any of those is coming from the web server (or any of its aliases) or not, but simply it cants tells if the web element is infected or not. Based on this fact, and to obtain the best results, the web developer should avoid using inline scripts and styles as the CSP cant identify the infection at this level.

If the web content does not contain such vulnerable structure, then that maximum utilization of the CSP can be achieved and used by setting the needed policies by the web server and include that information in the response HTTP header to inform the user agent of the client’s browsers to use the safe and actual contents during the active sessions and avoid

using any injected and/or fake contents [33]. The user agent is responsible for specifying what files and elements from the web server will be called and used to respond to the client's request, based on the client's browser and device.

Succeeding in applying such protection technique would lead to create a very effective networking protection layer that is deep to the networking layer level which is applied even before the malicious content can be involved in the HTTP protocol session rather than detecting it by higher level protection tools when the malicious content is already transferred over the network, buffered, and defragmented at the clients memory. This protection layer is considered as one of the Defense-In-Depth effusive methods, as it presents an effective protective layer on the networking protocol level.

4.7 ODP Specific Web Vulnerabilities

Based on the functionality and the construction of the OPD's we selected the applicable web vulnerabilities that might affect the security of the ODP.

- Injection: Injection vulnerabilities, such as SQL, NoSQL, OS, and LDAP injections, occur when untrusted or malicious queries (also called smart queries), such as user input to some web based application, is sent to the database engine interpreter.Attacker's can access confidential or sensitive data by tricking the database interpreter to execute these inputs as unintended commands. Simply, the database engine interpreter will understand these inputs as commands instead of just data to be saves in the database.

- Sensitive Data Exposure: Web applications forms must properly protect confidential information, such as financial, healthcare, and PII. Gainign access to such insufficiently secured information is a mean reason for credit card fraud, identity theft, etc.

- Broken Authentication and Session Management: Several sensitive services and information that require solid authentication methods. The common cased where the authentication credentials can be compromised are when sending the credentials over weakly or not encrypted connection, or by accessing the access authentication session data which is the cookies.

- Cross-Site Scripting (XSS): This case will occur when a user request some content from the web and an attacker inject or replace some malicious content within the response. The user will receive this malicious content and compile it by his or her browser causing a security issue. This can happen when the user does not validate the response content for his or her request.

- Using Components with Known Vulnerabilities: Attacker job will be much easier when they target a well known server. In general, knowing what components the target is using such as libraries, frameworks, and so on will let the attacker knowing all these components vulnerabilities and default values and settings. For example, if a host is running a FTP server, then we can easily gues that port number 21 is open and can interact with any incoming FTP connection on that port.

- Underprotected APIs: Application programming interface (API) is considered a developer level connection method in which the client sends specific requests to fetch data from the ODP database. Many developers allow the client to request from the application or the website without any restrictions. As the ODPs contain a huge amount of data saved as a big size of files, so it is possible for a request can retrieve a big file of several Gigabytes. If many clients launched such requests at the same time then it can perform a DDoS attack keeping the server busy for a period of time trying to process these giant requests. In some cases, the client would be able to send information to the API server if it is not protected with an efficient authentication method. Our recommendations to avoid such cases by limiting the request size and how many times that a client can request based on the MAC address or the IP. Another method to apply an efficient authentication method is by asking the clients to create an account with sufficient identification information and showing the purpose of accessing the data and provide the client after approving his request with a unique identifier that must be included and verified in each API request the client does.

4.8 ODP Security Comparison For 10 Cities

Here, the most important and variant web traffic security factors have been compared. We included in our comparison the analysis for the following areas:

- XSS injection protection and Defense-In-Depth with Content Security Policy
- Broken Authentication and Session Management with Cookies Security and privacy

and X-XSS-Protection

- Cross-origin Resource Sharing
- HTTP Public Key Pinning
- HTTP Strict Transport Security
- Redirection
- Referrer Policy
- Subresource Integrity
- X-Content-Type-Options
- X-Frame-Options

Therefore, our multi-method results for all the 10 cities is concluded in Table 2

Security index formulation and calculation: The evaluation values are out of 100 for each city, and each one is a combination for measuring eleven different web security metrics. The security level score is computed as below.

$$m = 10 \times (b - w)/b \tag{4.1}$$

$$I = \sum_{i \in M} \frac{m_i}{|M|} \tag{4.2}$$

Where M is the set of security factors and m_i is a normalized individual security factor. M includes Content Security Policy, Cookies, Cross-origin Resource Sharing, HTTP Public Key Pinning, HTTP Strict Transport Security, Redirection, Referrer Policy, Sub-resource Integrity, X-Content-Type-Options, X-Frame-Options, and X-XSS-Protection. The individual security factors are normalized using the values of worst and best security configuration values. w is the worst security configuration value, and b is the best security configuration value. When the security metric is set to the basic default setting without considering security, we consider this as a very potential security vulnerability as any security setup should be confidential to avoid the common security issue of Using Components with Known Vulnerabilities. The worst configuration case is representing a big negative impact as it can cause functionality issue due to incompatibility, and security risk. The sum of normalized individual security levels for all factors considered is the score of security setting of an ODP. (See Equation4.2). We found that ODPs vary significantly in their security settings. Due to the space limitation, we only show the three prominent cases of ODPs among the cities we considered. The security scores of Chicago, Dubai and Kansas City are shown in Figures 7, 8 and 9, respectively.

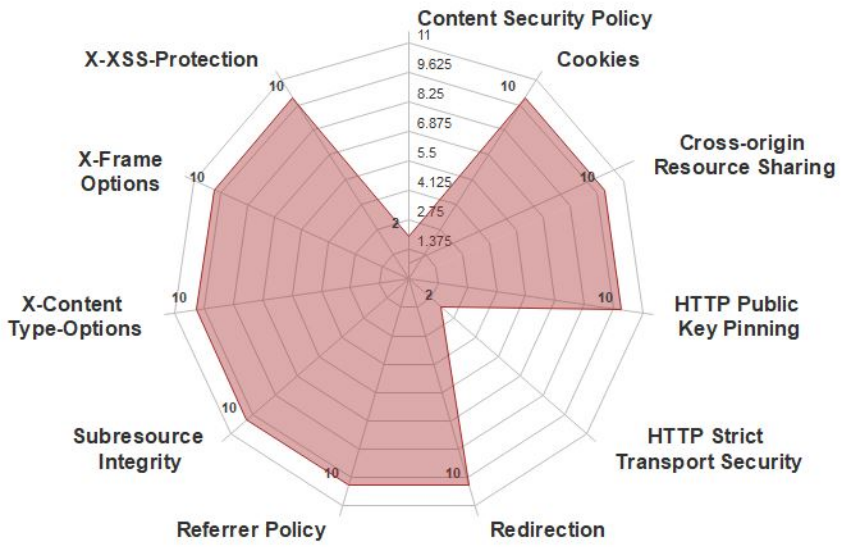


Figure 7: ODP security metric scores for Chicago

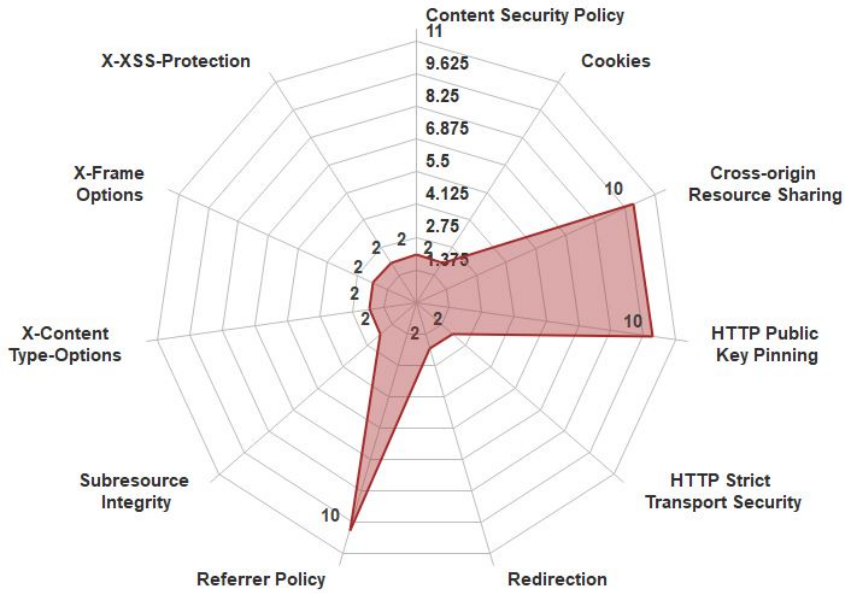


Figure 8: ODP security metric scores for Dubai

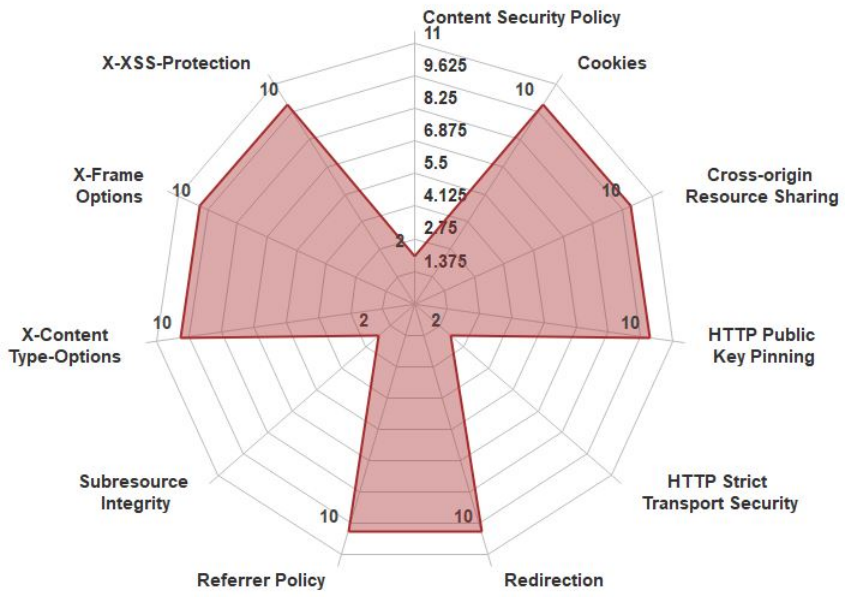


Figure 9: ODP security metric scores for Kansas City

Table 2: ODP's privacy and security comparison

City Name	Security Evaluation	Network Security	Data Security	Privacy Compliance
Chicago	82	4	4	8
Kansas City	73	2	3	5
Los Angeles	82	3	3	6
New York	54	2	2	4
Washington DC	63	2	2	4
Casablanca	46	2	2	4
Barcelona	36	2	2	4
Dubai	27	2	2	4
Seoul	32	3	2	5
London	82	3	3	6

CHAPTER 5

SMART CITY OPEN DATA (SCOD) OPENNESS

SCOD openness refers to how a city is open to its citizens and the entire world by providing accurate, reliable, periodic information through its ODP. This openness is a significant indication of whether a city is well connected, utilizing latest technologies, transparent, and interested in research and development. In [6] a representation of a method to measure the openness of any open data portal based on the availability of the required supporting data. [18] (The Open Data Barometer) is a web application that ranks the countries openness based on the overall open data available online. When a city provides its records to its ODP that allows many great impact parties such as the analysts, technology vendors, and local or foreign citizens to see it clearly. Analysts such as academic researchers can use this actual city records to analyze it and discover the city problems and provide feasible solutions for them. Technology vendors will be more confident to invest and provide solutions that are fitting the city needs by analyzing its open data. A City can be more transparent to its citizens by showing them what data the city is collecting. Citizens from outside the city or who are not familiar with the city can find the piece of the information that they need to make important decisions in their life, such as finding the proper neighbourhood to move in based on several metrics such as the crime rate, the education level of its educational institutes, and the abandoned buildings. Further, citizens will be aware of the major and frequently occurring issues in any specific

neighborhood.

Analysis tools and technique: A very valuable information piece of any website is the traffic records, such as the number of visits per a specific period of time. This information is crucial to measure the accessibility and find out the size of the expected traffic. The value of these records is going higher when it going older, so it can show a larger range of the portal traffic history, which is an essential detail that can be used to evaluate and determine several important the portal performance and activity, such as;

- The traffic (i.e. visits, data download requests, visits duration) size and if it has any seasonal trend so it would be caused by some incident, a development, or change that caused the increase or decrease

- What is the most important information that the ODP visitors are interested in so it can be updated constantly and more precisely. For instance, we found that for Kansas City the top five keywords that were the Kansas City ODP looking for are; KC water, Kansas City water, KCMO water (KCMO refers to Kansas City Missouri as Kansas City has both Missouri and Kansas states sides), Kansas water services, and Kansas City parcel viewer. Figure 10 shows the number of the for all the ten ODPs during the last 30 days. From this figure we can say that the cities of a higher population (Chicago and Los Angeles) are scoring high numbers of visits, however, Washington DC and Seattle are scoring the highest. by looking closer, Washington DC is attracting very high attention as it is the capital of the United States, also Seattle is scoring high because of their wide initiatives and activities towards the smart city and open data [4]. Another interesting measure is shown in Figure 11

which shows more precise metric that is how many pages each visitor is visiting. This metric can indicate how visitors are engaged to a certain ODP. Figure 11 also gives a time trend view for the ten cities and we can say that there is no clear one. However, we can see a clear peak for London City during the second week of November when the 3ie London Evidence Week taking place which is several public activities to focus about the city challenges based on the actual evidence [5].

- Who is visiting the ODP (in terms of the gender, education, and age). An interesting indicator is the age which is shown in Figure 12 as it divides the ODPs visitors age (from 18 to 65+) into six groups. In the academic world, we can classify that, 18 to 24 are the undergraduate students, 25 to 44 are the graduate students, and 45 to 64 are the scholars such as professors and Ph.D. candidates. Based on this classification, it is clear that the 25 to 44 group is dominating the ODP visits records. This can be strongly justified by assuming that the graduate students are representing a major group who targets the open data topic and doing research on it.

- How a certain ODP is popular by measuring its reputation of what websites are directing to an ODP such as a search engine, somewhat social media website, or a link from a well-known website. Figure 13 shows the percentage of the traffic source for the ODPs visitors which means that the previous page was visited by the visitor prior to visiting the ODP page. It is clear that the social networks are the smallest source for the ODP traffic, however, the search engines are the larger traffic source.

We have used a number of web analytics tools from Google Analytic, Amazon Web Analysis, and Similarweb. It is important to mention that the graph that been generated here are using the web access records for the visits counting, and using the average number of internet users based on the geographic area such as the zip code.

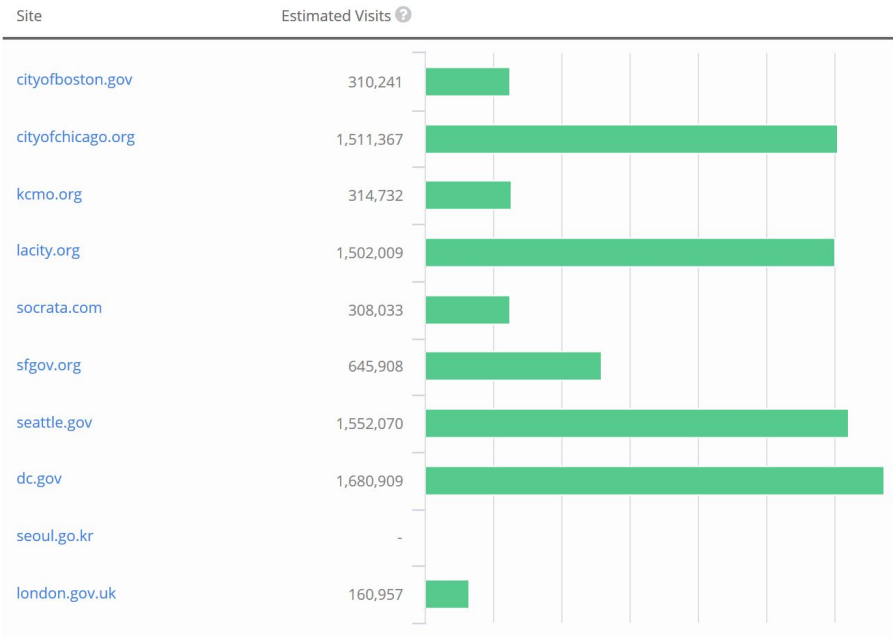


Figure 10: Number of estimated visits for the last 30 days

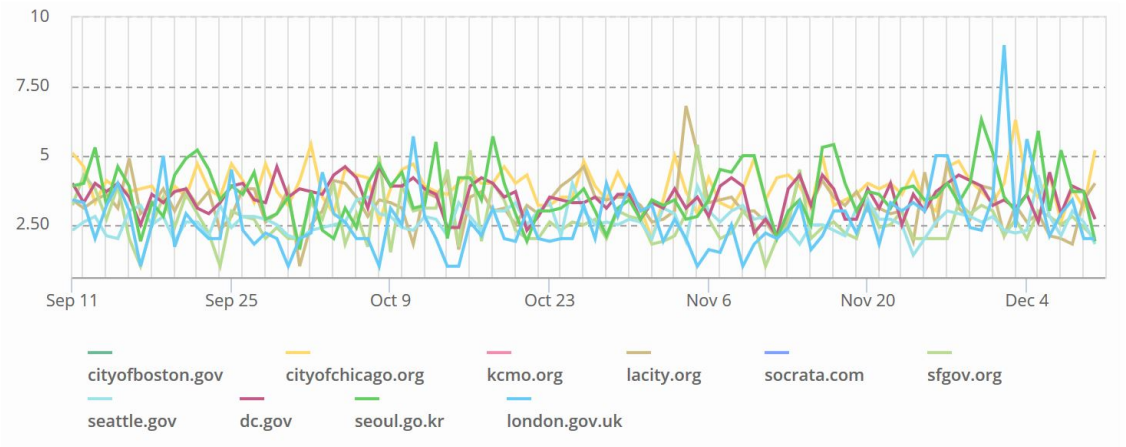


Figure 11: Number of page views per user for the last three months

Figure 12: ODPs visitors average age for the last three months

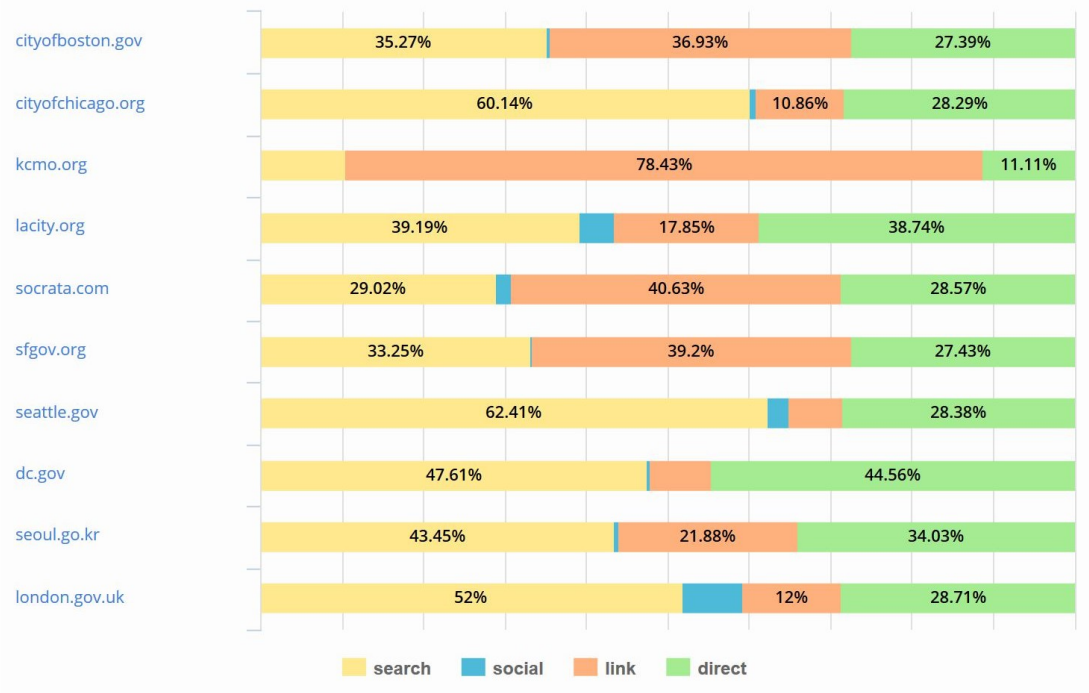


Figure 13: ODPs traffic sources for the last three months

5.1 Open Data Platforms and Data Formats

The OPDs analyzed in our study are using third-party cloud SaaS services that are Socrata and CKAN. Each is providing a set of features for the ODP administrators to provide different features to the ODP users. We found that Socrata is the most used platform among these ODPs. Boston is using most of Socrata's platform features such as the different illustration methods such as tables, maps, graphs and almost all the common used big data formats, such as CSV, Excel, JSON, RDF, RSS, XML, JSON API connection. However Pittsburgh is using CKAN platform and they are hosting and managing their data in University of Pittsburgh campus, so it is one of if the uncommon ODPs cases where the cities are using others clouds infrastructure [21]. Other open data portal platforms are

used by many other ODPs such as DKAN, Junar, OpenDataSoft, etc. In fact, all the mentioned ODP platforms and others are using Amazon AWS IaaS and IaaS cloud services [1]. These platforms are providing very detailed and well-organized documentation for their platforms for web and application developers.

5.2 Number of Datasets

The datasets refer to data contents and their classification available in ODPs. Each dataset contains open data related to any particular categories such as traffic, housing, health and 311 calls. The number of datasets of each ODPs are summarized in Figure 14.

5.3 Population Vs Number of The Visits

It is very expected that the city of a high population will provide more data than the cities of lower populations. We found this is not true for two main reasons. First, some big cities donât provide a sufficient information grouping, so they release the data in very big general sets (very general classification). The second reason is that some big cities donât publish or utilize the data they are collecting in an efficient way, sometimes for privacy and security reasons and/or because they lack to use the efficient open data framework and portal. All the references for these data can be found in the extension report of our work. Our results are concluded in Figure 15.

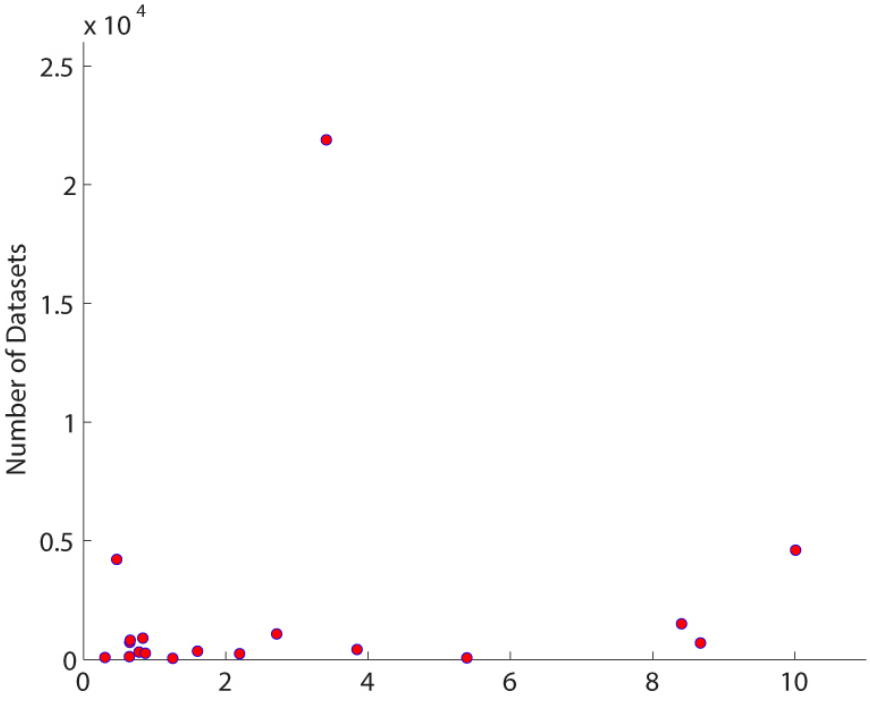


Figure 14: Number of datasets available of different cities

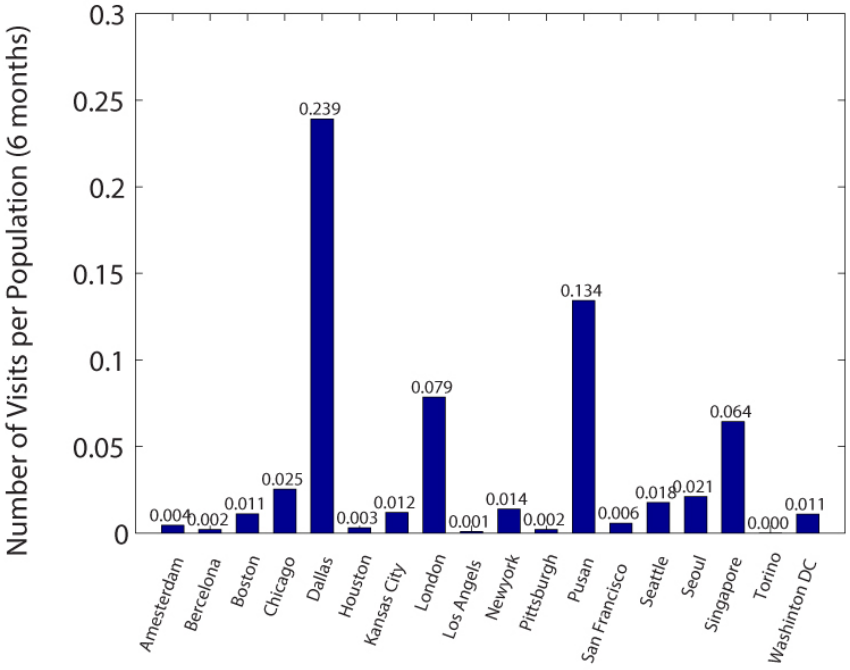


Figure 15: Number of ODP visits per city population

By knowing and understanding the ODP, it is very useful to design, customize, and set the performance metrics of the ODP so that it serves the highest percentage of their visitors. Such as adding tools that the visitor would be comfortable with (sophisticated or basic), using the right load balancing if a high volume of requests occurred. Enriching the most used datasets and keywords and give them a priority in updating and provide them with more customizable search, filtration, and illustration.

CHAPTER 6

PRIVACY ISSUE ANALYSIS

As the smart cities experience is being more mature and expand to be involved in more services, the number of the sensing points will increase, therefore the amount of the collected data increase with a variety of types. Sensors can be cameras, motion sensors, light sensors. Data can be gathered in other ways such as from the Wi-Fi servers, police reports, etc. Most of these data sources are containing private and/or sensitive information such as locations, addresses, pictures, network access information, etc. Therefore, citizens are concerned about who is controlling these sensors, how is the data is being stored and for how long, and many other questions. From that, it is important to find out if the smart city system is privacy compliant or not, and if not, then to what extent. The main and most common source of this data is the ODP, so we studied these cities to see how they are maintaining the data privacy.

6.1 Privacy Metrics Analysis

The metrics we tested include: Cookies, SSL, Encryption, Privacy Policy Disclosure. The grading criteria were '0' for not present or could not find the research methodology to '4' that is equivalent to the best standard found or moving in that direction. The finding from this table is cities with higher security has more privacy compliance. For example, Chicago has security evaluation 82 and privacy compliance 8, on the other hand, Dubai has security evaluation 27 and privacy compliance as 4. Our findings are

Figure 16: PII (complete addresses - blurred for privacy) in 311 call data from KC-ODP

summarized in Table 2.

6.2 PII Filtration Issue

An important privacy issue is sensitive data exposure such as Personal Identifiable Information (PII), as we find many incidents of PII (i.e. personal full address) exposed to the public on the ODP (Figure 16). The study proposing a real-time filtration tool to filter all such sensitive information based on the patterns and machine learning to operate within a specified identification (or detection) confidence level, other than the field or content matching [13].

We found the need for a specialized tool that can filter the private information in

an effective way in ODP. Though police crime reports are handled by the police department and most cases information are not transferred to open data portal, however, city government receives calls from citizens regarding neighborhood complaints, sometimes the records contain their private information online on the ODP. In an explanation for these mistakes, we found that person who is reporting, add their personal information in a field which is not supposed to be filtered. In Figure 16 shows in Kansas City ODP 311 data has full address information (for privacy reason addresses are blurred here). These situations do not have an effective solution rather than removing the private data manually, which would a failure for ODP. We developed a real-time filtration tool that is able to distinguish the PII information pattern rather than discovering it by the exact match. This tool is developed and tested using JSON files that contain name, email address etc. The tool can be found in Github [13].

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Summary

In this study, we have evaluated the current status of ODPs in multiple cities around the world. Many cities collect and publish data for transparency, attracting businesses and solving important issues such as traffic, security, and utility service quality. However, our study indicates that these cities do not have these high expected data records that should meet their sizes and need. Another concern is the ability to maintain the privacy and security of these data that keep on increasing over time. The security risk grows significantly as the smart services are using these data over different parties. Therefore, the challenge is to balance between the utility of smart services and the data security. Privacy protection is also one of the growing concern in ODP systems. We developed a tool to identify and filter PII from the raw data before it is published in the portal. While our tool identifies PII such as name, email address, phone number, further studies and developments are needed to identify other private information such as in images, locations, and networking logs through more sophisticated intelligent algorithms.

Constant penetration testing, network security (fast scanners for ports, encapsulation, protocols flags, congestion control, etc.) are effective security practices that can maintain a secure system. Further, a simple and trivial ends connected to a sophisticated and more intelligent core is a networking system design that is proofed to be efficient and

adopted by Software Defined Networks (SDN) systems to limit the sensitive controlling data within specific traffic area, which can be considered as sensitive data traffic isolation. A solution for identification and authentication security issue in the smart city environment is to implement a smart city security system that can adapt to the need of blockchain methodology, as its substantially lower-cost solutions can be instantiated, which can disrupt existing business models, offering a secure authentication transaction sessions.

7.2 Technical Enhancements and Educational institutions involvement

These big cities are expected to have the highest open data portal visits and data sets classification because the number of the studies and analysis should be higher to satisfy their need. However, our study indicates that these cities do not have these high expected records that should meet their sizes and need. In this direction, many suggestions can participate in better open data utilization such as apply multiple enhancements to the open data portal such as provide a higher level of search, sort, classification, and more user-friendly design. Another thing is by involving the educational researchers and institutions to make more analysis and research to solve any of the common issues of the big cities. In addition to this, we found that many of the open data portals do not keep records for their open data portals such as the visits numbers, visitors locations, and so on. this information is important in evaluating the portal performance and accessibility.

7.3 Multi-vendors and Unified Policies

Another point is that we found that is almost most of the US open data portals are using Socrata as database platform provider. Few among them are using either their own data solution or using other database platform providers, as shown in Table.1. The issue here is the data privacy and security especially when it comes to smart cities that have many different vendors are involved in that. In the most cases, the data ownership is not clear. Furthermore, it is almost impossible to find a unified privacy or security policies applied. Our other research is on the track to solve this issue in a collaboration of the school of computing and engineering and Law school of University of Missouri Kansas City which aims to provide a unified security and privacy gauging tool for any smart city open data portal.

The open data field is a huge promising field and many research is going in many of its directions. Mainly how to unleash the powerful information in this data and how to utilize it. The other direction is the expected coming issue is how to maintain the privacy and security of this data as it is in a rapid increase and its value is increasing with time. The security risk is increasing as the smart services that are using this data is increasing. Therefore, a very rich direction is how to deal with the tradeoffs between the smart services and the securing them.

Appendices

APPENDIX A

SECURITY COMPARISON RANKING DATA FOR ALL CITIES

Ten smart cities ODP's security evaluation The OPD's security evaluation parameters are provided in detail in the following table. Further, an explanation for the evaluation for each parameter is discussed too.

Table 3: Ten smart cities ODP's security evaluation

City	CSP	Cookies	CORS	HTTP PKP	STS	Redir	Pref Pol	SI	XCTO	XFO	X-XSS-Prot
Chicago	0	10	10	10	0	10	10	10	10	10	10
Kansas City	0	10	10	10	0	0	10	10	0	0	0
Los Angeles	0	10	10	10	0	10	10	0	10	10	10
New York	0	10	10	10	0	10	10	10	10	10	10
Washington DC	0	10	10	10	10	10	10	10	10	10	10
Casablanca	0	10	10	10	10	10	10	10	10	10	10
Bercelona	0	10	10	10	10	10	10	10	10	10	10
Dubai	0	0	10	10	10	10	10	10	10	10	10
Seoul	0	5	10	10	10	10	10	10	10	10	10
London	0	10	10	10	10	10	10	10	10	10	10

APPENDIX B

SECURITY COMPARISON FIGURES FOR ALL CITIES

All the cities comparison figures are provided in this appendix. It is important here to mention that the spider figure is used for illustration purpose only and the area under the spider net does not refer to the security rank or the metric value. The most important thing in this illustration is the values of each metric that is represented by the ends of the spider metric. It also illustrates the variations of security metrics values among the cities.

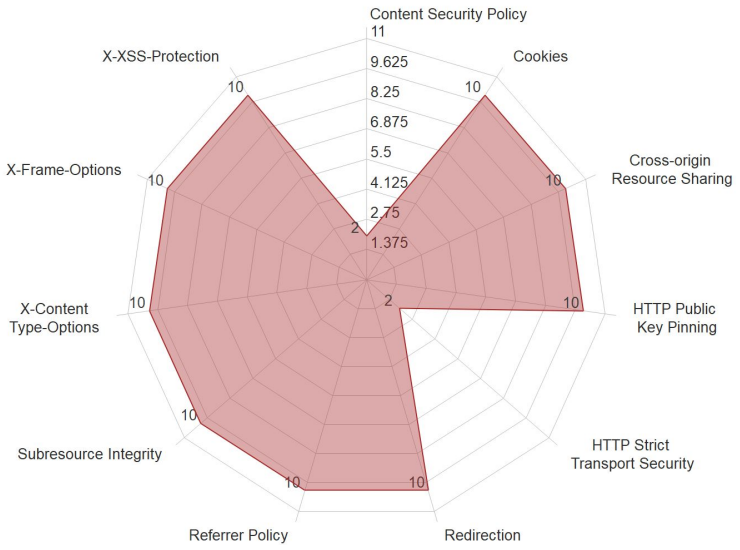


Figure 17: Los Angeles

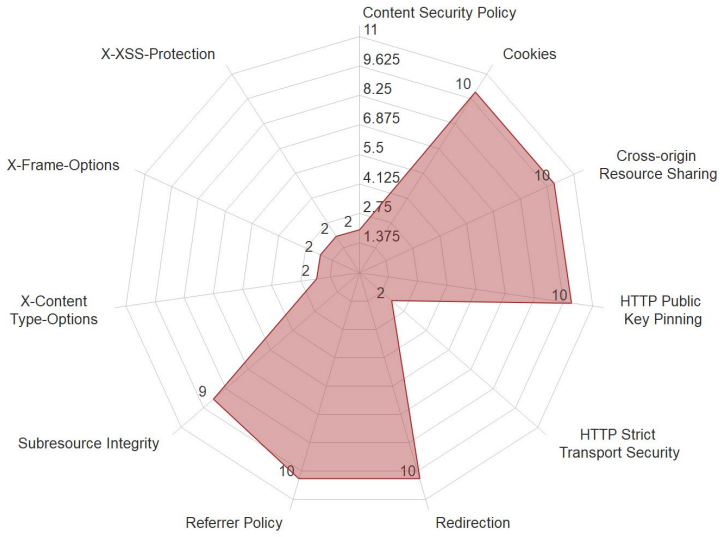


Figure 18: New York

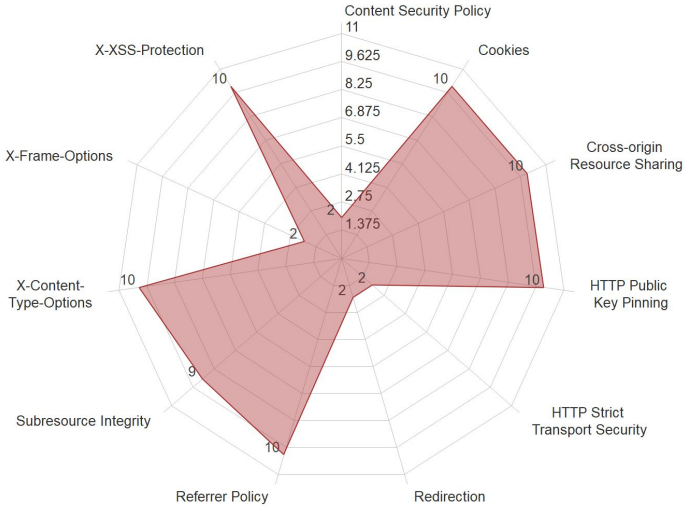


Figure 19: Washington DC

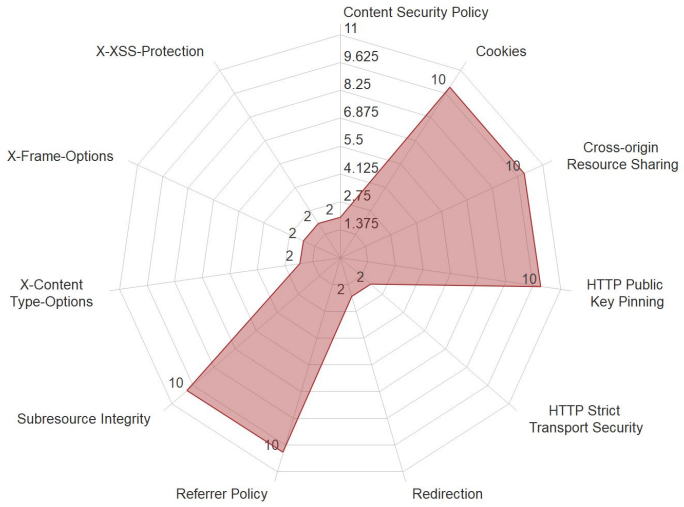


Figure 20: Casablanca

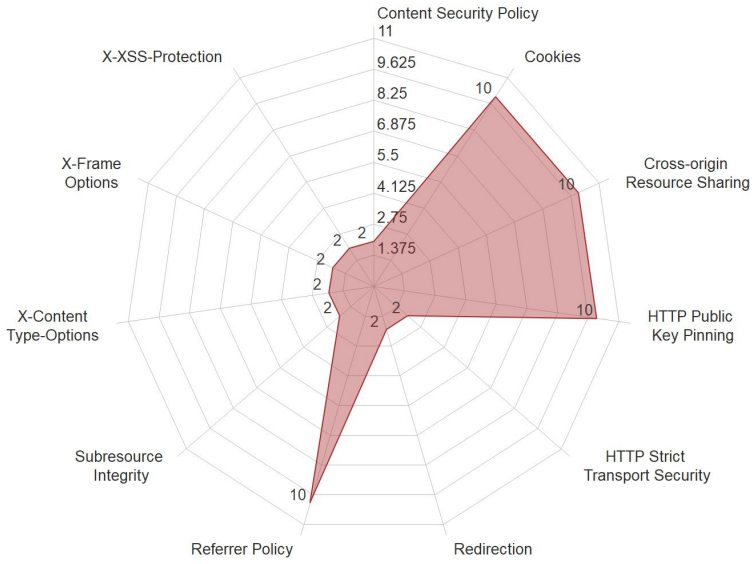


Figure 21: Barcelona

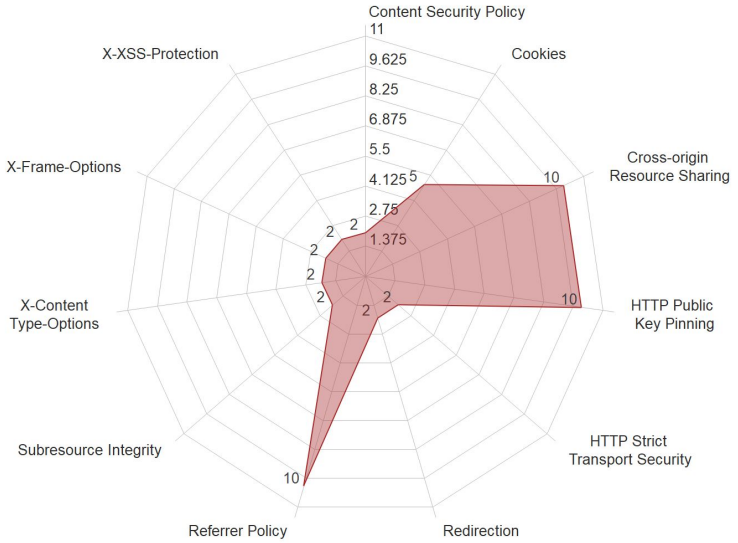


Figure 22: Seoul

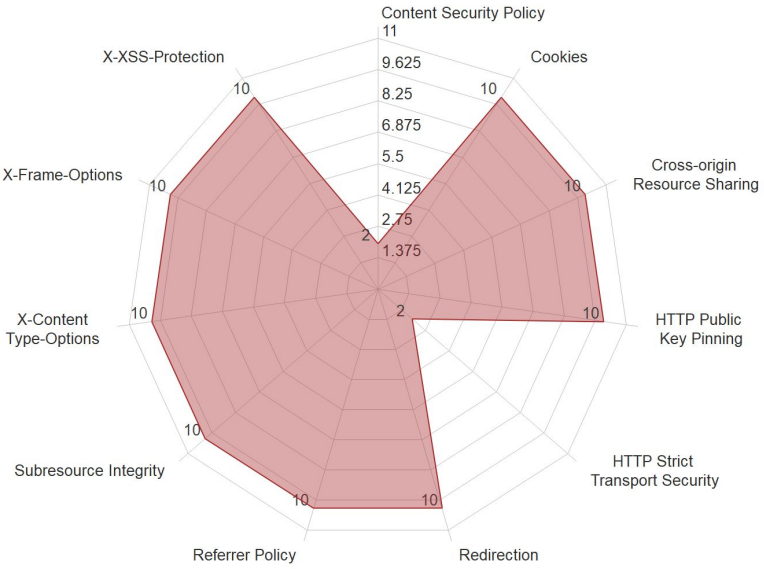


Figure 23: London

REFERENCE LIST

[1] Amazon-AWS. Cloud-based data democratization solutions for government. https://aws.amazon.com/government-education/open-data/. Accessed: 2017-11-08.

[2] Olayiwola Bello, Victor Akinwande, Oluwatoyosi Jolayemi, and Ahmed Ibrahim. Open data portals in africa: An analysis of open government data initiatives. *African Journal of Library, Archives and Information Science*, 26(2):97–106, 2016.

[3] Cesar Cerrudo, Mohamad Hasbini, and Brian Russell. Cloud security alliance: Cyber security guidelines for smart city technology adoption. 2015.

[4] Jenny Durkan. Smart, data-driven city. https://www.seattle.gov/tech/initiatives/smart-cities. Accessed: 2017-11-07.

[5] International Initiative for Impact Evaluation. London evidence week. http://www.3ieimpact.org/en/events/3ie-conferences-and-workshops/london-evidence-week-2017/. Accessed: 2017-12-07.

[6] Mark Fox and Christopher Pettit. On the completeness of open city data for measuring city indicators. In *2015 IEEE First International Smart Cities Conference (ISC2)*, pages 1–6, Oct 2015.

[7] Site Guarding. Real time pii detector and filter. https://www.siteguarding.com/. Accessed: 2017-11-23.

[8] Homer-Project. Homer project. http://www.homerproject.eu. Accessed: 2017-11-03.

[9] Web Inspector. Real time pii detector and filter. https://webinspector.com. Accessed: 2017-11-03.

[10] Heyseoung Jo and J Ramon Gil-Garcia. User acceptance of open data portals in korea: The role of policy tools. In *Proceedings of the 17th International Digital Government Research Conference on Digital Government Research*, pages 540–541. ACM, 2016.

[11] Erik Lakomaa and Jan Kallberg. Open data as a foundation for innovation: The enabling effect of free public sector information for entrepreneurs. *IEEE Access*, 1:558–563, 2013.

[12] Renata Máchová and Martin Lnénicka. Evaluating the quality of open data portals on the national level. *Journal of Theoretical and Applied Electronic Commerce Research*, 12(1):21–41, 2017.

[13] Mohammed Mansoori. Real time pii detector and filter. https://github.com/moemanson/RealTimePIIDetectorandFilter. Accessed: 2017-10-20.

[14] Adria Mercader. Ckan: Open source data portal platform. https://ckan.org/. Accessed: 2017-04-18.

[15] MetroLab. A city-university collaborative for urban innovation. https://metrolabnetwork.org. Accessed: 2017-09-05.

[16] NetCraft. Internet security and data mining. https://www.netcraft.com. Accessed: 2017-08-19.

[17] Sebastian Neumaier, Jürgen Umbrich, and Axel Polleres. Automated quality assessment of metadata across open data portals. *Journal of Data and Information Quality (JDIQ)*, 8(1):2, 2016.

[18] Open-Data-Barometer. World wide web foundation: Open data barometer global report. http://opendatabarometer.org/4thedition/country-sheets. Accessed: 2017-11-06.

[19] Open-Data-Soft. Open data soft. https://www.opendatasoft.com. Accessed: 2017-06-14.

[20] OWASP. Open web application security project. https://webinspector.com. Accessed: 2017-10-02.

[21] Pittsburgh-Open-Data-Portal. Cloud-based data democratization solutions for government. http://www.wprdc.org/data-wizards/. Accessed: 2017-11-01.

[22] Open Data Platform. Dkan open data platform. https://getdkan.org/. Accessed: 2017-04-25.

[23] Open Data Platform. Junar: Open data platfom. http://www.junar.com/index9ed2.html?lang=en. Accessed: 2017-04-02.

[24] PROGNOZ-PLATFORM. Prognoz platform. http://www.prognoz.com. Accessed: 2017-10-22.

[25] Quttera. Malware detection, blacklisting check, site clean-up services. https://www.quttera.com. Accessed: 2017-11-17.

[26] Shawn Ralko and Sathish Kumar. Smart city security. *KSU Conference on Cybersecurity Education, Research and Practice*, 2016.

[27] Alejandro Sáez Martín, Arturo Haro De Rosario, and María Del Carmen Caba Pérez. An international analysis of the quality of open government data portals. *Social Science Computer Review*, 34(3):298–311, 2016.

[28] SANS-Information-Security. Managing risk with defense in depth]. https://www.sans.org. Accessed: 2017-09-15.

[29] Socrata-Documentation. Socrata: Build something awesome with open data. https://dev.socrata.com/. Accessed: 2017-07-03.

[30] SSL-Labs. Ssl labs: Ssl web server on the public internet. www.ssllabs.com. Accessed: 2017-10-04.

[31] SUCURI. Website malware and security scanner. sucuri.net. Accessed: 2017-10-04.

[32] Jeffrey Thorsby, Genie NL Stowers, Kristen Wolslegel, and Ellie Tumbuan. Understanding the content and features of open data portals in american cities. *Government Information Quarterly*, 34(1):53–61, 2017.

[33] Mike West, Adam Barth, and Dan Veditz. Content security policy level 2. https://www.w3.org/TR/CSP2/. Accessed: 2017-10-22.

[34] Jan Whittington, Ryan Calo, Mike Simon, and Jesse Woo. Push, pull, and spill: A transdisciplinary case study in municipal open government. *Berkeley Technology Law Journal*, 30(2):1967, 2016.

VITA

Mohammed Almansoori was born in Iraq in 1984. He graduated from Graryounis University in 2008 as Telecommunication Engineer with honor and first rank. After graduation, he worked in University of Kufa in web development and network architecture. Also he worked in Aljazeera Telecom in ISP networking. In 2014, he received full scholarship for graduate study in University of Missouri Kansas City. He started his Masters in Telecommunication and Computer Networking in 2015. He worked in University of Missouri Kansas City as Research Assistance and Teacher Assistance in Electrical Engineering and Computing department, and Network Architect in the networking department of Information Services.

His research area was in smart city, open data, information and networks security and privacy, internet of things, web development, and multimedia solutions such as virtual reality 360.