

# **Final Project Report: SOS Threat Analytics**

**Submission Due Date: 10/05/2017**

**Prepared by:  
Stuart O'Shaughnessy – x13117084 – Business Information Systems**

---

## Table of Contents

<b>1. Introduction .....</b>	<b>5</b>
1.1 Background and Context .....	5
1.2 Project Concept.....	5
1.3 Project Scope & Deliverables .....	6
<b>2. Definitions, Acronyms &amp; Abbreviations .....</b>	<b>7</b>
<b>3. Technologies .....</b>	<b>10</b>
3.1 Paessler NetFlow Generator .....	10
3.2 Eclipse IDE .....	11
3.3 Apache Spark .....	11
3.4 SQL Server.....	11
3.5 ASP.net MVC .....	11
3.6 Marketing Website.....	11
3.6.1. Web Hosting .....	11
3.6.2. Search Engine Optimisation .....	12
<b>4. Cisco NetFlow.....</b>	<b>12</b>
<b>5. Requirements Elicitation .....</b>	<b>15</b>
<b>6. Basic User Requirements Definition .....</b>	<b>15</b>
<b>7. Requirements Specification.....</b>	<b>16</b>
7.1 Functional Requirements – Application .....	16
7.1.1. User Registration .....	17
7.1.2. User Sign-In .....	18
7.1.3. View User Profile .....	19
7.1.4. Update User Profile .....	20
7.1.5. Application Navigation.....	20
7.1.6. Access Dashboard.....	21
7.1.7. View IOC Graphic .....	21
7.1.8. View Twitter Feed .....	22
7.1.9. View Security Based RSS Feeds.....	23
7.1.10. Access the Indicators of Compromise Page .....	23
7.1.11. Access Detailed IOC Information.....	24
7.1.12. Access a list of monitored Devices .....	25
7.1.13. Access Device Details.....	26
7.1.14. Access Vendor & Client Contacts.....	26
7.1.15. Adding/Deleting/Amending a Contact .....	27
7.1.16. Log a Support Request .....	28
7.1.17. Print Pages .....	29
7.1.18. Alerting & Notifications .....	29
7.2 Functional Requirements – System.....	30
7.2.1. NetFlow Generation.....	30

- 7.2.2. Apache Spark Application ..... 31
- 7.2.3. Front-End Application ..... 31
- 7.3 Functional Requirements – Data ..... 32
  - 7.3.1. SQL Server – Relational Database ..... 32
  - 7.3.2. Relational Database Structure ..... 33
- 7.4 Functional Requirements – Marketing Website ..... 33
  - 7.4.1. Site Hosting ..... 33
  - 7.4.2. Site Connectivity / Accessibility ..... 33
- 7.5 Non-Functional Requirements ..... 34
  - 7.5.1. Performance/Response Time Required ..... 34
  - 7.5.2. Availability Requirement ..... 34
  - 7.5.3. Accessibility Requirement ..... 34
  - 7.5.4. Security Requirement ..... 35
  - 7.5.5. Maintainability Requirement ..... 35
  - 7.5.6. Portability Requirement ..... 35
  - 7.5.7. Reusability Requirement ..... 35
  - 7.5.8. Error Checking Requirement ..... 35
  - 7.5.9. Concurrency Requirement ..... 35
- 8. Solution Topology & Architecture ..... 36**
- 9. Project Implementation ..... 37**
  - 9.1 Programming Language Environment ..... 37
  - 9.2 Database ..... 45
    - 9.2.1. Database ERD ..... 46
    - 9.2.2. Front-End Application Integration ..... 48
  - 9.3 SOS Threat Analytics Application ..... 49
    - 9.3.1. Twitter & RSS Feeds ..... 50
    - 9.3.2. Display Threat Details ..... 51
    - 9.3.3. Print Details ..... 51
  - 9.4 SMS Alerting ..... 52
- 10. Testing & Evaluation ..... 55**
  - 10.1 Testing Objectives ..... 55
  - 10.2 Testing Scope ..... 55
  - 10.3 Testing Strategy ..... 55
  - 10.4 Testing & Evaluation Results Tables ..... 55
  - 10.5 Testing Results and Conclusion ..... 63
- 11. Graphical User Interface & Usability ..... 63**
  - 11.1 Splash/Home Screen ..... 63
  - 11.2 Navigation Bar - Logged in User ..... 64
  - 11.3 Dashboard ..... 64
  - 11.4 Indicators of Compromise ..... 65
  - 11.5 Device Listing ..... 67
  - 11.6 Vendor & Client Contacts ..... 67

<b>12. System Evolution.....</b>	<b>69</b>
<b>13. Target Market.....</b>	<b>71</b>
13.1 Market Rivals.....	71
<b>14. References .....</b>	<b>72</b>
<b>15. Appendices.....</b>	<b>73</b>
15.1 Journal Entries.....	74
15.1.1. September.....	74
15.1.2. October.....	76
15.1.3. November.....	78
15.1.4. December.....	79
15.1.5. January.....	81
15.1.6. February.....	82
15.1.7. March.....	83
15.2 Project Plan.....	87
15.3 Requirements Elicitation – Questionnaires.....	88

## 1. Introduction

The purpose of this document is to provide an in-depth overview and detailed description of the design and implementation of my final year project: SOS Threat Analytics. This document will give an extensive description of the project, its high-level deliverables and the associated methodologies, technologies and techniques that were used to achieve them. Furthermore, it will deliver a comprehensive explanation of the specific requirements that were researched and evaluated, to ultimately formulate the core functionality and usability of the final product. Outlined within will be the principal purpose of the application, a detailed breakdown of all functional and non-functional elements, an in-depth look at its implementation as well as visual representations of the GUI; illuminating the key end-user functionality. Ultimately, this document is intended as a structured guideline and overview of all aspects of my final year project.

### 1.1 Background and Context

I am currently employed as an 'IT Service Delivery Manager' within the financial services sector. Due to the regulated nature of the industry, security audits and external validation are a regular occurrence and an inherent requirement. There is an increased focus on consistently evaluating the integrity of a firm's infrastructure and their associated technologies. Cyber threats such as Malware and Ransomware attacks can threaten a firm's capability to conduct business and in some extreme cases their continuity and very existence. As such a constant evolution of defences and security measures is required.

One area that many organizations fail to sufficiently evaluate is their underlying internal network activity. There is in my opinion, an under appreciation for the level of knowledge and protection that a strong understanding of this activity can deliver. Many organizations are now very cost focussed and there are of course significant costs to implementing network monitoring solutions and/or all-encompassing security systems (Managed services, SIEM etc.) I believe however, that there is the potential for a lightweight solution that can be seamlessly configured on an organizations network environment and that can deliver the requisite information and knowledge about what is going on under the hood. Hence my choice of final year project was a security analytics tool with a core goal of adding value and protection to an organization through the delivery of pro-active alerting in the event of a potential network threat.

### 1.2 Project Concept

SOS Threat Analytics is focused in the realm of Information Security and perhaps more specifically around the concept of security analytics. Data analytics – in the more general sense – is a powerful tool that is currently utilised across almost all major industries e.g. financial, commercial, technological. Its purpose is as a method of ingesting, filtering and interpreting large amounts of data in-order to better understand and glean information from consumers, systems, activities etc. This project deals with the concept of extracting and interpreting network specific information to give users a clear understanding of what is occurring on their network. The focus is on Cisco networks (due to their market prevalence) and their proprietary network protocol – Cisco NetFlow. Crucially, the final application delivers

a modern user interface as well as pro-active alerting to notify users of spurious network activity and potential (cyber) security threats.

This project concept was borne out of a deep-rooted interest in Information Security and a desire to learn more about data analytics. Furthermore, the structure and technologies that were required for its successful implementation more than ensured that I was suitably challenged. This project required the use and continual development of skills that I had acquired through college as well as education in technologies in which I had no prior experience. Security is one of the most important topics in technology (and business) today and therefore I believe that this project is industry relevant, applicable, and highly topical.

### 1.3 Project Scope & Deliverables

Below is a list of the key, high level deliverables that guided the initial design and focused the successful implementation, of all aspects in the SOS Threat Analytics project. The methodologies and technologies required to complete the below, will be further detailed within this document:

- A device (or Software solution) to actively generate NetFlow information
- A method of ingesting large amounts of data from network devices/simulators
- A method of storing and filtering the ingested data
- A database structure to hold relevant/filtered data
- A method of interrogating the data in an efficient manner
- A front-end web application to display relevant data to the end user
- An alerting system to communicate with the end user
- A marketing website to display information and sales data about the application
- Structured documentation & deliverables at all phases of the project

## 2. Definitions, Acronyms & Abbreviations

The below explains some of the common terms and language that will be used throughout the document and gives some added context to the details provided.

### ➤ Apache Spark

"Apache Spark is an open-source engine developed specifically for handling large-scale data processing and analytics. Spark offers the ability to access data in a variety of sources, including Hadoop Distributed File System (HDFS), OpenStack Swift, Amazon S3 and Cassandra." (Webopedia, 2017). Within SOS Threat Analytics, Apache Spark was utilised to stream and filter NetFlow data received from the NetFlow generator. 'Spark is considered a fast engine for big data processing and has inbuilt modules for SQL, data streaming etc.

### ➤ Amazon Simple Storage Service (S3)

Amazon S3 is "a scalable, high-speed, low-cost, web-based cloud storage service designed for online backup and archiving of data and application programs." (SearchAWS, 2016). Storage is S3 is very popular within S3 and as such this platform was chosen to host the marketing website ([s3-eu-west-1.amazonaws.com/www.sosthreatanalytics.com/Index.html](https://s3-eu-west-1.amazonaws.com/www.sosthreatanalytics.com/Index.html) - sosthreatanalytics.com).

### ➤ ASP.net MVC

"The ASP.NET MVC is a web application framework developed by Microsoft, which implements the model-view-controller (MVC) pattern." (Wikipedia, 2017). The front-end web application for SOS Threat Analytics was developed entirely using this framework.

### ➤ Bootstrap

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. (BootstrapDocs, 2017). The design of both the application front-end and marketing website were completed in Bootstrap.

### ➤ C#

C# (pronounced "C-sharp") is an object-oriented programming language from Microsoft that aims to combine the computing power of C++ with the programming ease of Visual Basic. C# is based on C++ and contains features similar to those of Java. (SearchWinDevelopment, 2007). Used in the development of the SOS Threat Analytics front-end web application within the ASP.net MVC Framework.

### ➤ Functional Requirements

The official definition of 'a functional requirement' is that it essentially specifies something the system should do. Typically, functional requirements will specify a behavior or function, for example: "Display the name, total size, available space and format of a flash drive connected to the USB port." Other examples are "add customer" and "print invoice". (Rqtest,2012). Sections 7 explains the requirements used to ensure that the basic functions of the system are met. The requirements refer to the basic functionality that underpins the system/user interaction and engagement.

### ➤ GUI

*A graphical user interface (GUI) is a human-computer interface (i.e., a way for humans to interact with computers) that uses windows, icons and menus and which can be manipulated by a mouse (and often to a limited extent by a keyboard as well). (linfo.org, 2004). In the context of this project the GUI was built using the ASP.NET MVC framework.*

### ➤ Indicators of Compromise (IOCs)

*Indicators of compromise (IOCs) are "pieces of forensic data, such as data found in system log entries or files, that identify potentially malicious activity on a system or network." Indicators of compromise aid information security and IT professionals in detecting data breaches, malware infections, or other threat activity. (DigitalGuardian, 2016). SOS Threat Analytics displays all potential IOCs to the end user via the GUI and allows them to access detailed information about the potential threat.*

### ➤ Java

*The Java™ Programming Language is a general-purpose, concurrent, strongly typed, class-based object-oriented language. It is normally compiled to the bytecode instruction set and binary format defined in the Java Virtual Machine Specification. (Oracle, 2014). Java was used in the development of the Apache Spark Streaming application. This enabled ingestion of the core data and integration within the SOS Threat Analytics database environment.*

### ➤ JavaScript

*JavaScript is an interpreted programming or script language from Netscape. It is somewhat similar in capability to Microsoft's Visual Basic, Sun's Tcl, the UNIX-derived Perl, and IBM's REXX. (Search Microservices, 2014). JavaScript was used to enable functionality on both the front-end application and marketing website.*

### ➤ NetFlow

Proprietary network protocol that was developed by Cisco to allow for the collection of IP traffic information. This is the primary data source and core element of this project. NetFlow data is streamed, transformed, stored and analysed to produce meaningful results about the underlying network activity. See Section 4 of this document.

### ➤ Non-Functional Requirements

*The definition for a non-functional requirement is that it essentially specifies how the system should behave and that it is a constraint upon the systems behaviour. One could also think of non-functional requirements as quality attributes for a system. (Rqtest, 2012). In terms of the project, these focused on the overall performance and delivery of the end goal for the end-product.*



### ➤ SEO

SEO will be utilised to enhance the visibility of this projects marketing website. *SEO is short for search engine optimization. Search engine optimization is a methodology of strategies, techniques and tactics used to increase the number of visitors to a website by obtaining a high-ranking placement in the search results page of a search engine (SERP) -- including Google, Bing, Yahoo and other search engines. (Webopedia, 2017).* SEO will be the final piece added to the SOS Threat Analytics project and is a 'nice to have' element.

### ➤ SQL Server

SQL Server was used for the design and implementation of the SOS Threat Analytics database – a key part to the project and final deliverable. Back-end integration to the Apache Spark application as well as front-end integration to ASP.net are what holds the solution together.

*SQL Server is a Microsoft product used to manage and store information. Technically, SQL Server is a "relational database management system" (RDMS). Broken apart, this term means two things. First, that data stored inside SQL Server will be housed in a "relational database", and second, that SQL Server is an entire "management system", not just a database. (DatabaseJournal, 2008).*

### ➤ Stored Procedure

*A stored procedure is a group of one or more database statements stored in the database's data dictionary and called from either a remote program, another stored procedure, or the command line. (EssentialSQL, 2017).* A detailed stored procedure has been utilized to call the alerting functionality within SOS Threat Analytics.

### 3. Technologies

A plethora of different technologies were considered as part of this project implementation. The final technology set has changed somewhat from the initial infrastructure design but this had no impact on the end-deliverables. The tools and technologies utilised focussed on several key areas i.e. NetFlow generation, data streaming & filtering, data storage and application front-end development. Furthermore, the decision was taken to create an online marketing website which had a requirement for external hosting and SEO (Search Engine Optimisation). The final technology components have been listed in the table below (Fig1) and the rationale for these choices has been subsequently broken out in more detail.

Function/Service	Chosen Technology
<b>NetFlow Generation</b>	Paessler NetFlow Generator
<b>Development Environment (IDE)</b>	Eclipse
<b>Data Streaming</b>	Apache Spark
<b>Data Filtering</b>	
<b>Relational Database / Data Storage</b>	SQL Server
<b>Front-End Web Application</b>	ASP.Net MVC
	Bootstrap
	JavaScript
<b>Marketing Website</b>	Bootstrap
	JavaScript
<b>Web Hosting</b>	Amazon S3
	SEO

Fig1: Table of required Functions and Chosen Technologies on which they were delivered

#### 3.1 Paessler NetFlow Generator

Ultimately, it would have been ideal to have the use of a Cisco device in-order to produce the NetFlow data directly. However, this would be highly cost prohibitive and not conducive to getting the project completed. Paessler (producers of network monitoring software), have developed an open source NetFlow generator which was utilized to simulate NetFlow traffic from a Cisco device. The use of this software allowed for the validation of the project concept and solution in the absence of a production-like environment. This tool was downloaded from the following location: (<https://www.paessler.com/tools/NetFlowgenerator>)

### 3.2 Eclipse IDE

As the choice was made to develop in Java, the most functional IDE on the market is Eclipse. The other alternative was NetBeans, but as this project was using Apache Spark Streaming, all information available online pointed to the use of Eclipse as the preferred option.

### 3.3 Apache Spark

For the purposes of this project, an Apache Spark (Streaming) context was developed in Java as a method of ingesting, streaming and filtering the incoming NetFlow traffic. At a high level, the NetFlow generated is set to target the IP address and port of the Apache Spark context, which actively listens for connections on that specific IP and port combination. This context was designed with multiple operations, to serialize, stream and filter the relevant NetFlow information for transfer to a waiting SQL server instance.

### 3.4 SQL Server

The foundation of this project was based around data. It should be no surprise that the database itself is perhaps the most crucial aspect of the overall infrastructure. With direct links to both the data generation and end user visualisation, the successful design, configuration and implementation of the database structure was essential to the overall project functionality. SQL was chosen due to a combination of previous experience and ease of integration with an ASP.net front end.

### 3.5 ASP.net MVC

The front-end web application was developed in C# using Microsoft's ASP.net MVC framework. Previously familiarity and experience with this product informed the final decision. The goal was to create a customer facing application that delivered on all project requirements in terms of data visualization, look & feel, navigability and ease of use. It's seamless integration with SQL server also proved a driving factor in this technology choice. The look and feel is - by default - created using a combination of Twitter Bootstrap and JavaScript within the MVC environment.

### 3.6 Marketing Website

To present the specifics of the product to an online audience, a static marketing website was also developed. The site presents a single page view with a scroll aspect to cover key information such as product functionality, company information, contact details etc. Consideration was given to the idea of merging the website and front-end application however this was dismissed at an early stage. The marketing website is a separate entity, developed entirely in Bootstrap with elements of JavaScript and web plug-ins to add some stylish features to the look and feel. The site acts as a key marketing tool to promote the value of the final product and offers another stream of activity to the end deliverable.

#### 3.6.1. Web Hosting

The marketing website is fully hosted in Amazon S3. This platform is typically used for file storage, however when developing static websites, it is also seen as an ideal solution. Initial research demonstrated that this was a quick and easy solution that would work well for

delivering a highly accessible/available marketing website. This proved to be the case and the hosting element has proved stable, reliable and cost efficient! This site has been provisionally hosted here, before being migrated to a permanent address of: [www.sosthreatanalytics.com](http://www.sosthreatanalytics.com)

### 3.6.2. Search Engine Optimisation

To establish increased visibility online, a level of SEO was applied to the site. This delivers a higher level of online recognition and an appropriate ranking on search engines such as Google, Bing etc.

## 4. Cisco NetFlow

This project is focused around the fundamental concept of ingesting, streaming, filtering and visualising data. It is therefore crucial to understand the type of data on which these operations will be occurring. SOS Threat Analytics focusses purely on NetFlow data i.e. all analysis and threat detection is based on network traffic emanating from Cisco devices. NetFlow is a proprietary protocol that is exclusive to Cisco and can empower network analysts by providing detailed information such as the source and destination address of a packet, the target port, the protocol etc. "NetFlow is a networking protocol designed by Cisco Systems for logging and recording the flow of traffic received and sent within a network." (Techopedia, 2017). SOS Threat Analytics looks to harness that information and use it to actively identify potential network threats with a view to alerting administrators within seconds of detection.

So how are these packets to be analysed? Firstly, it is imperative that the structure and breakdown of each packet is known and understood. A typical Cisco NetFlow packet is broken out into two sections: 1) The Header and 2) the Flow Data. Both sections contain multiple different pieces of information about the packet – these are shown below in Fig2:

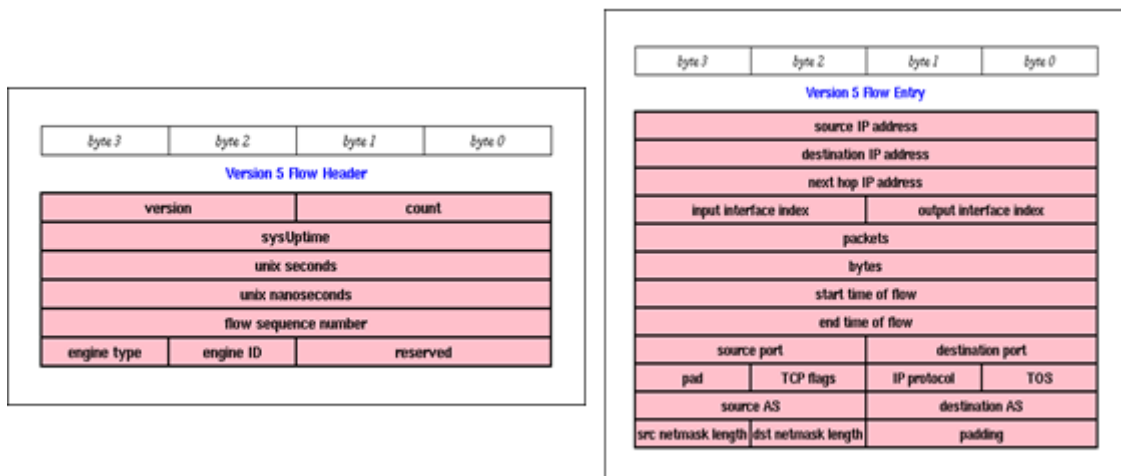
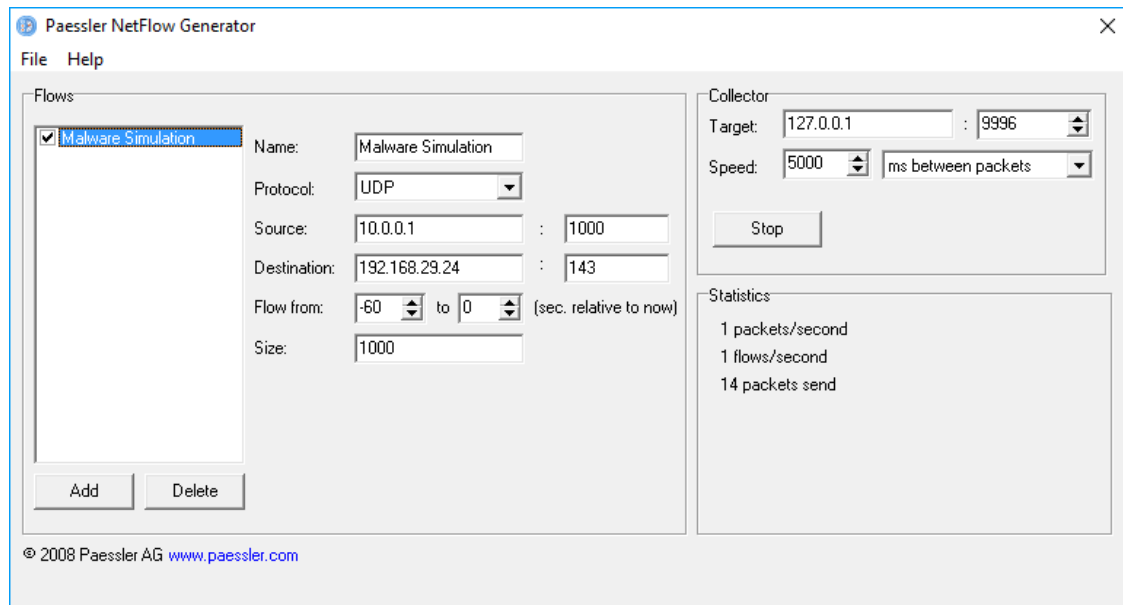


Fig2: NetFlow Packet Header & Flow Data Structures

For the purposes of this project, a feasible method of NetFlow generation would be required. Acquiring a dedicated Cisco switch of any kind would be difficult, not to mention cost prohibitive. Through online research the aforementioned: 'Paessler NetFlow Generator' was

discovered. "NetFlow Generator creates artificial NetFlow Version 5 data streams without the need for NetFlow compatible hardware. It is a perfect tool to test the NetFlow functionality of PRTG or other NetFlow compatible programs. (Paessler, 2008)." This tool allows for the simulation of Cisco standard NetFlow data, effectively delivering a software based alternative that acts as the Cisco Hub (or multiple Cisco hub) to enable traffic analysis. The configured tool is setup as below:



**Fig3: Paessler NetFlow Generator**

Fig3 demonstrates a 'Flow' that has been configured with a destination IP and Port of a known threat vector. The protocol has been set to UDP and the flow has been named 'Malware Simulation'. The area labelled 'Collector' sets out where you would like to access this flow and over what connection. SOS Threat Analytics targets the local Eclipse IDE on the machine's localhost (127.0.0.1) over port 9996. (A production implementation would target a physical or virtual machine with this embedded programming language environment). The 'Statistics' section details the number of packets that have been successfully generated/sent since the flow was initiated. The challenge from the project perspective was the understand the flow of this information and to create a method to ingest and stream it.

The first step to achieving this was to be able to visualise and analyse the 'Flows' being generated from the Paessler program. Effectively, it was crucial to verify that what was being sent/received matched the expectation in terms of packet structure, set out in Fig2. As a starting point, Wireshark was used to visualise an incoming packet from the 'Paessler NetFlow Generator' to ensure that the data could be captured and the various elements within would be visible. Fig4 shows a Wireshark capture from a typical Cisco NetFlow packet i.e. proving that the data from the tool could indeed be captured:

No.	Time	Source	Destination	Protocol	Length	Info	Dst
521	17.632425	192.168.223.76	192.168.223.77	CFLOW	114	total: 1 (v5) flow	9996
773	27.632983	192.168.223.76	192.168.223.77	CFLOW	114	total: 1 (v5) flow	9996
1017	37.636244	192.168.223.76	192.168.223.77	CFLOW	114	total: 1 (v5) flow	9996
1483	47.640586	192.168.223.76	192.168.223.77	CFLOW	114	total: 1 (v5) flow	9996
1775	57.645869	192.168.223.76	192.168.223.77	CFLOW	114	total: 1 (v5) flow	9996

Fig4: Paessler Generated NetFlow Packets being captured within Wireshark

Wireshark captures NetFlow data under a protocol known as CFLOW (something that was established after searching blindly within the tool for a significant amount of time!!). Expanding the data on one of these connections provides a more fine-grained level of detail. The constituent parts of the packet and their alignment to the Header and Flow Data structures shown in Fig2 become evident. Fig5 (below) shows a breakdown of the 'Header' within Wireshark.

```

> Frame 45: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
> Ethernet II, Src: IntelCor_74:b6:c0 (c4:d9:87:74:b6:c0), Dst: CompalBr_51:60:ee (54:67:51:51:60:ee)
> Internet Protocol Version 4, Src: 192.168.0.129, Dst: 192.168.223.77
> User Datagram Protocol, Src Port: 51387, Dst Port: 9996
v Cisco NetFlow/IPFIX
  Version: 5
  Count: 1
  SysUptime: 604610.968000000 seconds
  > Timestamp: Jun 6, 2002 17:21:20.551123316 GMT Daylight Time
  FlowSequence: 2539030436
  EngineType: Unknown (254)
  EngineId: 255
  11.. .... .... = SamplingMode: Unknown (3)
  ..11 1111 1111 1111 = SampleRate: 16383
  > pdu 1/1
    
```

Fig5: NetFlow Packet 'Header' structure expanded within Wireshark

```

v pdu 1/1
  SrcAddr: 10.0.0.1
  DstAddr: 10.0.0.2
  NextHop: 0.0.0.0
  InputInt: 0
  OutputInt: 0
  Packets: 0
  Octets: 1000
  > [Duration: 60.000000000 seconds]
  SrcPort: 1000
  DstPort: 80
  Padding: 80
  TCP Flags: 0x69
  Protocol: UDP (17)
  IP ToS: 0xff
  SrcAS: 65535
  DstAS: 65535
  SrcMask: 0 (prefix: 10.0.0.1/32)
  DstMask: 0 (prefix: 10.0.0.2/32)
  Padding: 0000
    
```

Expanding the item labelled 'pdu 1/1' at the bottom of Fig5 allows you to open the 'Flow Data' section of the packet to see its more detailed information (Fig6). This is of crucial importance as it demonstrates that the structure of the NetFlow packet within Wireshark is directly in line with initial expectations i.e. from Fig2. This 'Proof of Concept' (being able to demonstrate that the NetFlow packet from the Paessler tool could in fact be interpreted out into its constituent parts) formed the baseline for the next step i.e. physically dissecting a packet, streaming it, and storing the output. This will be broken out in detail as part of Section9 – Implementation.

Fig6: NetFlow Packet 'Flow Data'

## 5. Requirements Elicitation

When engaging in any level of project design or implementation, it is critical to perform the requisite primary research to ensure that the concept is valid, well thought out and applicable to your chosen market or customer base. It is exceptionally important to engage with potential users/customers to accurately understand their requirements and expectations from a given product. As such, focussed questionnaires and conversations (face to face and phone) were conducted with five individuals who were well placed to provide applicable and actionable feedback. These individuals were chosen due to their fields of expertise and levels of familiarity with technology. Furthermore, they also demonstrated a genuine interest in the project following initial telephone discussions.

All five individuals were asked to review the 'Project Proposal' document in advance of completing a structured questionnaire so that they could attain a strong understanding of both the core concepts and end-goals of the project. This level of engagement proved a highly useful exercise and assisted in shaping the core set of requirements for the end-product. All five interviewees later engaged as 'software testers' once there was an accessible version of the application (test results are detailed later within this document). The basic set of user requirements can be found in 'Section 6' of this document and the completed questionnaires can be found in the 'Appendices'.

## 6. Basic User Requirements Definition

The core functionality – established with the help of the structured interviews and questionnaires – that was required for the application to function at an optimum level from an end-user perspective is as follows:

- Initial Application connectivity/accessibility
- Ability to register a user account
- Area to Sign-In & Sign-Out (Registered Users)
- Ability to browse network activity sections (site navigability)
- Ability to print details
- Ability to log support cases/queries/provide feedback
- Facility to receive alerts from the application
  - Email
  - SMS
- Ability to access Marketing Website

## 7. Requirements Specification

Requirements, both functional & non-functional will be detailed in the below section. Collectively this will deliver a solid picture of the collective requirements that were core to the overall solution design and determination of the application functionality and performance in production. Section 6 detailed the basic user requirements for the application and these have expanded/evolved as the product has moved through different iterations of design, testing and implementation.

### 7.1 Functional Requirements – Application

The functional requirements in this section highlight all fundamental functionality that that was implemented for the application to run in the desired manner and to operate all core functionality as expected. As previously mentioned, these core functional requirements were initially constructed and informed by the primary research executed with potential users. Firstly, below is a use case diagram for the complete system:

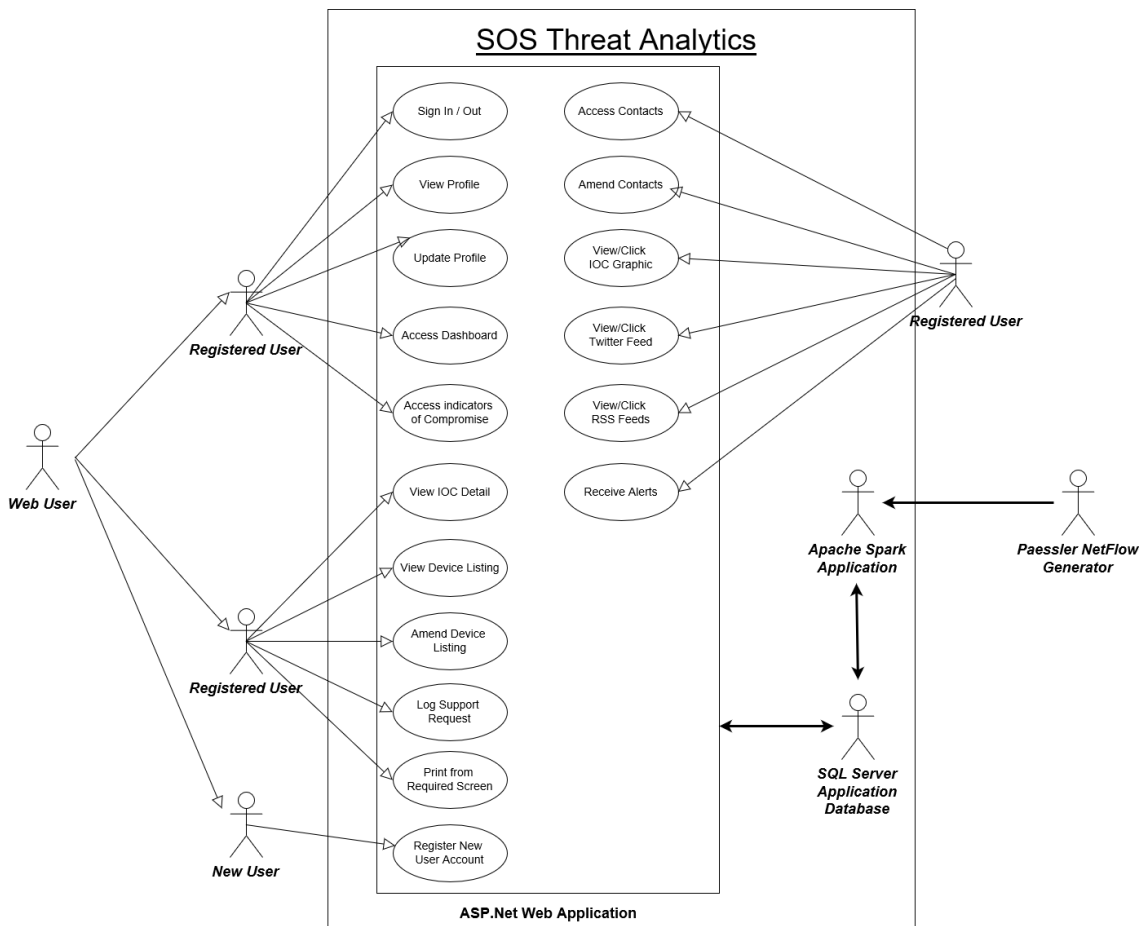


Fig7: SOS Threat Analytics – High Level System Overview

The functional requirements identified will now be broken out and expanded as Use Cases in structured tables detailing their basic scenarios. These Use Cases will give an overview of the flow of some key activities and user interactions.



**7.1.1. User Registration**

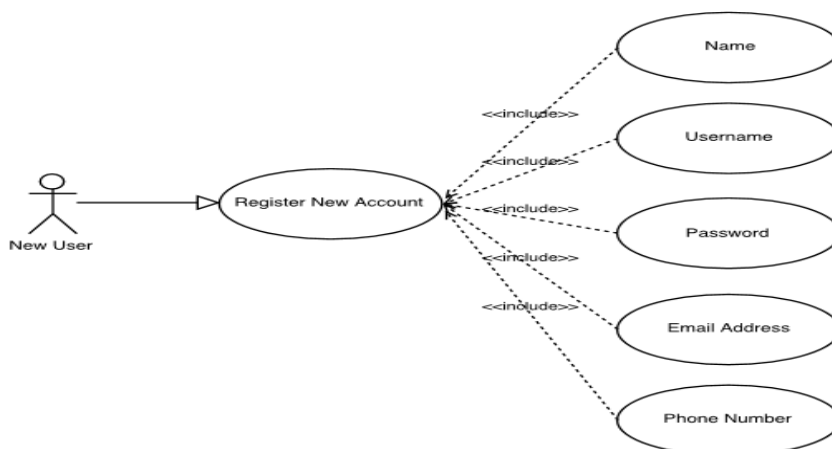
➤ Description & Priority

This requirement relates to an 'unregistered' user who is required to create a new application account to become an active or 'Registered User'. This process is crucial as without it, no users can have accounts on the system and thus cannot access any of the functionality/services.

➤ Use Case

Below is the Use Case for User Registration:

<b>Description/Scope</b>	Allows an individual user to register with the application and create a user account to gain access to the available services etc.
<b>Pre-Condition</b>	User has not yet registered an account
<b>Activation</b>	'New User' accesses the application and clicks on the 'Registration' link
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User visits the 'Registration page' and enters all requested information i.e. name, email etc.</li> <li>Application displays a confirmation page/message and advises the user that they have entered the information correctly and that their account has been successfully created</li> </ul>
<b>Alternate Flow</b>	<p><u>Fields not completed:</u></p> <ul style="list-style-type: none"> <li>User has not completed all relevant fields so the application will highlight all required fields and wait until the user re-submits with the correct information</li> </ul> <p><u>Username already exists</u></p> <ul style="list-style-type: none"> <li>User supplies a 'username' that is already registered. The application will re-direct the user to the registration page (blank)</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User Successfully registers account</li> <li>Alternate Flow: User fails to register</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: User reverted to Home Page</li> <li>Alternate Flows: User must attempt to register again</li> </ul>



**Fig8: Use Case Diagram for User Registration**

**7.1.2. User Sign-In**

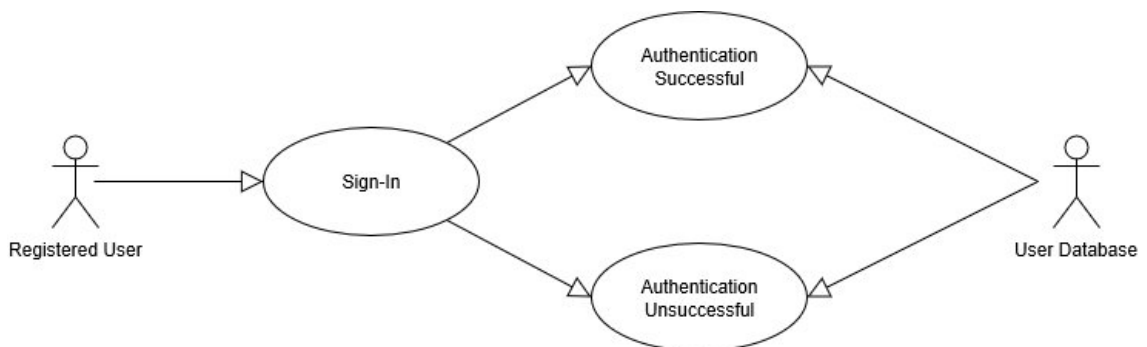
➤ Description & Priority

This requirement presents the ability for the 'Registered User' to log into the system. This requirement is key to the system in terms of allowing the user access to functionality as well as access to their own profile etc.

➤ Use Case

Below is the Use Case for User Sign-In:

<b>Description/Scope</b>	Allows a registered user to identify themselves on the application and access the required functionality/services via their designated account.
<b>Pre-Condition</b>	User holds a valid account but has not yet authenticated onto the application during their session.
<b>Activation</b>	The use case starts when a 'Registered user' visits the log-in page of the application, enters their credentials and clicks Sign-In.
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>'Registered User' navigates to the application and enters the required credentials followed by clicking the 'Sign-in' button.</li> <li>System validates their credentials (authentication) &amp; now provides access to the application under their account</li> <li>User can now access their profile and view all navigable pages that are available to them.</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>'Registered User' opens the application and enters invalid credentials followed by clicking the 'Sign-in' button.</li> <li>Application displays an error requesting that the User re-enters their credentials and again attempts to authenticate.</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: Credentials authenticated</li> <li>Alternate Flow: Credentials unauthenticated</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: User can access application and functionality</li> <li>Alternate Flow: User returned to sign-in page</li> </ul>



**Fig9: Use Case Diagram for User Sign-In**

**7.1.3. View User Profile**

➤ Description & Priority

This requirement presents the ability for the 'Signed-In User' to view their account details. Offers the opportunity for a user to review their details so that they can confirm they are still relevant etc. This requirement is not deemed critical however it is a useful function for each individual user.

➤ Use Case

Below is the Use Case for View User Profile:

<b>Description/Scope</b>	Allows the user access to their profile page where they can view their permanent information (Name, email address etc.)
<b>Pre-Condition</b>	User must be registered i.e. have a valid user account
<b>Activation</b>	Users clicks on their name on the top right of the screen
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>• Users clicks on their name on the top right of the screen</li> <li>• User can view their details</li> <li>• User has the option to revert to the main page</li> <li>• User has the option to edit their details</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>• User clicks on their name on the top right of the screen</li> <li>• Profile is not accessible                             <ul style="list-style-type: none"> <li>○ Issue with application</li> <li>○ Users internet connection issue</li> </ul> </li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>• Main Flow: User successfully views their Profile</li> <li>• Alternate Flow: User cannot access their profile</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>• Main flow: User can remain on or navigate away from page</li> <li>• Alternate Flow: User has no access – investigation required</li> </ul>

**7.1.4. Update User Profile**

➤ Description & Priority

This requirement presents the ability for the 'Signed-In User' to review and edit their account details. This requirement is not deemed critical however it is important in respect to the details required for event notification etc. Users must ensure that their relevant contact details (phone and email) are kept up to date.

➤ Use Case

Below is the Use Case for Update User Profile:

<b>Description/Scope</b>	Allows the user access to their profile page and be granted the ability to update their personal account details (Name, email address etc.)
<b>Pre-Condition</b>	User must be registered i.e. have a valid account with the site
<b>Activation</b>	User clicks on the 'Update Profile' icon/link
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>• User clicks option to 'View Profile'</li> <li>• User can view their details</li> <li>• User clicks – 'Update Profile' option</li> <li>• User makes necessary changes</li> <li>• User chooses to 'Save' necessary changes</li> <li>• Database updates profile, stores and saves the new information</li> <li>• User reverted to 'View Profile' page and can see updated details</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>• User decides against making changes</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>• Main Flow: User saves changes</li> <li>• Alternate Flow: User navigates from page</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>• Main Flow: User profile is updated</li> <li>• Alternate Flow: User profile in left unchanged</li> </ul>

**7.1.5. Application Navigation**

➤ Description & Priority

The navigation across any application or website is of critical importance as it allows the user to access the functionality and information they require. The requirement allows the user to browse to different application sections such as support, reports etc.

➤ Use Case

Below is the Use Case for Application Navigation:

<b>Description/Scope</b>	Allows users to browse different sections of the application
<b>Pre-Condition</b>	User must be registered i.e. have a valid application account
<b>Activation</b>	Users' clicks away from main page to an alternative menu option

<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User clicks on menu option (profile, reports etc.)</li> <li>User is presented with requested page</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>User remains on original page</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User halts navigation at requested page</li> <li>Alternate Flow: User remains on original page</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: User accesses/views requested page</li> <li>Alternate Flow: User remains on same page</li> </ul>

### 7.1.6. Access Dashboard

➤ Description & Priority

The 'Dashboard' section within SOS Threat Analytics delivers a high-level overview of all threat activity. A JavaScript plug-in delivers a visual representation of the number of live threats that have hit the network. Furthermore, the dashboard provides a host of news feeds including Twitter and RSS feeds related to Information Security

➤ Use Case

Below is the Use Case for Accessing the Dashboard:

<b>Description/Scope</b>	Allows access to the 'Dashboard' section of the application
<b>Pre-Condition</b>	User must be registered i.e. have a valid application account
<b>Activation</b>	User clicks on the 'Dashboard' menu option
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User clicks on the 'Dashboard' menu option</li> <li>User is presented with the Dashboard page</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>User remains on original page</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User halts navigation at Dashboard page</li> <li>Alternate Flow: User remains on original page</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: User accesses/views Dashboard page</li> <li>Alternate Flow: User remains on same page</li> </ul>

### 7.1.7. View IOC Graphic

The first thing that is visible on the Dashboard is a graphic (pie or bar chart) that offers a visual representation of the number of potential threats that have hit the users' infrastructure to that date. We refer to these threats as 'Indicators of Compromise' (IOC). This graphic appears at the top left-hand side of the 'Dashboard view'.

➤ Use Case

Below is the Use Case for Viewing the IOC Graphic on the Dashboard:

<b>Description/Scope</b>	Delivers a visible representation of the number of IOCs hitting the user's infrastructure
<b>Pre-Condition</b>	User must be registered i.e. have a valid application account
<b>Activation</b>	User clicks on the 'Dashboard' menu option
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User clicks on the 'Dashboard' menu option</li> <li>User is presented with the Dashboard page including the 'IOC Graphic' in the top left of the display</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>User remains on original page (does not click Dashboard)</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User halts navigation at Dashboard page</li> <li>Alternate Flow: User remains on original page</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: User accesses/views the IOC Graphic</li> <li>Alternate Flow: User remains on same page</li> </ul>

**7.1.8. View Twitter Feed**

➤ Description & Priority

Embedded within the Dashboard is a Twitter feed directly linked to the official 'SOS Threat Analytics' twitter account. This is a useful interface between the product team and end user, delivering up to date news as soon as it comes to light. SOS Threat Analytics highlight network specific threats to ensure that clients are fully aware of the ever-expanding threatscape.

➤ Use Case

Below is the Use Case for Viewing the SOS Threat Analytics Twitter Feed:

<b>Description/Scope</b>	Allows user to see a live Twitter Feed from SOS Threat Analytics
<b>Pre-Condition</b>	User must be registered i.e. have a valid application account
<b>Activation</b>	User clicks on the 'Dashboard' menu option
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User clicks on the 'Dashboard' menu option</li> <li>User is presented with the Dashboard page including the live SOS Threat Analytics Twitter feed</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>User remains on original page (does not click Dashboard)</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User halts navigation at Dashboard page</li> <li>Alternate Flow: User remains on original page</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: User accesses/views Twitter Feed</li> <li>Alternate Flow: User remains on same page</li> </ul>

**7.1.9. View Security Based RSS Feeds**

➤ Description & Priority

Embedded within the Dashboard is a set of pre-configured security related RSS Feeds. By default, 3 feeds are provided and these can be changed at a user’s request. RSS feeds such as ‘Krebs on Security’ provide the latest in cyber threat news and can be an invaluable tool in keeping users informed on current risks and security trends.

➤ Use Case

Below is the Use Case for Viewing the set of Security based RSS Feeds on the Dashboard:

<b>Description/Scope</b>	Allows users access to live security related RSS Feeds
<b>Pre-Condition</b>	User must be registered i.e. have a valid application account
<b>Activation</b>	User clicks on the ‘Dashboard’ menu option
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>• User clicks on the ‘Dashboard’ menu option</li> <li>• User is presented with the Dashboard page including the live SOS Threat Analytics Twitter feed</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>• User remains on original page (does not click Dashboard)</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>• Main Flow: User halts navigation at Dashboard page</li> <li>• Alternate Flow: User remains on original page</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>• Main Flow: User accesses/views Twitter Feed</li> <li>• Alternate Flow: User remains on same page</li> </ul>

**7.1.10. Access the Indicators of Compromise Page**

➤ Description & Priority

The Indicators of Compromise (IOC) page is perhaps the most important area of the SOS Threat Analytics application. It is in here that a user can access an overview of the potential threats that have been flagged on their network. Furthermore, they can request a detailed breakdown of the threat which will include advisory information on rectifying/removing the threat as well as preventing future reoccurrence. There are two ways to access the IOC page and both will be broken out below in separate use cases:

➤ Use Case for Nav-Bar Access

Below is the Use case for accessing the IOC page from the Nav-Bar at the top of the page

<b>Description/Scope</b>	Allows users access to ‘Indicators of Compromise’ page which provides details of all potential threats flagged on the network
<b>Pre-Condition</b>	User must be registered i.e. have a valid application account
<b>Activation</b>	User clicks on the ‘Indicators of Compromise’ Nav-Bar menu option
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>• User clicks on the ‘Indicators of Compromise’ Nav-Bar menu option</li> <li>• User is presented with the Indicators of Compromise page which provides high-level details on all potential network threats</li> </ul>

<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>User remains on original page (does not click IOC Nav-Bar menu)</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User halts navigation at IOC page</li> <li>Alternate Flow: User remains on original page</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: User accesses/views IOC page and reviews information</li> <li>Alternate Flow: User remains on same page</li> </ul>

➤ Use Case for Access from IOC Graphic on Dashboard

Below is the Use case for accessing the IOC page by clicking the IOC Graphic on the Dashboard page:

<b>Description/Scope</b>	Allows users access to 'Indicators of Compromise' page which provides details of all potential threats flagged on the network
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>User must be registered i.e. have a valid application account</li> <li>User has navigated to the 'Dashboard' page</li> </ul>
<b>Activation</b>	User clicks on the 'IOC Graphic' within the 'Dashboard' page
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User clicks on the 'IOC Graphic' within the 'Dashboard' page</li> <li>User is presented with the Indicators of Compromise page which provides high-level details on all potential network threats</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>User remains on original page (does not click IOC Graphic)</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User halts navigation at IOC page</li> <li>Alternate Flow: User remains on original page</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: User accesses/views IOC page and reviews information</li> <li>Alternate Flow: User remains on same page</li> </ul>

**7.1.11. Access Detailed IOC Information**

➤ Description & Priority

One of the key selling points of the SOS Threat Analytics application is the ability to highlight potential issues to a user. Perhaps however, the unique selling point of the product is the ability to drill down further into each identified 'Indicator of Compromise' to access an in-depth overview of the issue, its potential impact as well as methods of resolution and future prevention. Within the IOC page there is an option to click 'Detailed IOC Information' where the high-level details of the threat will be presented alongside an 'iFrame' containing the relevant security advisory document. This is a critical aspect of the application and a key piece of functionality.

➤ Use Case to access 'Detailed IOC Information'

Below is the Use case for accessing the Detailed IOC Information page

<b>Description/Scope</b>	Allows the user to view and review and in-depth breakdown of the potential threat, proposed resolution and future preventive actions.
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>User must be registered i.e. have a valid application account</li> <li>User has navigated to the 'Indicators of Compromise' page</li> </ul>



<b>Activation</b>	User clicks 'Detailed IOC Information' option
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User clicks on the 'Detailed IOC Information' within the 'IOC' page</li> <li>User is presented with an in-depth breakdown of the potential threat accompanied by an embedded document/web page that describes the threat, potential avenues for resolution and options for future preventative measures.</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>User remains on IOC page</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User halts navigation or navigates away from page</li> <li>Alternate Flow: User remains on IOC page</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: User has access to a detailed breakdown of the chosen IOC</li> <li>Alternate Flow: User remains on IOC page</li> </ul>

**7.1.12. Access a list of monitored Devices**

➤ Description & Priority

Users of the application will want to ensure that an appropriate list of network devices are catered for within their environment. They will also want to be able to check and validate that the list of monitored devices is in line with their expectation. The 'Device Listing' Page provides this functionality and delivers an easily understandable overview of the list of devices in scope for the SOS Threat Analytics application.

➤ Use Case for Accessing a List of Devices

Below is the use case to access the list of devices in scope for monitoring

<b>Description/Scope</b>	Allows the user to view the list of devices in scope for monitoring
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>User must be registered i.e. have a valid application account</li> <li>User has logged into the application</li> </ul>
<b>Activation</b>	User clicks 'Device Listing' option from the Nav-Bar
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User clicks 'Device Listing' option from the Nav-Bar</li> <li>User is presented with list of all devices in scope for monitoring by the SOS Threat Analytics application</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>User fails to access page</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User halts navigation or navigates away from page</li> <li>Alternate Flow: User continually fails and cannot gain access</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: User has access to a full list of devices</li> <li>Alternate Flow: User logs a support request with SOS Threat Analytics</li> </ul>

**7.1.13. Access Device Details**

➤ Description & Priority

Rather than simply viewing a list of devices, the user has a further option to view each device individually alongside a visual of the device in question. In a production environment, this is a helpful addition if the device is in a computer room or in an off-site location as it can refresh the user’s memory as to the physical dimensions, ports available etc. on a particular device.

➤ Use Case for Viewing Device Details

Below is the use case to access the details of an individual device

<b>Description/Scope</b>	Allows the user to view a single device in detail alongside a visual of the device
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>• User must be registered i.e. have a valid application account</li> <li>• User has logged into the application</li> <li>• User has navigated to the 'Device Listing' page</li> </ul>
<b>Activation</b>	User clicks 'Details' option for an individual device
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>• User clicks "Details' option for an individual device</li> <li>• User is presented with detailed information about the chosen device alongside a visual of that device</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>• User fails to access page</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>• Main Flow: User halts navigation or navigates away from page</li> <li>• Alternate Flow: User continually fails and cannot gain access</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>• Main Flow: User has access to the full details of the device</li> <li>• Alternate Flow: User logs a support request with SOS Threat Analytics</li> </ul>

**7.1.14. Access Vendor & Client Contacts**

➤ Description & Priority

To ensure that both the vendor (SOS Threat Analytics) and the client understand the preferred contact list for queries, support etc. The 'Contacts' page details all support contact email addresses and telephone numbers for teams/individuals engaged in the monitoring process/contract.

➤ Use Case for Viewing the Contacts

Below is the use case to access 'Contacts' list

<b>Description/Scope</b>	Allows the user to view the list of key contacts for the support contract
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>• User must be registered i.e. have a valid application account</li> <li>• User has logged into the application</li> </ul>
<b>Activation</b>	User clicks 'Contacts' option from the Nav-Bar
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>• User clicks 'Contacts' option from the Nav-Bar</li> <li>• User is presented with the list of key Contacts for the support contract</li> </ul>

<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>User fails to access page</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User halts navigation or navigates away from page</li> <li>Alternate Flow: User continually fails and cannot gain access</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: User has access to a full list of contacts</li> <li>Alternate Flow: User logs a support request with SOS Threat Analytics</li> </ul>

**7.1.15. Adding/Deleting/Amending a Contact**

➤ Description & Priority

Not a high priority but a facility has been provided to enable the user to add, edit or delete a contact as they see fit. This ensure that the dedicated 'Contacts' list remains current and relative. Each use case has been broken out below.

➤ Use Case for Adding a Contact

Below is the use case to add to the 'Contacts' list

<b>Description/Scope</b>	Allows the user to add a new contact to the 'Contacts List'
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>User must be registered i.e. have a valid application account</li> <li>User has logged into the application</li> <li>User has navigated to the 'Contacts' page</li> </ul>
<b>Activation</b>	User clicks 'Create new' option from the Contacts Page
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User clicks 'Create new' option from the Contacts Page</li> <li>User presented with several fields for completion to add a new user</li> <li>User completes all fields and adds the user as required</li> </ul>
<b>Alternate Flows</b>	<ul style="list-style-type: none"> <li>User does not complete all required fields</li> <li>User cannot access required page</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: New User is successfully added to the contacts list</li> <li>Alternate Flow1: User must re-complete the required fields</li> <li>Alternate Flow2: User continually fails and cannot gain access</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: New Contact added to list</li> <li>Alternate Flow: Fails to add new Contact</li> </ul>

➤ Use Case for Editing a Contact

Below is the use case to edit and existing 'Contact' within the list:

<b>Description/Scope</b>	Allows the user to amend an existing contact in the 'Contacts' list
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>User must be registered i.e. have a valid application account</li> <li>User has logged into the application</li> <li>User has navigated to the 'Contacts' page</li> </ul>
<b>Activation</b>	User clicks 'Edit option for a particular contact name

<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User clicks 'Edit option for a particular contact name</li> <li>User presented with several fields that can be edited</li> <li>User makes required changes</li> </ul>
<b>Alternate Flows</b>	<ul style="list-style-type: none"> <li>User does not make any changes</li> <li>User cannot access required page</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User makes all required changes and saves Contact</li> <li>Alternate Flow1: User navigates from change without making a change</li> <li>Alternate Flow2: User continually fails and cannot gain access</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: Change made to specific 'Contact'</li> <li>Alternate Flow: Contact remains unchanged</li> </ul>

➤ Use Case for Deleting a Contact

Below is the use case to delete from the 'Contacts' list

<b>Description/Scope</b>	Allows the user to delete an existing contact from the 'Contacts List'
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>User must be registered i.e. have a valid application account</li> <li>User has logged into the application</li> <li>User has navigated to the 'Contacts' page</li> </ul>
<b>Activation</b>	User clicks 'Delete' option from the Contacts Page
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User clicks 'Delete' option from the Contacts Page</li> <li>User presented with confirmation page for deletion</li> <li>User confirms deletion</li> </ul>
<b>Alternate Flows</b>	<ul style="list-style-type: none"> <li>User does not confirm deletion</li> <li>User cannot access required page</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User clicks delete and is reverted to contacts list</li> <li>Alternate Flow1: User navigates away with no deletion action</li> <li>Alternate Flow2: User continually fails and cannot gain access</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: Contact successfully deleted</li> <li>Alternate Flow: No change to existing Contacts list</li> </ul>

**7.1.16. Log a Support Request**

➤ Description & Priority

If a user experiences an issue with the application, it is important that they can log a support call and get some assistance. Whilst not deemed a core feature of the application this is a nice to have and a good customer service element to the overall product.

➤ Use Case

Below is the Use Case to Create a Support Request:

<b>Description/Scope</b>	Facility to create a support request on the application
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>User must be registered i.e. have a valid application account</li> <li>User has logged into the application</li> </ul>

<b>Activation</b>	User clicks on 'Contacts' page
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User clicks on 'Contacts' page from the Nav-Bar</li> <li>User presented with an email form on bottom right of page</li> <li>User populates details</li> <li>User clicks send</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>User clicks on 'Contacts' page from the Nav-Bar</li> <li>User presented with an email form</li> <li>User does not complete request</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: User clicks send</li> <li>Alternate Flow: User navigates from page</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main flow: Query is logged with support</li> <li>Alternate Flow: No query is logged</li> </ul>

### 7.1.17. Print Pages

➤ Description & Priority

Delivers the ability to print details directly from the application e.g. IOC's, Device Details etc. This is an attractive function to an engaged end-user with a focus on network activity.

➤ Use Case

Below is the Use Case to print a page:

<b>Description/Scope</b>	Facility to print details directly from the application
<b>Pre-Condition</b>	<ul style="list-style-type: none"> <li>User must be registered i.e. have a valid application account</li> <li>User has logged into the application</li> </ul>
<b>Activation</b>	User navigates to the required page (IOC, Device Listing etc.)
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>User navigates to the required page (IOC, Device Listing etc.)</li> <li>User clicks Print option</li> <li>User configures local printer</li> <li>Job is printed</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>User navigates to the required page (IOC, Device Listing etc.)</li> <li>User fails to print details</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: Details are printed and user navigates from page</li> <li>Alternate Flow: User fails to print details</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main flow: Print job is successfully completed</li> <li>Alternate Flow: No printing takes place, user may log a support call</li> </ul>

### 7.1.18. Alerting & Notifications

➤ Description & Priority

Whilst the capture, filtering and visualisation is at the core of this project it is also worth noting the importance of alerting users to the issues identified. This is a core service offering to the end user and is the key selling point for the product. It is essential for the application to have the necessary capabilities to communicate with the end users via the email and SMS details they will have provided within the contacts list.

➤ Use Case

Below is the Use Case for Alerts & Notifications:

<b>Description/Scope</b>	Registered users can receive notifications around network activity – particularly of a spurious/potentially malicious nature
<b>Pre-Condition</b>	User must hold a registered account and have provided the required details i.e. email address and mobile number
<b>Activation</b>	Event occurs onsite that requires user notification
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>Event occurs; user(s) receives notification via either phone or email. User then takes appropriate action.</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>Event occurs but user fails to receive notification</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: Application sends required notification</li> <li>Alternate Flow: No notification sent to user</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: User receives notification from site.</li> <li>Alternate flow: Site fails to send notification. User to log a support call with SOS Threat Analytics. Technical and contact details to be reviewed and tested.</li> </ul>

## 7.2 Functional Requirements – System

The functional requirements in this section will highlight all fundamental functionality that will need to be in operation for the back-end system operations to function at an optimal level. These core functionalities relate to the infrastructure and generation of data to which the user will be given access. This side of the functionality will not be directly seen or interacted with, by the user.

### 7.2.1. NetFlow Generation

The core concept of this project is the interpretation of NetFlow data to better inform the end-user of the underlying activity on their network. Therefore, the most fundamental aspect of the project is the generation of that NetFlow data. In the absence of a live Cisco switch – due to prohibitive expense – NetFlow generation will be done using a software based solution. The software tool is open source and has been developed by Paessler (see Section4).

➤ Use Case

Below is the Use Case for NetFlow Generation:

<b>Description/Scope</b>	Facility to generate NetFlow Data
<b>Pre-Condition</b>	Paessler NetFlow generator has been installed successfully and is functioning as expected
<b>Activation</b>	A stream of NetFlow data is configured to be sent to a specific address – in this case the IP address of the Hortonworks instance which is hosted on Microsoft Azure
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>System admin generates NetFlow data to intended IP Address</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>No NetFlow data generated or NetFlow data generated and targeting an incorrect IP address</li> </ul>

<b>Termination</b>	<ul style="list-style-type: none"> <li>• Main Flow: System Admin terminates Stream of NetFlow Data</li> <li>• Alternate Flow: System Admin terminates Stream of NetFlow Data</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>• Main Flow: No generation of NetFlow data</li> <li>• Alternate Flow: No generation of NetFlow data</li> </ul>

### 7.2.2. Apache Spark Application

➤ Description & Priority

The 'Apache Spark' application is responsible for the streaming and filtering of the incoming NetFlow data. This application must take the ingested data, manipulate it as required and seamlessly stream it to the SQL Server Database. A full breakdown of the application will be provided in Section 9 (Implementation) of this document.

➤ Use Case

Below is the Use Case for the Apache Spark Application

<b>Description/Scope</b>	Apache Spark application (written in Java) must successfully manipulate the ingested NetFlow data to extract the required IP and Port information. It must then stream this data to the SQL database.
<b>Pre-Condition</b>	System Admin must have full access to the Eclipse IDE in which the code is written and must have all required username password details for the SQL server instance.
<b>Activation</b>	System admin writes and test the code to ensure that it performs in line with expectation
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>• System Admin writes and tests Apache Spark application as required</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>• System Admin fails to write the Apache Spark application appropriately</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>• Main Flow: Functioning Apache Spark application is written and tested appropriately</li> <li>• Alternate Flow: Apache Spark application is not completed</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>• Main Flow: NetFlow data ingested is successfully filtered and streamed to database</li> <li>• Alternate Flow: NetFlow data cannot be streamed to the database and product will not function correctly</li> </ul>

### 7.2.3. Front-End Application

➤ Description & Priority

For the application functionality outlined in section 7.1 to function, the application itself needs to be written and tested in the correct manner. The application must be written to present the required look & feel as well as a high-level of usability. Critically, the application must be designed and implemented to read from the database to display the data in an accurate manner.

➤ Use Case

Below is the Use Case for the development of the Front-End Application

<b>Description/Scope</b>	A front-end web application must be developed in ASP.net MVC and the pages must be suitably designed for navigability, usability as well as look and feel. A combination of Bootstrap and JavaScript must be present
<b>Pre-Condition</b>	System Admin must have full access to the required development tools in order to test all aspects of the application (design and usability, database integration etc.)
<b>Activation</b>	System admin writes and test the code to ensure that it performs in line with expectation
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>System Admin writes and tests the ASP.Net application as required</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>System Admin fails to write the ASP.net application</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>Main Flow: Functioning front-end application is written and tested appropriately</li> <li>Alternate Flow: Front-End application is not completed</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>Main Flow: Application is ready and fit for use</li> <li>Alternate Flow: Application not available for use</li> </ul>

### 7.3 Functional Requirements – Data

This section will detail the project requirements from a data perspective. Within Section 7.2, the system requirements have been outlined and the core concept of NetFlow data streaming into the Apache Spark application has been covered. This section will describe the storage of data and its path to the end user.

#### 7.3.1. SQL Server – Relational Database

➤ Description & Priority

For its straightforward integration with ASP.net web applications, SQL server has been chosen to store all data and to interact with the front-end web application. Streaming data from the Apache Spark application will be injected into the SQL server database table as required. The ASP.net MVC application will be designed in a 'Database First' manner so that the models can be built using the required database tables from SQL server.

➤ Use Case

Below is the Use Case for SQL Server:

<b>Description/Scope</b>	An instance of SQL server will be used to store all data and this data will interact directly with the end-user's front-end application
<b>Pre-Condition</b>	A valid version of SQL server must be available to the system admin
<b>Activation</b>	System Admin creates the database along with all relevant database tables, views and Stored Procedures
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>System Admin tests configuration and can successfully access and store the requisite NetFlow data that is streamed from Apache Spark</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>System admin fails to create the SQL server database in the correct manner and data cannot be ingested/stored</li> </ul>



<b>Termination</b>	<ul style="list-style-type: none"> <li>• Main Flow: System admin creates and validates the SQL Server virtual machine within the Microsoft Azure portal</li> <li>• Alternate Flow: System admin fails to create the SQL server VM</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>• Main Flow: SQL server VM created and tested. NetFlow data stored in required table and accessible via the front-end application</li> <li>• Alternate Flow: SQL server instance unavailable</li> </ul>

**7.3.2. Relational Database Structure**

Once the above requirements have been met and the required data can populate a SQL server database, it is important that the database structure has been established and configured. The specific structure of the database tables will be outlined in detail within Section9: Implementation.

**7.4 Functional Requirements – Marketing Website**

As part of the project, a static marketing website will be built as a promotional tool and will contain all relevant product data and contact information. This site will be developed using bootstrap and will be hosted on Amazon’s S3 platform. There is a requirement for ‘Search Engine Optimization’ to be enabled and configured for the site to increase in profile.

**7.4.1. Site Hosting**

➤ Description & Priority

The marketing website will be hosted on the Amazon S3 platform. There is a requirement for the site to be configured and be made accessible to users on a 24/7 basis

➤ Use Case

Below is the Use Case for the hosting of the proposed marketing website

<b>Description/Scope</b>	A single page bootstrap web page must be hosted on the Amazon S3 platform. This site is purely developed for marketing purposes
<b>Pre-Condition</b>	Website must be built, configured, and populated with the appropriate content
<b>Activation</b>	Site is created and signed-off. S3 account is created and site is ready for upload/hosting
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>• System Admin configures the site on S3, commits the data and put the page into a ‘Production’ state</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>• System Admin fails to place the site in a ‘Production’ state</li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>• Main Flow: Site is live and available for admin testing</li> <li>• Alternate Flow: Site is not live and no web address is produced</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>• Main Flow: Site is ready to be accessed/tested by users</li> <li>• Alternate Flow: Site not accessible to users</li> </ul>

**7.4.2. Site Connectivity / Accessibility**

➤ Description & Priority

A key requirement for the site is accessibility. The users must have an appropriate web address so that they can connect to the site as required.

➤ Use Case

Below is the Use Case for user access to the marketing website

<b>Description/Scope</b>	User must be able to connect to the site in order to access the marketing information and product details
<b>Pre-Condition</b>	Site must be live and hosted via Amazon S3. User must be connected to the internet and have a valid address on which to connect
<b>Activation</b>	User enters the site address into their web browser
<b>Main Flow</b>	<ul style="list-style-type: none"> <li>• User attempts connection to web address</li> <li>• Site loads and presents user with product details etc.</li> </ul>
<b>Alternate Flow</b>	<ul style="list-style-type: none"> <li>• User attempts connection to web address</li> <li>• Site fails to load                             <ul style="list-style-type: none"> <li>○ Site unavailable</li> <li>○ Users internet connection is down</li> </ul> </li> </ul>
<b>Termination</b>	<ul style="list-style-type: none"> <li>• Main Flow: User connects successfully</li> <li>• Alternate Flow: Connection fails – investigation required</li> </ul>
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>• Main Flow: User has access to site</li> <li>• Alternate Flow: Site unavailable</li> </ul>

### 7.5 Non-Functional Requirements

Specifies any other non-functional attributes required by the system. Examples are provided below.

#### 7.5.1. Performance/Response Time Required

The application must be responsive to promote usability and returning users. If there was a significant lag in response to user interaction this could discourage users from active use. It is important that the application is correctly configured and that the underlying hardware and software is fit for purpose and appropriately designed.

#### 7.5.2. Availability Requirement

The application must be available to the individuals required to use it. This should include the system admin, required testers, and a facility should be available to demonstrate the application as required. The application is running on a local computing environment at present so availability is limited to people with access to same.

#### 7.5.3. Accessibility Requirement

The application should be simple, intuitive, and highly navigable with a focus on usability. This will include all key functionality i.e. sign-in, moving from page to page as well as the overall look and feel. The application should have a fluid and responsive manner to deliver a high level of service to all users. The overall design and cosmetics of the site should also be simple yet effective to encourage repeat usage and industry recommendations.

#### **7.5.4. Security Requirement**

The application must enforce a robust level of security with a focus on information segregation and access levels. There is no requirement to store personal data and the information being fed to the site is irrelevant to the outside world. Regardless, it would be remiss to build an application without the appropriate level of security. Considerations will be made around preventing access to the underlying application and protecting the data from malicious threats such as SQL injection etc.

#### **7.5.5. Maintainability Requirement**

There may be a requirement for additional support roles (app developer, database admin etc.) to maintain and evolve functionality and service levels. For the time being there will be a facility for a user to contract a support address in which they can log queries & issues and get an appropriate response and subsequent resolution.

#### **7.5.6. Portability Requirement**

The application is not currently built to be portable - it was not designed for this capability. However, Section 11 outlines the plan for future portability/mobility through a cloud based implementation.

#### **7.5.7. Reusability Requirement**

It is critical that the application is designed and implemented in an intuitive and user-friendly manner. This relates to the look & feel, functionality, responsiveness, and overall navigability. It is important that users get an immediate feeling of comfort and that the application meets their needs in a straightforward fashion. Appropriate, timely and coherent notifications that alert the user to issues will also be a key motivation for active re-use and recommendation.

#### **7.5.8. Error Checking Requirement**

This will be of critical importance in terms of the sign-in process. The application must instantly recognize that only specific information in specific formats will be accepted in required fields e.g. when requesting an email address, it must be entered in the format 'person@domainname'. This will be important in ensuring that the right people get access to the right data.

#### **7.5.9. Concurrency Requirement**

Any successful application has a fundamental ability to accept multiple users at any one time. This maximizes potential profits and grows usage. The application must be able to accept multiple concurrent log-ins without user experience being degraded. Without concurrency, the application could be rendered unfit for purpose.

## 8. Solution Topology & Architecture

This project was designed and implemented using the key system components, products and technologies that were detailed within Section 3 of this document. Combined, these tools facilitated the generation, computation, storage, distribution, and visualization of all relevant network activity. The process flow i.e. event, compute, data management and presentation has been graphically mapped out in Fig 10 below:

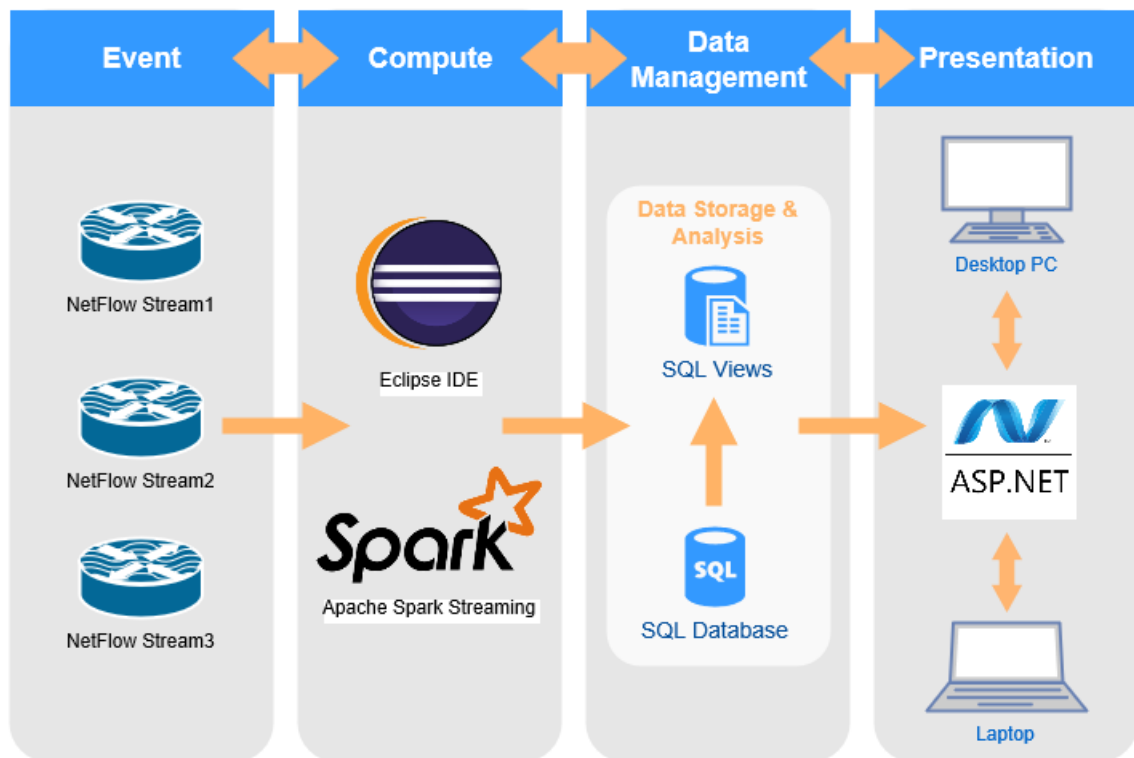


Fig10: Solution Architecture and Data Flow / Interaction

It should be noted (again) that for the purposes of this project the NetFlow data will be generated using an open source software tool developed by Paessler. In Fig10 above, Cisco NetFlow Switches have been displayed for this function. Ultimately, this is the way the system should/would operate in a true enterprise environment. However, due to cost restrictions, such a live implementation would be outside of the resources available for this project.

The Paessler NetFlow Generator has been configured to send its NetFlow data to a specific IP and Port that represents the Apache Spark Streaming Application. This application has been developed and resides within the Eclipse Integrated Development Environment (IDE). This application – developed in Java – ingests and subsequently filters down the required information. From this point the Apache Spark application identifies the data that is to be transferred into the database i.e. potential threat related data. This data is automatically inserted into a SQL server database (dbo.SOSThreatAnalytics). The finalized data (tables, views etc.) is then accessible to the user via a customer build ASP.net MVC application (SOS Threat Analytics). This implementation of all the above is detailed in Section 9.

## 9. Project Implementation

This section will detail all aspects of the actual implementation of the previously defined Solution architecture, technology sets and required functionality that encompasses SOS Threat Analytics. This implementation will be broken out into 4 key sections and their constituent parts:

Section	Components/Details
<b>Programming Language Environment</b>	Eclipse IDE
	Apache Spark Streaming
	Database Connectivity
<b>Database</b>	Design & Implementation
	Back-End Integration
	Front-End Integration
<b>SOS Threat Analytics Application</b>	Navigation, Look & Feel
	Back-End Integration
	Functionality
<b>Marketing Website</b>	Design & Implementation
	Features & Usability
	Hosting & SEO

Fig11: Breakdown of Implementation Steps

### 9.1 Programming Language Environment

In order to ingest, stream and filter the NetFlow data, a suitable programming language environment/engine was required. Apache Spark was chosen as the preferred programming engine due to its ability to seamlessly handle, stream and perform operations on, incoming data packets. Another driver for this choice was to think longer term and in relation to future development. 'Spark was built to handle large amounts of data and in a production environment, the daily NetFlow activity would be extremely large. The chosen engine for ingestion, filtering etc. would need to be capable of managing such activity. "Apache Spark is an open-source engine developed specifically for handling large-scale data processing and analytics. Spark offers the ability to access data in a variety of sources, including Hadoop Distributed File System (HDFS), OpenStack Swift, Amazon S3 and Cassandra." (Webopedia, 2017). An Apache Spark application/context can be programmed in a multitude of different languages including Java, Scala, and Python. Due to previous experience and familiarity, SOS Threat Analytics utilises Java for the construction and operation of the Spark application. The chosen development environment (IDE) that would host the application was Eclipse.

In setting up the environment, several installations (plug-ins) were required to appropriately build an Apache Spark Context within Eclipse e.g. Apache Maven. Once everything was configured to an optimum level, the required classes were created. The final list of classes for the SOS Threat Analytics application were as follows:

➤ [JavaCustomReceiver.java](#)

This was the main Java class and the one into which all other required data structures and classes were instantiated. This class contains the main method and all core methods that allows the overall solution to function. The majority of code extracts that will be covered in this section will be taken directly from this class.

➤ [ApacheSpark.java](#)

Creates the Apache Spark Streaming Context which will be instantiated by JavaCustomerReceiver.java. This class sets out the functions that can be accessed through Apache Spark e.g. parallelize, filter, flatmap etc. These operations can then be performed on the ingested NetFlow data to manipulate it into the structure/elements that are required.

➤ [NetFlowV5Message.java](#)

This class create the a standard NetFlow data structure in the form of an Object. In Section4 of this document the structure of a NetFlow packet is described. This class creates all of the required variables to align with that packet structure and contains the relevant getters/setters for each one. A NetFlowV5Message will be create for each NetFlow packet that is ingested and subsequently injected into the SQL Server database

➤ [IOCDetail.java](#)

A critical aspect of the JavaCustomReceiver.java class is to read in a list of known malicious IPS/Ports. This list is then compared against the incoming packets in order to identify potential threats. This list must be stored for comparison and the IOCDetail.java class creates the object in which these details will be stored. IOC stands for Indicators of Compromise and these stored IOC's are read from the database each time the program is run.

➤ [IOCEventType.java](#)

This is less of an important class and was used primarily to assist in the translation of protocol types. On reading the NetFlow packet the protocol appeared simply as a number e.g. 1 = ICMP, 17 = UDP. The hash table allows for us to enter the protocol value into the database as a name (ICMP, UDP) rather than as a meaningless number (from a user's perspective).

*\*All the above classes will be included as part of the Project Code Submission 14/05/2017.*

In term of initial development, the first item to address was the initiation of the 'Spark streaming context itself. The below code extract (Fig12), outlines the instantiation of the streaming context:

```
//Creating an instance of an Apache Spark Context
SparkConf sparkConf = new SparkConf();
sparkConf.setAppName("Hello Spark");
sparkConf.setMaster("local[*]");

// Create the context with a 10 second batch size
JavaStreamingContext ssc = new JavaStreamingContext(sparkConf, new org.apache.spark.streaming.Duration(10000));

ssc.start();
ssc.awaitTermination();
```

Fig12: Creating an instance of an Apache Spark Streaming Context

The NetFlow Generator has at this point been configured to collect all information from the Localhost (127.0.0.1) over port 9996 (this detail has been embedded within the 'Spark context class). For the 'Spark streaming context to begin to ingest this data, we must create a local socket on port 9996 that will listen for this traffic. The below code (Fig13) was used to generate the socket and print a message within Eclipse to verify that it is listening and awaiting incoming traffic.

```
//1. creating a server socket, parameter is local port number (9996)
sock = new DatagramSocket(port);

//buffer to receive incoming data
byte[] buffer = new byte[65536];
DatagramPacket incoming = new DatagramPacket(buffer, buffer.length);

//2. Wait for an incoming data
System.out.print("Server socket created. Waiting for incoming data... on port " +port);
```

Fig13: Creating the Server Socket which will listed over Port 9996 (previously defined port variable)

The below shows an output from Eclipse when the application is run, confirming that the socket is opened and awaiting data:

```
17/05/06 15:27:00 INFO ReceiverSupervisorImpl: Starting receiver 0
17/05/06 15:27:00 INFO ReceiverSupervisorImpl: Called receiver 0 onStart
17/05/06 15:27:00 INFO ReceiverSupervisorImpl: Waiting for receiver to be stopped
Server socket created. Waiting for incoming data... on port 9996
```

Fig14: Application in a waiting state i.e. awaiting packets from the NetFlow Generator

Once the socket starts to receive data, we need specific code to parse that data. Looking back at our NetFlow packet structure we need to handle both the 'Header' and 'Flow Data' to ensure that everything is captured prior to any filtering operations. The below loop (Fig15) is in operation once the streaming context is listening. On receipt of data it begins to read in each individual section via its byte-count/range within the packet.

```

while(true)
{
    sock.receive(incoming);
    byte[] data = incoming.getData();

    byte[] header = Arrays.copyOfRange(data, 0, 2);
    int version = unsignedByteToInt(header, 0,header.length);
    byte payload[] = Arrays.copyOfRange(data,2,4);
    int recordCount = unsignedByteToInt(payload, 0,payload.length);
    byte sysuptime[] = Arrays.copyOfRange(data,4,8);
    byte epoch[] = Arrays.copyOfRange(data,8,12);
    byte nano[] = Arrays.copyOfRange(data,12,16);
    byte flow[] = Arrays.copyOfRange(data,16,20);
    byte engineType[] = Arrays.copyOfRange(data,20,21);
    byte engineId[] = Arrays.copyOfRange(data,21,22);
    byte sampleInfo[] = Arrays.copyOfRange(data,22,24);

    System.out.println("Starting Stream NetFlow Parsing .... ");
    System.out.println("Version "+version);
    System.out.println("Count "+recordCount);
    System.out.println("system up time "+unsignedIntToLong(sysuptime));
    System.out.println("Epoch "+unsignedIntToLong(epoch));
    System.out.println("Nano "+unsignedIntToLong(nano));
    System.out.println("Flow "+unsignedIntToLong(flow));
    System.out.println("Engine Type "+unsignedByteToInt(engineType, 0,engineType.length));
    System.out.println("Engine Id "+unsignedByteToInt(engineId, 0,engineId.length));
    System.out.println("Sample SD "+unsignedByteToInt(sampleInfo, 0,sampleInfo.length));
}

```

**Fig15: Parsing the NetFlow Header Information & Printing the Details to the Eclipse Console**

From the earlier Wireshark capture we noted that each element within the packet took up a certain number of bytes. Therefore, we need to capture each specific element (header, payload, engineType etc.) according to its particular byte range (size) – this is evidenced in Fig15. For the avoidance of doubt, each element within the header is printed to the console. This confirms that the data is being parsed as expected for each incoming packet. Similarly, the next section of the packet (and the one we are most interested in), the 'Flow Data', must now also be parsed. As this detail will be added to our NetFlowV5Message object the byte position (start and end points) take on increased importance. Fig16 demonstrates the code required to parse and capture all elements of the Flow Data:

```

NetflowV5Message msg = new NetflowV5Message();

baseOffsetCount = baseOffsetCount*1;
startPos = baseOffsetCount;
endPos = baseOffsetCount+4;
byte ipSrcaddr[] = Arrays.copyOfRange(data, startPos, endPos);
startPos = endPos;
endPos = endPos+4;
byte ipDstaddr[] = Arrays.copyOfRange(data, startPos, endPos);
startPos = endPos;
endPos = endPos+4;
byte nextHop[] = Arrays.copyOfRange(data, startPos, endPos);
startPos = endPos;
endPos = endPos+2;
byte inboundSNMPIndex[] = Arrays.copyOfRange(data, startPos, endPos);
startPos = endPos;
endPos = endPos+2;
byte outboundSNMPIndex[] = Arrays.copyOfRange(data, startPos, endPos);
startPos = endPos;
endPos = endPos+4;
byte packetCount[] = Arrays.copyOfRange(data, startPos, endPos);
startPos = endPos;
endPos = endPos+4;
byte byteCount[] = Arrays.copyOfRange(data, startPos, endPos);
startPos = endPos;
endPos = endPos+4;
byte startFlowTime[] = Arrays.copyOfRange(data, startPos, endPos);
startPos = endPos;
endPos = endPos+4;
byte endFlowTime[] = Arrays.copyOfRange(data, startPos, endPos);
startPos = endPos;
endPos = endPos+2;

```

**Fig16: Code used to parse the 'Flow Data' of the NetFlow packet**



Fig17 below, then demonstrates the code used to add the required elements from the 'Flow Data' into the NetFlowV5Message' Object.

```

msg.setDstAddress(ntoa(unsignedIntToLong(ipDstaddr)));
msg.setSrcAddress(ntoa(unsignedIntToLong(ipSrcaddr)));
msg.setDstPort(new BigInteger(dstPort).intValue());
msg.setSrcPort(new BigInteger(srcPort).intValue());

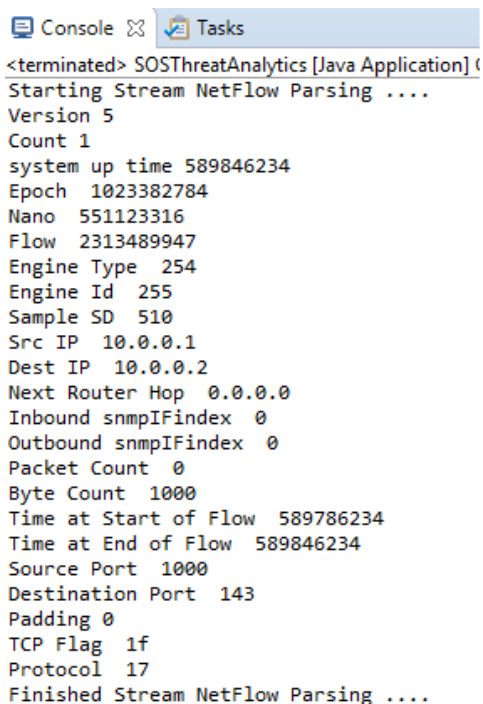
msg.setNextRouterHop(ntoa(unsignedIntToLong(nextHop)));
msg.setByteCount(unsignedIntToLong(byteCount));
msg.setPacketCount(unsignedIntToLong(packetCount));
// msg.setStartFlowTime(new Date(unsignedIntToLong(startFlowTime)));
// msg.setEndFlowTime(new Date(unsignedIntToLong(endFlowTime)));
msg.setPacketTime(new Date(unsignedIntToLong(epoch)*1000));
msg.setTcpFlag(Integer.toHexString(unsignedByteToInt(tcpFlag, 0,tcpFlag.length)));
msg.setProtocol(unsignedByteToInt(protocol, 0,protocol.length));

store(msg);
System.out.println("Src IP "+ntoa(unsignedIntToLong(ipSrcaddr)));
System.out.println("Dest IP "+ntoa(unsignedIntToLong(ipDstaddr)));
System.out.println("Next Router Hop "+ntoa(unsignedIntToLong(nextHop)));
System.out.println("Inbound snmpIFindex "+unsignedByteToInt(inboundSNMPIFindex, 0,inboundSNMPIFindex.length));
System.out.println("Outbound snmpIFindex "+unsignedByteToInt(outboundSNMPIFindex, 0,outboundSNMPIFindex.length));
System.out.println("Packet Count "+unsignedIntToLong(packetCount));
System.out.println("Byte Count "+unsignedIntToLong(byteCount));
System.out.println("Time at Start of Flow "+unsignedIntToLong(startFlowTime));
System.out.println("Time at End of Flow "+unsignedIntToLong(endFlowTime));
System.out.println("Source Port "+ new BigInteger(srcPort).intValue());
System.out.println("Destination Port "+ new BigInteger(dstPort).intValue());
System.out.println("Padding "+unsignedByteToInt(padding, 0,padding.length));
System.out.println("TCP Flag "+Integer.toHexString(unsignedByteToInt(tcpFlag, 0,tcpFlag.length)));
System.out.println("Protocol "+unsignedByteToInt(protocol, 0,protocol.length));

```

**Fig17: Adding the required packet details to the object and printing the parsed information**

As you can see from this code extract, the finalized data is again printed to the console to provide verification that the packet was parsed and ingested as expected. Below is a sample of this output (Packet Header and Flow Data) within the Eclipse IDE console:



The screenshot shows the Eclipse IDE console with the 'Console' tab selected. The output text is as follows:

```

<terminated> SOSThreatAnalytics [Java Application] (
Starting Stream NetFlow Parsing ....
Version 5
Count 1
system up time 589846234
Epoch 1023382784
Nano 551123316
Flow 2313489947
Engine Type 254
Engine Id 255
Sample SD 510
Src IP 10.0.0.1
Dest IP 10.0.0.2
Next Router Hop 0.0.0.0
Inbound snmpIFindex 0
Outbound snmpIFindex 0
Packet Count 0
Byte Count 1000
Time at Start of Flow 589786234
Time at End of Flow 589846234
Source Port 1000
Destination Port 143
Padding 0
TCP Flag 1f
Protocol 17
Finished Stream NetFlow Parsing ....

```

**Fig18: Console showing a fully captured NetFlow packet**

What the above demonstrates is the successful ingestion and interpretation of a simulated Cisco NetFlow packet. In many ways, this part of the implementation served as a 'Proof of Concept' for the project as a whole. Successfully dissecting NetFlow packets in real time evidenced that the core concept of the project was plausible (to this point it has only been a high-level design) and that work from this point could continue toward the end goal. The next phase was to connect the development environment to a database and verify that this data could then be moved between the two.

The first step was to configure a direct connection from the Eclipse IDE to SQL server. This was managed using an open source 'jdbc' connector that was added to the javaCustomerReceiver.java class as shown in Fig 19 below:

```
//create a connection to the SOSThreatAnalytics Database
private static Connection getDBConnection() {

    Connection dbConnection = null;
    String dbURL = null;
    try {

        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver").newInstance();
        dbURL = "jdbc:sqlserver://localhost:1433;DatabaseName=SOSThreatAnalytics;user=sa;password=Holly1812";

    } catch (Exception e) {

        System.out.println(e.getMessage());

    }

    try {
        dbConnection = DriverManager.getConnection(dbURL);

        return dbConnection;
    } catch (SQLException e) {

        System.out.println(e.getMessage());

    }

    return dbConnection;
}
```

**Fig19: Establishing a Database Connection**

The 'jdbc' connector successfully created a level of integration between the programming environment and the SQL server database. This meant that an established route had been created for the direct injection of data from Eclipse to the SOS Threat Analytics database. It is important to note however that not all NetFlow packets should be imported into the database. The only packets that are required are ones that contain potential threats. This added a layer of complexity as in order to identify these suspicious packets, a list of known threats would be required against which all incoming packets could be compared. The database structure will be discussed in detail in Section9.2 so for this section it is safe to assume that this has been completed and a table(s) of threats has been loaded.

From a programming point of view the key is to have the Apache Spark streaming context read in the list of threats as soon as the application starts. This was done through the previously established database connection and a subsequent embedded 'SELECT' statement as shown in Fig20 below:

```

//read current IOC Listings (Identified Threats) from SOS Threat Analytics Database
private static List<IOCDetail> getIOCList()
{
    Connection conn = null;
    //Create Array List for current IOC Listings called badIpList (using IOCDetail.java as the object type)
    List<IOCDetail> badIpList = new ArrayList<IOCDetail>();
    try {
        //Use .jdbc SQL connector to access the database (username and password included)
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver").newInstance();
        String dbURL = "jdbc:sqlserver://localhost:1433;DatabaseName=SOSThreatAnalytics;user=sa;password=Holly1812";
        conn = DriverManager.getConnection(dbURL);
        if (conn != null) {
            Statement statement = conn.createStatement();

            //using a successful connection, we can now run a query from the IOC_Listing table
            ResultSet result = statement.executeQuery("select * from IOC_Listing");
            while(result.next()) {
                //An IOCDetail object is created for each valid listing
                IOCDetail tempDetail = new IOCDetail();
                tempDetail.setId(result.getInt("ID"));
                tempDetail.setIocDetail(result.getString("IOC_DETAIL").trim());
                tempDetail.setIocType(result.getInt("IOC_TYPE"));
                //we add each valid listing to the badIP List array
                badIpList.add(tempDetail);
            }
            result.close();
            statement.close();
        }
    }
}

```

**Fig20: Code Extract for Reading in and Storing the Current list of Known malicious IPs and Ports**

The above method connects to the SOS Threat Analytics database and reads a list of 'Indicators of Compromise' from the IOC\_Listings table (see section 9.2). This table provides a list of known malicious IP's and Ports against which the incoming NetFlow packets will be compared. The theory is that if a match was to be found, this packet would be exported to the database and visualised for the end user – ultimately sending an SMS alert that there is a problem on the network. The data is stored in an 'IOCDetail' object which was established in one of the java classes outlined at the start of this section.

Since we are only looking to match on IP and/or Port, there is a requirement to filter and extract the IP and Port information from the IOCDetail object. This was one of the key reasons Apache Spark streaming was chosen as the programming engine - filtering operations are a readily available function within 'Spark. The following code (Fig21) was written to filter what was taken in from the database so that the focus is purely on the IP and Port for that IOC.

```

//Grab the list of known bad IPs and ports from the database (SOS Threat Analytics) using the getIOCList() method
List<IOCDetail> badIps = getIOCList();

/* Now Grab only the IP's from the IOC_Listing Table using a filter operation
 * These IP's will be stored in an ArrayList called filterBadIP list which will be compared against the incoming
 * NetFlow packet*/
List<IOCDetail> filteredBadIpList = badIps.stream()
    .filter(individualBadIP -> individualBadIP.getIocType() == IOCEventType.BAD_IP.getCode())
    .collect(Collectors.toList());

/*Now Grab only the Ports from the IOC_Listing Table using a filter operation
These IP's will be stored in an ArrayList called filterBadIP list which will be compared against the incoming
NetFlow packet*/
List<IOCDetail> filteredBadPortList = badIps.stream()
    .filter(individualBadIP -> individualBadIP.getIocType() == IOCEventType.BAD_PORT.getCode())
    .collect(Collectors.toList());

```

**Fig21: Apache Spark filtering operations to identify the IP & Port from the IOCDetail object**

This code initially calls the method ( getIOCList() ) which we created previously in pulling the list of IOCs' from the database table. The filter operations then extract the IP and Port, placing

them into two separate Array Lists: filterBadIPList and FilterBadPortList respectively. The incoming NetFlow packets IP and Port will then be compared against these two lists. If a comparison is found with either list – the packet in question will be added to the database (dbo.NETFLOW\_PKT table). To enable the comparison, we must now filter the requisite details from an incoming packet – this is done using the coded examples below. First is the code to match by IP:

```

/*Examine each individual NetFlow packet stored in a 'NetflowV5Message' object
 * and subsequently stream and parse the individual elements of the message/object.
 * This code will perform the filter for Destination IP and Port which are the basic
 * criteria for threat identification within SOS Threat Analytics
 */
JavaReceiverInputStream<NetflowV5Message> lines = ssc.receiverStream(
new JavaCustomReceiver(args[0], Integer.parseInt(args[1])));
JavaOutputStream<NetflowV5Message> words = lines.filter( x ->
{
    // Check if the packet has a Bad IP (from filtered IOC_Listing)
    Optional<IOCDetail> matchingObjects = filteredBadIPList.stream().filter(p -> p.getIocDetail().equals(x.getDstAddress())).findFirst();

    IOCDetail matchingIOC = matchingObjects.orElse(null);

    /* If an IP match is found print a message confirming this.
     * Also state that this message is a potential threat and it will be moved
     * to the database
     */
    if (matchingIOC != null)
    {
        System.out.println("Bad IP Detected, now collating packet to database");
        x.setIocListingId(matchingIOC.getId());
        return true;
    }
    else
    {
        System.out.println("No Bad IP Detected, now running bad ports check!!!");
    }
}

```

**Fig22: Code Extract to filter the IP from an incoming NetFlow packet and compare it against the known list**

Below is the code used to match by Port:

```

// Check if the packet has a Bad Port (from filtered IOC_Listing)
Optional<IOCDetail> matchingBadPortsObjects = filteredBadPortList.stream().filter(p -> Integer.valueOf(p.getIocDetail()).intValue()
== (x.getDstPort().intValue())).findFirst();
matchingIOC= matchingBadPortsObjects.orElse(null);

/* If a port match is found print a message confirming this.
 * Also state that this message is a potential threat and it will be moved
 * to the database
 */
if (matchingIOC != null)
{
    System.out.println("Bad Port Detected, now collating packet to database");

    x.setIocListingId(matchingIOC.getId());
    return true;
}
else
{
    System.out.println("There is no Indicator of Compromise detected!!!");
    return false;
}

```

**Fig23: Code used to filter the Port from an incoming NetFlow packet and compare it against the known list**

If either match is found the 'insertRecordIntoTable' method is called. This effectively takes all information from the 'Flow Data' of the NetFlow packet and inserts it into the database table (NETFLOW\_PKT). See the code extract below:

```
private static void insertRecordIntoTable(NetFlowV5Message message) {
    Connection dbConnection = null;
    PreparedStatement preparedStatement = null;

    String insertTableSQL = "INSERT INTO NETFLOW_PKT (ID, SRCADDR, SRCPORT, DSTADDR, DSTPORT, NEXT_ROUTER_HOP, PACKET_COUNT, BYTE_COUNT, PACKET_TIME, TCP_FLAG, PROTOCOL, IOC_LISTING_ID)"
        + " VALUES (NEXT VALUE FOR netflow_seq, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

    try {
        dbConnection = getDBConnection();
        preparedStatement = dbConnection.prepareStatement(insertTableSQL);

        preparedStatement.setString(1, message.getSrcAddress());
        preparedStatement.setInt(2, message.getSrcPort());
        preparedStatement.setString(3, message.getDstAddress());
        preparedStatement.setInt(4, message.getDstPort());
        preparedStatement.setString(5, message.getNextRouterHop());
        preparedStatement.setLong(6, message.getPacketCount());
        preparedStatement.setLong(7, message.getByteCount());
        preparedStatement.setDate(8, new java.sql.Date(message.getPacketTime().getTime()));
        preparedStatement.setString(9, message.getTcpFlag());
        preparedStatement.setString(10, returnProtocol(message.getProtocol()));
        preparedStatement.setInt(11, message.getIoCListingId());

        // execute insert SQL statement
        preparedStatement.executeUpdate();

        System.out.println("Record is inserted into DBUSER table!");
    }
}
```

**Fig24: Code Extract for the insertion of a record into the NetFlow table**

A message will be printed to the console confirming that either a Bad IP or Port has been found within the NetFlow packet and that this packet will be added to the database:

```
<terminated> SOSThreatAnalytics [Java Application] C:\Java\jdk1.8.0_121\bin\javaw.exe (6 May 2017, 15:43:33)
17/05/06 15:44:00 INFO BlockManager: Found block input-0-1494081831400 locally
No Bad IP Detected, now running bad ports check!!
Bad Port Detected, now collating packet to database
17/05/06 15:44:00 INFO Executor: Finished task 0.0 in stage 1.0 (TID 1). 1490 bytes result sent to driver
17/05/06 15:44:00 INFO TaskSetManager: Finished task 0.0 in stage 1.0 (TID 1) in 36 ms on localhost (executor driver) (1/1)
17/05/06 15:44:00 INFO TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
17/05/06 15:44:00 INFO DAGScheduler: ResultStage 1 (collect at JavaCustomReceiver.java:140) finished in 0.041 s
17/05/06 15:44:00 INFO DAGScheduler: Job 3 finished: collect at JavaCustomReceiver.java:140, took 0.061796 s
Record is inserted into DBUSER table!
```

**Fig25: Console Output confirming a bad port detection and a successful database insertion**

As this example has focused on known malicious port, the code has first failed to find a match by IP (which was expected) but it has picked up the bad Port and added this record to the database (as per the console output).

## 9.2 Database

SOS Threat Analytics is primarily a data-driven product. As such, the importance of a well-designed and logically structured database cannot be understated. SQL Server was chosen as the 'Relational Database Management System' for the project. A database called 'SOSThreatAnalytics' was created and the final implementation of this database included multiple tables, a consolidated database 'View' and a detailed 'Stored Procedure' that was utilised for the operation of the SMS alerting feature. The full list of tables utilised within the SQL Server database were as follows:

- IOC\_URL Details
- IOC\_Listing
- IOC\_DETAILS\_XREF
- IOC\_Ref
- NETFLOW\_PKT
- Device\_Details
- Contacts
- SMS\_TRACKER

9.2.1. Database ERD

Below is the final design (Entity Relationship Diagram) for the SOS Threat Analytics database. The individual tables and their relationships have been included within the diagram. One item of note is the fact that the 'DEVICE\_DETAILS' and 'CONTACTS' tables are not connected to any other table. These tables are built for static data and will display the network device details and key contact information respectively. The function of each table will be detailed below.

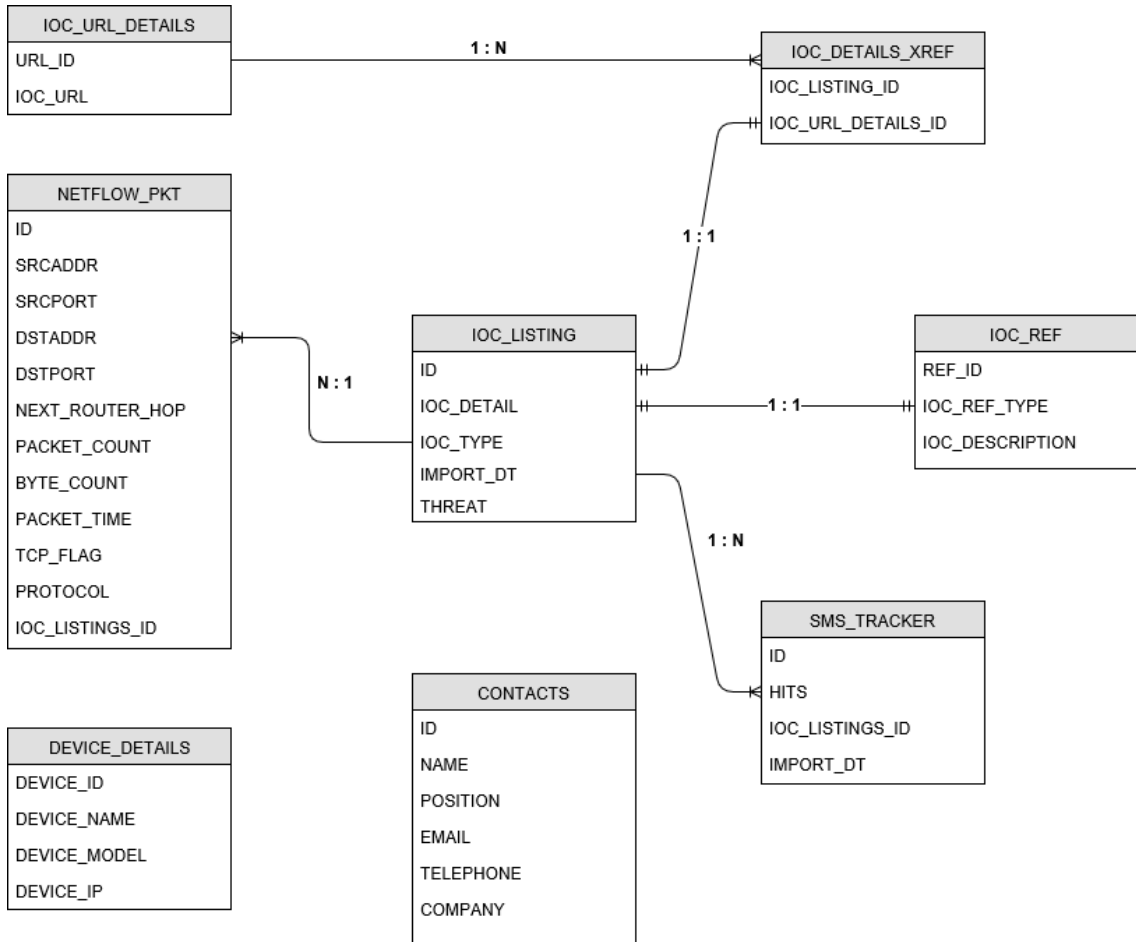


Fig26: SOS Threat Analytics Database Design

➤ NETFLOW\_PKT

This table, alongside IOC\_LISTING are the core elements of the database and are used for the final 'view' which is integrated into the front-end application. NETFLOW\_PKT however taken on even more importance – as shown in 'Section 9.1, Fig24', this is the table that a potentially threatening NetFlow packet gets inserted into. This back-end integration with Apache Spark enables the capture and storage of the 'threat' data. This table is an exact replica of the 'Flow Data' element of the NetFlow packet structure outlined in detail within Section4. There exists a many to one relationship with the IOC\_LISTING table, in that many NetFlow packets will require an individual IOC\_Listing.

### ➤ IOC\_LISTING

The direct relationship with the NETFLOW\_PKT table highlights the importance of the IOC\_LISTING table. However, this table also ties together crucial information relating to the description of each potential threat (Indicator of Compromise) through its relationships with other tables i.e. IOC\_DETAILS\_XREF and IOC\_REF. This table also contains the name of the specific threat which plays a role in the view presented to the user.

### ➤ IOC\_REF

Direct connection to the IOC\_LISTINGS table to clarify the type of threat. IOC\_LISTINGS uses this table to translate its IOC\_TYPE from a number to the actual threat type i.e. IP Address or Port.

### ➤ IOC\_DETAILS\_XREF

This table acts as a bridge and assists once more in the definition and detail of the potential threat (Indicator of Compromise). For each IOC\_Listing, there will exist a reference to this table. Furthermore, each of the IOC's will have a link to a specific URL (listed within the IOC\_URL\_DETAILS table). Many IOC's can utilise the same URL link i.e. multiple IOC's can execute the same threat e.g. email extraction.

### ➤ IOC\_DETAILS\_XREF

A table of links to documents and websites that provide information and methods of remediation for specific threat. Each IOC\_Listing requires access to an 'id' from this table. This is handled through the previously mentioned IOC\_DETAILS\_XREF table. It should be noted that these URL's are crucial to the end user experience with the SOS Threat Analytics application.

### ➤ DEVICE\_DETAILS

This table simply contains a list of devices that are being monitored. This is a single table that stands alone and can be directly accessed by the end user through the application.

### ➤ CONTACTS

This table contains a list of vendor and client contacts. This is a single table that stands alone and can be directly accessed by the end user through the application.

### ➤ SMS\_TRACKER

This table actively captures the number of hits that each IOC has received at a particular point in time. This detail is used to decide when to send an SMS alert. The penultimate number of hits is subtracted from the most recent result and if this value is over 5, an SMS is actively sent to the end user. This is managed via a stored procedure that polls the database for hit counts every few minutes. This stored procedure contains the client contact details to send the SMS.

### 9.2.2. Front-End Application Integration

With the database design complete and data actively being stored, the next step was to integrate the data with the SOS Threat Analytics application to make it accessible to the end user. As opposed to attempting to create a complex structure within Visual Studio, a database view was created in SQL server: VW\_NETFLOW\_IOC. The SQL used to create this view is shown below:

```
SELECT ISNULL(MAX(A.ID), - 1) AS ID, MAX(B.Threat) AS THREAT, MAX(A.SRCADDR) AS SRCADDR, MAX(A.SRCPORT) AS SRCPORT,
MAX(A.DSTADDR) AS DSTADDR, MAX(A.DSTPORT) AS DSTPORT, MAX(A.NEXT_ROUTER_HOP) AS NEXT_ROUTER_HOP,
MAX(A.PACKET_COUNT) AS PACKET_COUNT, MAX(A.BYTE_COUNT) AS BYTE_COUNT, MAX(A.PACKET_TIME) AS PACKET_TIME,
MAX(A.TCP_FLAG) AS TCP_FLAG, MAX(A.PROTOCOL) AS PROTOCOL, A.IOC_LISTING_ID, MAX(B.IOC_URL) AS IOC_URL,
NULLIF (COUNT(*), - 1) AS HITS
FROM dbo.NETFLOW_PKT AS A INNER JOIN
dbo.Vw_IOC_LISTING AS B ON A.IOC_LISTING_ID = B.ID
GROUP BY A.IOC_LISTING_ID
```

Fig27: SQL to create the VW\_NETFLOW\_IOC database view

The above SQL statement creates an inner join between the NETFLOW\_PKT and IOC\_LISTING tables mentioned previously. This meant that all required and relevant data related to a particular IOC could be captured in a single table. The table structure within SQL appears as follows:

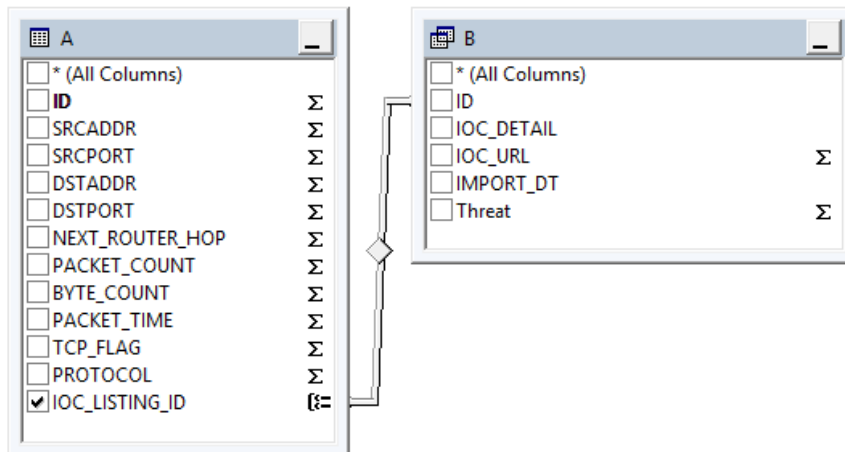


Fig28: VW\_NETFLOW\_IOC view (A=NETFLOW\_PKT & B=IOC\_LISTING)

This view is integrated into SOS Threat Analytics application and appears to the end user within the 'Indicators of Compromise' page. This is explained in detail in 'Section 9.3'.



### 9.3 SOS Threat Analytics Application

The front-end web application is the users visual tool for accessing the underlying NetFlow data and potential threats that have been identified. By logging into this application, users can gain access to in-depth detail on the most recent malicious IP and port 'Hit's on their network as well as security related news feeds from sources such as RSS feeds and Twitter. Furthermore, they can access a full list of their network devices that are licenses for monitoring as well as a detailed table of key support contacts from the Vendor and Client side. Section 11 of this document provides visualizations of all the above detail to deliver some context around the user experience within the SOS Threat Analytics application.

The application itself has been developed entirely within Microsoft Visual Studio 2015, utilizing their ASP.net MVC (Model – View – Controller) Framework. A web application was chosen for the front-end due to its industry wide acceptance, as well as strategic thinking around future development (Cloud) and enhanced accessibility, portability etc. The specific choice of ASP.net was due to its ease of integration with SQL server from a data access perspective and it's out of the box integration with Microsoft Azure which could be invoked at some point in the future. Previous familiarity and development experience with the platform also played a factor.

Within ASP.net there are two key types of application development: Code First and Database First. SOS Threat Analytics was developed using the 'Database First' approach, meaning that all the detail/structure set out in 'Section 9.2' above, was completed in advance of the application piece. The pages that link back to the database utilized specific database tables for the creation of their 'models' within the MVC framework. The most prominent and most important example of this is the 'Indicators of Compromise' page which is responsible for displaying the list of current potential threats in the environment. This page used the database view (table) 'dbo.VW\_NETFLOW\_IOC' to access and subsequently display its required data. Below is the model of this 'table' imported into Visual Studio:

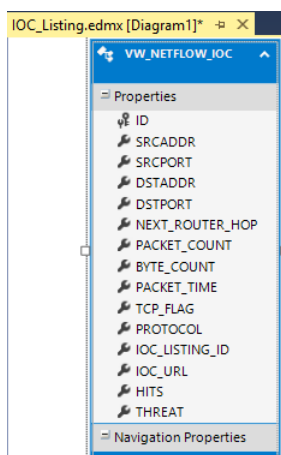


Fig29: View Table for IOC's successfully imported as Model within Visual Studio

A 'controller' and 'view' were then created for this model with the following code extract utilized to present the data to the end user:

```

Index.cshtml  IOC_Listing.edmx [Diagram1]*
19 <table class="table-bordered table-striped">
20 <tr>
21 <th>
22 @Html.DisplayNameFor(model => model.THREAT)
23 </th>
24 <th>
25 @Html.DisplayNameFor(model => model.SRCADDR)
26 </th>
27 <!--<th>
28 @Html.DisplayNameFor(model => model.SRCPORT)
29 </th-->
30 <th>
31 @Html.DisplayNameFor(model => model.DSTADDR)
32 </th>
33 <th>
34 @Html.DisplayNameFor(model => model.DSTPORT)
35 </th>
36 <!--<th>
37 @Html.DisplayNameFor(model => model.NEXT_ROUTER_HOP)
38 </th>
39 <th>
40 @Html.DisplayNameFor(model => model.PACKET_COUNT)
41 </th>
42 <th>
43 @Html.DisplayNameFor(model => model.BYTE_COUNT)
44 </th>
45 <th>
46 @Html.DisplayNameFor(model => model.PACKET_TIME)
47 </th>
48 <th>
49 @Html.DisplayNameFor(model => model.TCP_FLAG)
50 </th-->
51 <th>
52 @Html.DisplayNameFor(model => model.PROTOCOL)

```

Fig30: Table Created to Display IOC data from SOS Threat Analytics database

The output for this page has been demonstrated in Fig41 within Section11: GUI and Usability. All other pages with the SOS Threat Analytics application access and display their data in the same manner i.e. a model is built based on a specific database table, the controller and view are subsequently created and the view is designed and developed using a combination of Bootstrap and CSS for visualisation. The below sections will show some code extracts for some of the key features/functionality of the application.

### 9.3.1. Twitter & RSS Feeds

```

<!--Embed SOS Threat Analytics Twitter Feed-->
<div class="col-md-6 text-center">
  <a class="twitter-timeline" data-width="400" data-height="400" href="https://twitter.com/SOSAnalytics">Tweets by SOSAnalytics</a>
  <script async src="//platform.twitter.com/widgets.js" charset="utf-8"></script>
</div>

```

Fig31: Code Extract to embed the SOS Threat Analytics Twitter Feed

```

<!--Embed Krebs on Security RSS Feed-->
<div class="col-md-3">
  <h4 class="text-center">Krebs on Security</h4>
  <div id="widgetmain" style="text-align:left;overflow-y:auto;overflow-x:hidden;width:300px;background-color:transparent; border:1px solid #858585;">
  <div id="rsswidget" style="height:500px;"><iframe id="rssOutput"
  src="http://us1.rssfeedwidget.com/getrss.php?time=1493667470038&px=https%3A%2F%2Fkrebsonsecurity.com%2Ffeed%2F&w=300&am;
  h=500&bc=858585&bw=1&bgc=transparent&im=20&it=false&lc=FF4350&ls=14&lb=false&id=true&dc=333333&ds=14&id=
  Security&it=true&tc=333333&ts=20&tb=transparent&il=true&ic=FF4350&ls=14&lb=false&id=true&dc=333333&ds=14&id=
  idt=true&dtc=284F2D&dts=12" border="0" marginwidth="0" marginheight="0" vspace="0" hspace="0" style="border:0; padding:0; margin:0;
  width:300px; height:500px;">Reading RSS Feed ...</iframe></div><div id="widgetbottom" style="text-align:right;margin-bottom:0;border-top:1px
  solid #858585;"><span style="font-size:70%"><a href="http://www.rssfeedwidget.com">rss feed widget</a>&nbsp;</span><br></div></div>
</div>

```

Fig32: Code Extract to embed an RSS Feed from Krebs on Security

### 9.3.2. Display Threat Details

A key selling point of the application is its ability to allow a user to view information about a potential threat and to identify the root cause, potential resolution, and pro-active measures to prevent a reoccurrence. On drilling down into the detail of a particular threat, a user will be presented with a document or web page embedded within an iFrame. The code to embed this information is shown below:

```
<div class="col-md-offset-7">
</div>
<iframe width="600" height="600" src="@Html.DisplayFor(model => model.IOC_URL)"></iframe>
</div>
```

Fig33: Code Extract to create an iFrame with embedded document/webpage

As this page has been modelled on a 'view' from the SOS Threat Analytics database, the above code extracts a URL for the specific threat and embeds it within an iFrame on the page. This visual representation is a core piece of the user experience in terms of application functionality and acts as the applications unique selling point (USP).

### 9.3.3. Print Details

Users are given the option to print details from certain pages e.g. Indicators of Compromise, Device Details & Contacts. This offers another element to the overall user experience, allowing for the quick print and retrieval of reports for senior management or other interested stakeholders. Below is a code extract displaying a small bit of JavaScript that is responsible for the printing function:

```
<script>
function PrintFunction()
{
    window.print();
}
</script>
```

Fig34: JavaScript function for Enabling Printing

This print function is called when a user clicks the print button show below:



Fig35: Print Button

This button was coded to call the print function as follows:

```
<button onclick="PrintFunction()"
class="text-left btn btn-primary">&nbsp;&nbsp; Print Details&nbsp;&nbsp; </button>
```

Fig36: Create 'Print Details' button and call 'Print' function

## 9.4 SMS Alerting

If an IOC is identified on the network, it is imperative that the user is alerted so that they can actively investigate and remediate the issue in the quickest possible timeframe. SOS Threat Analytics is configured to send SMS alerts to the required party in the event of such an occurrence. The SMS alerting facility has been developed using a 'SQL CLR Service' within Visual Studio. The development of this service required substantial effort and makes use of the 'Vodafone Bulk Texting' facility to which the Project Technical Lead could gain access (through his current employer!!). The bulk texting URL is accessed via a web service call within the CLR code which can be seen in Fig37 below:

```
[Microsoft.SqlServer.Server.SqlProcedure]
public static void SqlStoredProcedure1 (String number,String text)
{
    SqlPipe sp;
    sp = SqlContext.Pipe;
    String url;
    HttpWebResponse response;
    try
    {
        // Build the URL up
        url = "https://bulktext.vodafone.ie/HTTP_API/V1/sendmessage.aspx?user=goodbody&password=Goodbody123!&api_id=12858&to="
            + number + "&text=" + text + "&from=AlertMon";

        // Create the request with the URL
        WebRequest webRequest = WebRequest.Create(url);
        webRequest.Method = "GET";
        webRequest.Timeout = 10000;
        sp.Send("Starting to call the url " + url);

        response = (HttpWebResponse)webRequest.GetResponse();
    }
}
```

**Fig37: SqlStoredProcedure1.cs – uses a Web Request & creates an equivalent stored procedure in SQL Server**

The SQL CLR Service class is called 'SqlStoredProcedure1' and was developed by creating a direct reference to the SOSThreatAnalytics database. The code in Fig37 was then built and published, generated this stored procedure within SQL server (this is critical in terms of calling the service later). The code extract effectively makes a request to the Vodafone Bulk Text SMS gateway, passing in a number and a text message. By accessing the URL via this Web Request (and passing in the text and number), the SMS gateway receives information and forwards it to the designated mobile number.

Whilst this worked as a proof of concept, the next step was to develop a way of generating these SMS's based on database behavior i.e. the identification of new IOC's. SMS message should only be received when there is an issue, so we were required to create another stored procedure within SQL server. This stored procedure (Alert\_Poll) developed a level of intelligence around the prompting and timing of SMS alerting. Section 9.2.1 outlined the requirement for the database table SMS\_TRACKER. The Alert\_Poll stored procedure makes use of this table to decide when an SMS should be sent. The crux of this intelligence is to first populate the SMS\_TRACKER with the current number of 'hits' for each IOC listing. This is demonstrated below in Fig38.

```

--Populate the SMS Tracker table with current IOC hit rates snapshot @ this moment in time
INSERT INTO SMS_TRACKER(HITS,IOC_LISTING_ID)
SELECT HITS,IOC_LISTING_ID FROM VW_NETFLOW_IOC;

--DECLARE THE VARIABLES FOR HOLDING DATA.
DECLARE @DIFF INT
        ,@IOC_ID INT
        ,@TEL_NUMBER VARCHAR(100)

```

Fig38: Read in the current hits and declare our variables

Once this is done we have our current baseline. A SQL job called 'ThreatAlerts' was developed to continually run this exercise after a pre-defined time-period e.g. 10 secs. Each time it runs, a new record and associated 'hit count' is created. The idea was then to subtract the penultimate reading from the most recent and if it has grown by more than 5 'Hits' i.e. there were 5 new occurrences of the IOC – an SMS would be sent. To do this the data in the table needed to be sorted by rank per IOC. Furthermore, as we are only interested in the most recent two readings, the rest could be discarded – this was all handled by the code extract in Fig39:

```

-- Initial Ranking
-- Puts a ranking order (1,2.. etc) for each individual grouping of IOC_LISTING_ID by ID DESC
-- and only returns the top 2 for each grouping
SELECT *
FROM (
select id , hits,IOC_LISTING_ID,
RANK() OVER (PARTITION BY IOC_LISTING_ID ORDER BY id desc) RANK_ORDER
from SMS_TRACKER )
X
where x.RANK_ORDER < 3
) Y
) Z
where
z.RANK_ORDER =1 ;

```

Fig39: Ranking of IOC Hit Counts and Returning only the two most recent records

The following code extract utilizes the results of Fig39 to compute the difference in those two most recent records:

```

-- Gets the difference between ioc listings and returns the difference
DECLARE ALERTS CURSOR READ_ONLY
FOR
-- Filters the ranking of the below SQL to return the top ranking record for each distinct
-- grouping IOC_LISTING
SELECT IOC_LISTING_ID,DIFF FROM
(
-- After the Initial Ranking (see the below ), we now need to take the difference
-- between each IOC_LISTING group i.e. take hits count from record 1 minus the hit count of record
SELECT Y.ID,Y.HITS,Y.IOC_LISTING_ID,
Y.HITS - LAG(Y.HITS, 1, 0) OVER (PARTITION BY Y.IOC_LISTING_ID ORDER BY Y.ID asc) AS DIFF,
RANK() OVER (PARTITION BY IOC_LISTING_ID ORDER BY id desc) RANK_ORDER
FROM
(
-- Initial Ranking

```

Fig40: Takes the Ranking results and generates the calculated difference between the two

Now we must decide what action to take if the difference is either greater than or less than 5 (IOC Events). Ultimately, if it is greater than 5 we need to call our previously developed Stored Procedure (containing web service to Vodafone Bulk Text) to send the required SMS. Before we can do that though we need to enter the number to be contacted. SMS Threat Analytics reads the contact numbers from the Contacts database table. An example of this is shown below:

```
-- Returns the Number associated to Contact id 2
| SELECT
    @TEL_NUMBER =Telephone
FROM [SOSThreatAnalytics].[dbo].[Contacts]
where id=2;

PRINT 'NUMBER is ' +@TEL_NUMBER
```

**Fig40: Reading in the SMS number**

This code reads the phone number of the person listed as second in the contacts table (Stuart O'Shaughnessy) and logs that number as the SMS contact. The following code (Fig41) is the decision maker for the sending/non-sending of the message:

```
--OPEN CURSOR.
OPEN ALERTS

--FETCH THE RECORD INTO THE VARIABLES.
FETCH NEXT FROM ALERTS INTO
@IOC_ID, @DIFF

-- Now we loop through each of records and check if the diff is greater than 5
-- if so we need to send an sms alert
WHILE @@FETCH_STATUS = 0
BEGIN

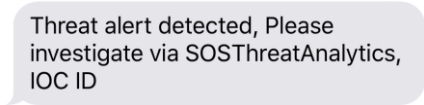
    IF @DIFF > 5
    BEGIN
        PRINT 'Send alert to IOC listing ' +CAST(@IOC_ID AS VARCHAR(100)) + ', the diff count is '
        +CAST(@DIFF AS VARCHAR(100))
        -- Now send SMS alerts to admin
        EXEC [dbo].[SqlStoredProcedure1]
            @TEL_NUMBER,
            @text = N'Threat alert detected, Please investigate via SOSThreatAnalytics, IOC ID is '
    END

    IF @DIFF < 5
    BEGIN
        PRINT 'Not alert required for IOC listing ' +CAST(@IOC_ID AS VARCHAR(100)) + ', the diff count is '
        +CAST(@DIFF AS VARCHAR(100))
    END

    -- Proceed onto the next record in the results
    FETCH NEXT FROM ALERTS INTO
    @IOC_ID, @DIFF
END
```

**Fig41: To Send or Not Send an SMS**

This simple while loop checks if the IOC count is above 5 hits. If so it sends an SMS to the number read in above (Fig40) and if not, no action will be taken. Below is a screenshot of an SMS from the service:



Threat alert detected, Please investigate via SOSThreatAnalytics, IOC ID

**Fig42: Sample SOS Threat Analytics SMS Alert**

## 10. Testing & Evaluation

With the implementation completed the next key step in finalising the system was the performance of extensive testing. This section will detail the test plan utilised, the over testing approach as well a detailed table of test results and functionality sign-off.

### 10.1 Testing Objectives

The objectives of this test plan are to analyse all areas of the SOS Threat Analytics project/platform and to verify that the end-product is fit for its designed purpose and suitable to be passed into production. The application must be verified in terms of performance, response, resilience and execution of its core functionalities and user requirements.

### 10.2 Testing Scope

The scope of the plan is based firmly around the requirements set out in Section7 of this document. These requirements include Application, System, Data and the Online Marketing Website. This test plan will further include information on the testing of the non-functional requirements also set out in Section7. Ultimately, all components that are required for the optimum level of operation of the SOS Threat Analytics platform must be tested and fully signed off.

### 10.3 Testing Strategy

The approach to testing here was two-fold. Due to time constraints and general availability, the majority of the system testing was performed by the Technical Project Lead – Stuart O’Shaughnessy. The local environment i.e. local installation of the required tools (IDE, programming environment, database engine etc.) was utilised for all aspects of the system testing. The configuration of the SOS Threat Analytics product is all based on a local environment so this represents full ‘production’ environment testing.

In terms of User Acceptance Testing, the items that were readily available for remote testing were completed first i.e. the Marketing Website from its publicly available web address. A demonstration of the back-end functionality, data flow and database storage was given to Project Supervisor Manuel Tova-Izquierdo in late March to verify that the project concept, complexity, functionality and end goals were all on track. As a final piece of engagement, the individuals who engaged with the ‘Requirements Elicitation’ process were given a combination of face to face and Skype demos of the functionality and its alignments to the agreed set of requirements.

### 10.4 Testing & Evaluation Results Tables

The following set of tables detail the full list of tests that were completed. This includes confirmation of their sign-off as well as some detailed comments and constructive feedback received from test users. The set of tests are based entirely on the ‘Requirements Specification’ that were set out in detail within Section7 above.

SOS Threat Analytics – Testing & Evaluation Sheet – Functional Requirements				
#	Requirement	Tested	Passed	Comments
<b>Application Testing</b>				
1	Application Access	Yes	Yes	Application is fully accessible in the locally installed environment - which is the actual production environment at present. Remote testers were required to test the application and all functionality on the Technical Project Lead's local computing environment. All connectivity tests were successful.
2	User Registration	Yes	Yes	User registration was constructed within the ASP.net environment using the local User Database. All registration testing was successful and multiple accounts/details have been configured on the environment
3	User Sign-In	Yes	Yes	All registered accounts were successfully tested for Sign-In operations. There were no issues with accessing the service once the users had successfully registered a valid user account
4	View User Profile	Yes	Yes	Each user validated that they could access their own individual user profile once logged into the system - all tests were passed successfully.
5	Update User Profile	Yes	Yes	Each user validated that they could amend their own individual user profile once logged into the system - all tests were passed successfully.
6	Application Navigation	Yes	Yes	Application Navigation is quite a wide ranging one but effectively the feedback here came down the consistency. The app was designed so that all key areas were readily available from the Nav-Bar and the layout was as user friendly as possible. General feedback was that the app was easily navigable, all information was presented in the right manner and there were no issues in terms of not finding particular pages or information. ASP.net web applications tend to lend themselves to good navigation and the feedback was universally positive.



SOS Threat Analytics – Testing & Evaluation Sheet – Functional Requirements				
#	Requirement	Tested	Passed	Comments
<b>Application Testing</b>				
7	Access Dashboard	Yes	Yes	All users confirmed that they could access the Dashboard page without issue and the page rendered as expected
8	View IOC Graphic	Yes	Yes	All users confirmed that the IOC Graphic appeared as expected on Dashboard page load
9	View Twitter Feed	Yes	Yes	All users confirmed that the SOS Threat Analytics Twitter Feed appeared as expected on Dashboard page load
10	View RSS Feeds	Yes	Yes	All users confirmed that the set of RSS Feeds appeared as expected on Dashboard page load
11	Access the Indicators of Compromise Page	Yes	Yes	All users confirmed that they could access the 'Indicators of Compromise Page' without issues and the page rendered as expected
12	Access Detailed IOC Information	Yes	Yes	All users confirmed that by clicking the 'Detailed IOC Information' they navigated to an in-depth breakdown of the particular threat as well as an Advisory note around same. Feedback on this feature was also universally positive
13	Access List of Monitored Devices	Yes	Yes	All users confirmed that they could access the 'Device Listing' page without issue and the page rendered as expected
14	Access Device Details	Yes	Yes	All users confirmed that by clicking the 'Details' link for an associated device, they automatically navigated to a 'Device Details' page that gave an overview of that particular device

SOS Threat Analytics – Testing & Evaluation Sheet – Functional Requirements				
#	Requirement	Tested	Passed	Comments
<b>Application Testing</b>				
15	Access Vendor & Client Contacts	Yes	Yes	All users confirmed that they could access the 'Contacts' page without issue and the page rendered as expected
16	Adding a Contact	Yes	Yes	All users successfully added a contact and were impressed that the contact would be rejected if the fields were not correctly entered. This ensures that all created contacts are consistent in terms of details
17	Amending a Contact	Yes	Yes	All users confirmed the successful amendment of a Contact detail e.g. phone number email address etc.
18	Deleting a Contact	Yes	Yes	All users successfully deleted a contact. Feedback was good around the confirmation message prior to deletion as it was felt that this feature would avoid mistakes
19	Log Support Request	Yes	Yes	This item passed but the feedback was that it was quite basic i.e. it was effectively just a 'mailto:' option. Users would like to see this evolved to a fully-fledged support system
20	Print Pages	Yes	Yes	Users confirmed that where the 'Print' button was available it worked perfectly. All reports are printed in .pdf which users were positive about. Furthermore, all users felt that they locations of the print buttons were user friendly and the process was very simple.
21	Alerting & Notifications	Yes	Yes	Tested successfully via SMS on all user's phones. Whilst the testers did not get access to the final 'automated' test a demo was constructed and the application tested each user individually. By far this was the strongest feedback received and most users outlined this feature as the most important one to them.

SOS Threat Analytics – Testing & Evaluation Sheet – Functional Requirements				
#	Requirement	Tested	Passed	Comments
<b>System Testing</b>				
1	NetFlow Generation	Yes	Yes	The Paessler NetFlow Generator testing revolve around the verification that the ingested NetFlow packet could be broken out and validated against the expected structure (this is detailed in section 4 of this document). Further testing focused on the running of multiple instances at once - this was also successful for all tests.
2	Apache Spark Application	Yes	Yes	Testing of the Apache Spark application was specific to debugging the code within the Eclipse IDE and verifying that the database integration was functioning as expected. From both a system admin and user perspective, all elements of the application were verified as functioning at an optimum level and fir for purpose for SOS Threat Analytics requirements.
3	Front-End Application (ASP.net MVC)	Yes	Yes	The functionality of the application was tested extensively as outlined above in 'Application Testing'. The system test element focused on the reliability of the application, its likelihood to fail etc. To this point there has been no action taken on the app that has resulted in failure. The application launches as expected and responds in a fast an efficient manner. All functionality is embedded and manifests itself on application start-up.
<b>Data Testing</b>				
1	SQL Server Relational Database	Yes	Yes	The SQL Server Database for the application is called simply (SOS Threat Analytics). The basic functions of the database were tested extensively i.e. creating/dropping tables, adding values, creating views etc. From a system perspective SQL servers' integration to the back-end (Apache Spark) and front-end (ASP.net) were tested extensively to ensure that all required data elements could be ingested and presented to the end user. All tests were completed successfully

SOS Threat Analytics – Testing & Evaluation Sheet – Functional Requirements				
#	Requirement	Tested	Passed	Comments
2	Access to Required Data	Yes	Yes	The ability to access data from a user perspective is a fundamental aspect of the front-end SOS Threat Analytics application. Verification that data was present on all required pages i.e. 'Indicators of Compromise', 'Device Listing' Contacts etc.' was the core focus of data testing. All users verified the presence of data in the required application areas and this test was fully signed off.
3	Update / Amend/ Delete	Yes	Yes	Users tested the manipulation of data within the front-end application and all tests were successful. Similarly, the system admin verified that NetFlow records were actively updated via the back-end integration to Apache Spark. The ability to update/amend/delete database records was tested extensively and was fully signed off.
<b>Marketing Website Testing</b>				
1	Site Hosting	Yes	Yes	The marketing website has been fully hosted on Amazon S3 with a provisional web address of (s3-eu-west-1.amazonaws.com). All testing was completed on this address but there is a plan to move the address to sosthreatanalytics.com - time willing this will be completed. However, the website and all functionality has been fully hosted and has been tested from desktop and mobile devices. This is fully completed and signed off.
2	Site Connectivity & Accessibility	Yes	Yes	Connectivity has been tested over direct Ethernet connection, Wi-Fi, 3G and 4G. All connectivity tested was completed successfully
3	Site Functionality	Yes	Yes	Functionality on the site is limited to JavaScript items for quick scrolling and navigability options. There are also some accessible links (Social Media, Email etc.) and an embedded video. All are working as expected from the live website.

SOS Threat Analytics – Testing & Evaluation Sheet – Non-Functional Requirements				
#	Requirement	Tested	Passed	Comments
1	Performance & Response Time	Yes	Yes	The application continuously starts-up as expected and there is no delay or poor performance against what would be expected. Response times in terms of navigation are excellent and all pages' render quickly
2	Application Availability	Yes	Yes	The application is running on a local computing environment at present so availability is limited to people with access to same. There has however been no issue with application availability from this location.
3	Accessibility	Yes	Yes	The site was created with usability in mind and the ASP.net front end delivers a seamless and consistent look and feel whilst enabling simple navigation throughout. All testing completed successfully and user feedback was exceptionally positive in this regard.
4	Application & User Security	Yes	Yes	Application functionality has been restricted and alternative levels of access exist for System Admin and Standard users. Access levels have been tested and have been fully signed off. In terms of the system itself, the choices of tool set such as SQL Server, ASP.net and Amazon S3 ensure that a strong level of security is implemented across the full platform by default
5	Maintainability	Yes	Yes	The local environment means that the application is currently very maintainable - everything sits in a singular location and access levels are highly restricted. This would change in the longer term and require deeper consideration. However, for this test the current levels are more than sufficient
6	Portability	Yes	Yes	The application is not currently built to be portable - it was not designed for this capability. However, Section 11 outlines the plan for future portability/mobility through a cloud based implementation.

SOS Threat Analytics – Testing & Evaluation Sheet – Non-Functional Requirements				
#	Requirement	Tested	Passed	Comments
7	Reusability	Yes	Yes	The application is built to deal with live/real-time data therefore it must be inherently re-usable. The choice of toolsets, navigability, look & feel have all been implemented in a way to enhance user experience and encourage re-use.
8	Error Checking	Yes	Yes	Error checking has been in-built for the user access sections (all fields must be completed; correct information must be supplied etc.) this has been tested extensively. In terms of wider error checking, if there is a serious error the application will fail in its ultimate deliverable - to provide data to the end user. Extensive debugging of the code has been implemented to limit this exposure as much as possible
9	Concurrency	Yes	Yes	The application is not currently built for concurrency - it was not designed for this capability. However, Section 11 outlines the plan for future concurrency through a cloud based implementation.
<b>SMS Alerting Testing</b>				
1	Receipt of SMS Alerts	Yes	Yes	Extensive testing of the SMS Alerting facility was completed and fully signed off. The final functionality was completed quite late in the day so no user testing was performed. System admin testing was fully completed and a demonstration will take place as part of the presentation.

### 10.5 Testing Results and Conclusion

All testing was based on the initial engagement with users as part of the 'Requirements Elicitation' process as well as the additional requirements that were built out as the system was being developed. This provided a clearly defined scope and allowed the testing process to be very specific. This structure afforded a focussed and very specific process, playing a large part in its overall success. The general feedback from the user testing was very positive. The chosen users all had IT backgrounds and – crucially – were very interested in the topic. The project concept and the implementation was something they were interested in and agreed was something of value to the current market. Coming out of the testing the main finding (from all users) was that if the application was to be developed further it should be fully migrated to a Cloud based infrastructure, delivering the requisite levels of portability, concurrency and overall accessibility. Outside of these points there were no major findings (fails) that came out of the system or user testing processes.

## 11. Graphical User Interface & Usability

This section will illuminate the majority of what has covered within this document to this point. A focal point of this project was to deliver something to the end user that was intuitive, easily navigable and had a simple look and feel. Applications live and dies by their ease of use and regardless of the value of the underlying data, if the application does not provide the user with something quick and easy to use, they will look for an alternative. SOS Threat Analytics treats the end user experience with the same dedication and scrutiny as the back-end architecture. This section will demonstrate some of the main pages that are accessible to users within the application as well as the requirement they were designed to fulfil.

### 11.1 Splash/Home Screen

Below is the first screen the user sees once the application has been successfully launched.



Fig43: Application Splash/Home Screen

This view was designed to be very simplistic presenting only the 'Register' & 'Sign-In' options.

### 11.2 Navigation Bar - Logged in User

You will notice that when a user initially hits the application, the navigation bar is empty. This is because the applications' functionality is only available to a valid and logged in user. When a user has successfully signed into the application they will notice several options have appeared within the Navigation bar:

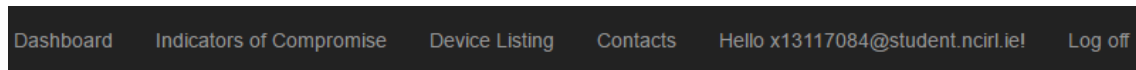


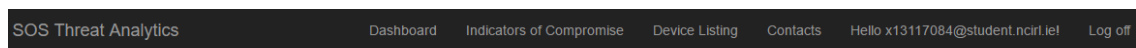
Fig44: Navigation bar as viewed by a validated user

This presents the user with the option to navigate around the application using the various different pages presented above. Each of these pages will be demonstrated individually within this section.

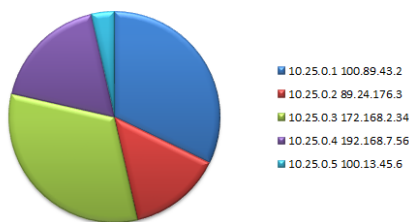
### 11.3 Dashboard

The first option on the menu is the 'Dashboard'. The page has two key features: 1) To graphically summarise the number of potential network threats, 2) Deliver live Information Security News Feeds to the end user. The Dashboard is comprised of a JavaScript Graphic that displays the number of current active threats or 'Indicators of Compromise'. Furthermore, it has an embedded Twitter feed from the official 'SOS Threat Analytics' account as well as 3 real time embedded RSS feeds from the following sources:

- Krebs on Security
- Threat Post
- PacketStorm Security



#### IOC Overview



#### Tweets by @SOSAnalytics



Fig45: IOC Graphic and embedded SOS Threat Analytics Twitter feed

The IOC Overview (graphic) can be clicked to open the 'Indicators of Compromise' page that will be detailed in Section 11.4. Below is a visual of the full dashboard page including the above, the RSS feeds and some embedded links to SOS Threat Analytics social media accounts:



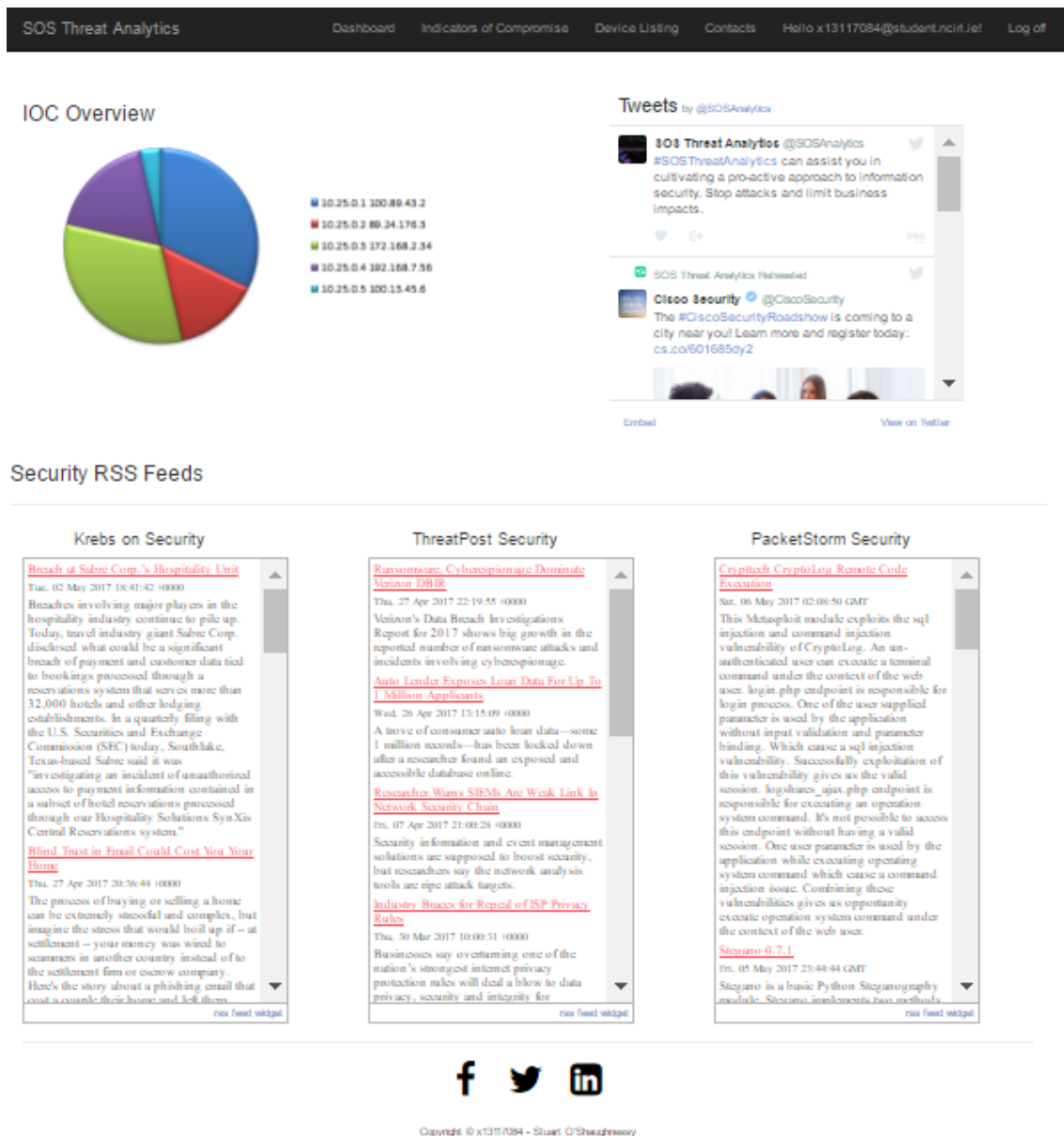


Fig46: The full SOS Threat Analytics Dashboard View

Each of the RSS feeds shown above are fully interactive and the user can click any of the links provided to gain access to the webpage that contains the full article. SOS Threat Analytics sees the Dashboard view as a hive of activity from which the user can glean important information and keep up to date on the latest security trends.

### 11.4 Indicators of Compromise

The main point of interest for the use of SOS Threat Analytics is actually getting access to the potential threats in order to understand the risk to your business. The page that delivers this level of insight is 'Indicators of Compromise' (IOC). Accessed either via the navigation bar or via clicking the 'IOC Overview' graphic, this page presents a high-level summary of the threats as shown in Fig41:

THREAT	SRCADDR	DSTADDR	DSTPORT	PROTOCOL	HITS	
RansomWare: Locky	10.0.0.1	95.181.171.58	80	TCP	196	<a href="#">Detailed IOC Information</a>
SMTP Data Extraction	10.0.0.1	10.0.0.2	110	TCP	392	<a href="#">Detailed IOC Information</a>
SMTP Data Extraction	10.0.0.1	10.0.0.2	143	UDP	63	<a href="#">Detailed IOC Information</a>
Bot: PinksipBot	10.0.0.1	10.0.0.2	31666	TCP	10	<a href="#">Detailed IOC Information</a>
Bot: PinksipBot	10.0.0.1	10.0.0.2	16666	TCP	8	<a href="#">Detailed IOC Information</a>
Trojan: PoweLike	10.0.0.1	31.184.192.80	80	TCP	10	<a href="#">Detailed IOC Information</a>
Trojan: PoweLike	10.0.0.1	195.2.241.84	143	UDP	10	<a href="#">Detailed IOC Information</a>

[Print Details](#)

**Fig47: Indicators of Compromise Overview Table**

This table and its 'Model & View' within ASP.net are based off the Database View detailed in Section 9.2. The table reads and displays all relevant data from this view including (The Threat Name, Source IP, Destination IP, and Port as well as the protocol and number of Hits. This overview table delivers information at a high-level directly to the end user. More importantly however, is the link that appears next to each individual threat – 'Detailed IOC Information'. By clicking this link, the user can drill down into the threat and receive information on what the threat is, its potential impact and all available method of remediation. This is delivered through a threat advisory note or website that has been embedded within an 'iFrame' on the page. An example of this is show below:

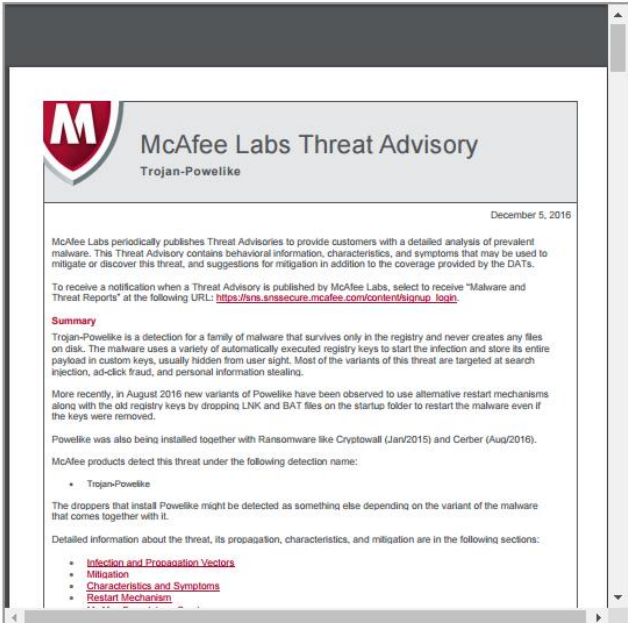
SOS Threat Analytics

[Dashboard](#)
[Indicators of Compromise](#)
[Device Listing](#)
[Contacts](#)
[Register](#)

## Threat Details: Trojan: PoweLike

THREAT Trojan: PoweLike  
 SRCADDR 10.0.0.1  
 DSTADDR 31.184.192.80  
 DSTPORT 80  
 PROTOCOL TCP  
 HITS 10

[Back to List](#)



The image shows a screenshot of a McAfee Labs Threat Advisory page. The header includes the McAfee logo and the title 'McAfee Labs Threat Advisory Trojan-PoweLike' with a date of December 5, 2016. The main content area contains a summary of the threat, stating that it is a detection for a family of malware that survives only in the registry and never creates any files on disk. It also mentions that new variants were observed in August 2016 and that the malware uses alternative restart mechanisms. A list of detection names is provided, including 'Trojan-PoweLike'. At the bottom, there are links to sections for 'Infection and Propagation Vectors', 'Mitigation', 'Characteristics and Symptoms', and 'Restart Mechanism'.


**Fig48: Detailed Information on the Threat including an Advisory notice for Potential Remediation**

Fig42 shows the screen a user would receive on clicking for more details on a potential threat identified over known malicious IP – 31.184.192.80 - this IP relates to the 'PoweLike' Trojan. As this is a known threat within our database, the requisite security advisory URL is retrieved via an 'iFrame' and is displayed to the user. The key benefit here is that the user can gain access to informed material around the threat and how best to deal with it. There is often detailed information within these advisories around preventing a reoccurrence. It should be noted that the document within the iFrame can also be printed directly from the page if required. This can be invaluable to the end-user and is a valuable aspect and selling point of the SOS Threat Analytics platform. This level of information is delivered to the end user for all known threats within the SOS Threat Analytics database.

### 11.5 Device Listing

Clicking on this option from the navigation bar does exactly what it says on the tin. A full list of the devices in scope for monitoring on the client network are displayed. This page offers a good snapshot of each individual device and is essentially the book of record for the monitored infrastructure. This page connects directly back to the SOS Threat Analytics database where the actual information is stored and the page itself has been created using a model of the Device details table (dbo.Device\_Details). As per Fig43 there is also an option to print the configuration if the user desires.

SOS Threat Analytics			
Dashboard		Indicators of Compromise	Device Listing
Contacts		Hello x13117084@student.nciri.ie!	
Device_Name	Device_Model	Device_IP	
HQ_SW_01	Cisco 4500 LAN Switch	10.25.0.1	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
HQ_SW_02	Cisco 4500 LAN Switch	10.25.0.2	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
HQ_SW_03	Cisco 4500 LAN Switch	10.25.0.3	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
HQ_SW_04	Cisco 4500 LAN Switch	10.25.0.4	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
HQ_ISR_01	Cisco Integrated Services Router	10.25.100.1	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
HQ_ISR_02	Cisco Integrated Services Router	10.25.100.2	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Regional_SQ_01	Cisco 3800 Router	10.16.25.1	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Regional_SQ_02	Cisco 3800 Router	10.16.25.2	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Regional_SQ_03	Cisco 3800 Router	10.16.232.1	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Regional_SQ_04	Cisco 3800 Router	10.16.232.2	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
DC_ASA_WEBRTR_01	Cisco ASA 3310 Firewall	10.150.25.1	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
DC_ASA_WEBRTR_02	Cisco ASA 3310 Firewall	10.150.25.2	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
DC_SW_01	Cisco CSS 11500 Content Switch	10.150.25.100	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
DC_SW_02	Cisco CSS 11500 Content Switch	10.150.25.200	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>



[Print Details](#)


Fig49: List of Actively Monitored Devices

### 11.6 Vendor & Client Contacts

As with any application provided by a 3<sup>rd</sup> party the support structure is key. The 'Contacts' page within SOS Threat Analytics provides a list of key individuals responsible for the application from both a vendor and client perspective. This ensures that all contact information i.e. email addresses and phone numbers are readily available – removing the need to

constantly refer to a paper based SLA or support agreement. Fig44 demonstrates a sample 'Contact' page for the National College of Ireland. Contacts from both SOS Threat Analytics as well as those from the 4<sup>th</sup> Year Project module have been included.

SOS Threat Analytics      Dashboard   Indicators of Compromise   Device Listing   Contacts   Hello x13117084@student.ncirl.ie   Log off



Key Account Contacts

Company	Name	Position	Email	Telephone	
SOS Threat Analytics	SOS Service Support	Application Support	<a href="mailto:support@sosthreatanalytics.com">support@sosthreatanalytics.com</a>	+353 1 6419437	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
SOS Threat Analytics	Stuart O'Shaughnessy	Technical Project Lead	<a href="mailto:stuartos@sosthreatanalytics.com">stuartos@sosthreatanalytics.com</a>	+353 868077524	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
National College of Ireland	Manuel Tova-Izquierdo	Project Supervisor	<a href="mailto:manuel.tova.izquierdo@ncirl.ie">manuel.tova.izquierdo@ncirl.ie</a>	+353 1 4498500	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
National College of Ireland	Eugene McLaughlin	Project 2nd Marker	<a href="mailto:eugene.mclaughlin@ncirl.ie">eugene.mclaughlin@ncirl.ie</a>	+353 1 4498500	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
National College of Ireland	Eamon Nolan	Project Coordinator	<a href="mailto:eamon.nolan@ncirl.ie">eamon.nolan@ncirl.ie</a>	+353 1 4498500	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
National College of Ireland	Joe Mulumby	Project 2nd Marker	<a href="mailto:joe.mulumby@ncirl.ie">joe.mulumby@ncirl.ie</a>	+353 1 4498500	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

[Create New](#)

**Fig50: Vendor and Client Contacts table**

Contacts can be added, amended and deleted as required directly from this page. The page reads its information from the SOS Threat Analytics database and has been created using a model of the 'Contacts' table (dbo.Contacts). When adding a contact, the required fields appear as per Fig45 below:

SOS Threat Analytics      Dashboard   Indi

### Create

Contact

---

**Name**

**Position**

**Email**

**Telephone**

**Company**

[Back to List](#)

**Fig51: Page for Creating a new Contact (Updating the Database)**

## 12. System Evolution

The initial plan for this project is very much based around delivering a working version of the core concept i.e. providing an end-user with a view of the activity on their network. In a production environment, large changes would be required to handle the volume of data and to enhance resilience, connectivity, and availability. The core of the application (Apache Spark, SQL Server, ASP.net) would remain the same. However, a migration to a Cloud based solution consisting of multiple virtual machines would be a necessity. To handle the large volumes of data a clustered file structure optimized for 'Big Data' would also be required. Following detailed consideration, I believe Fig46 below, represents the required system architecture for a large-scale implementation of SOS Threat Analytics:

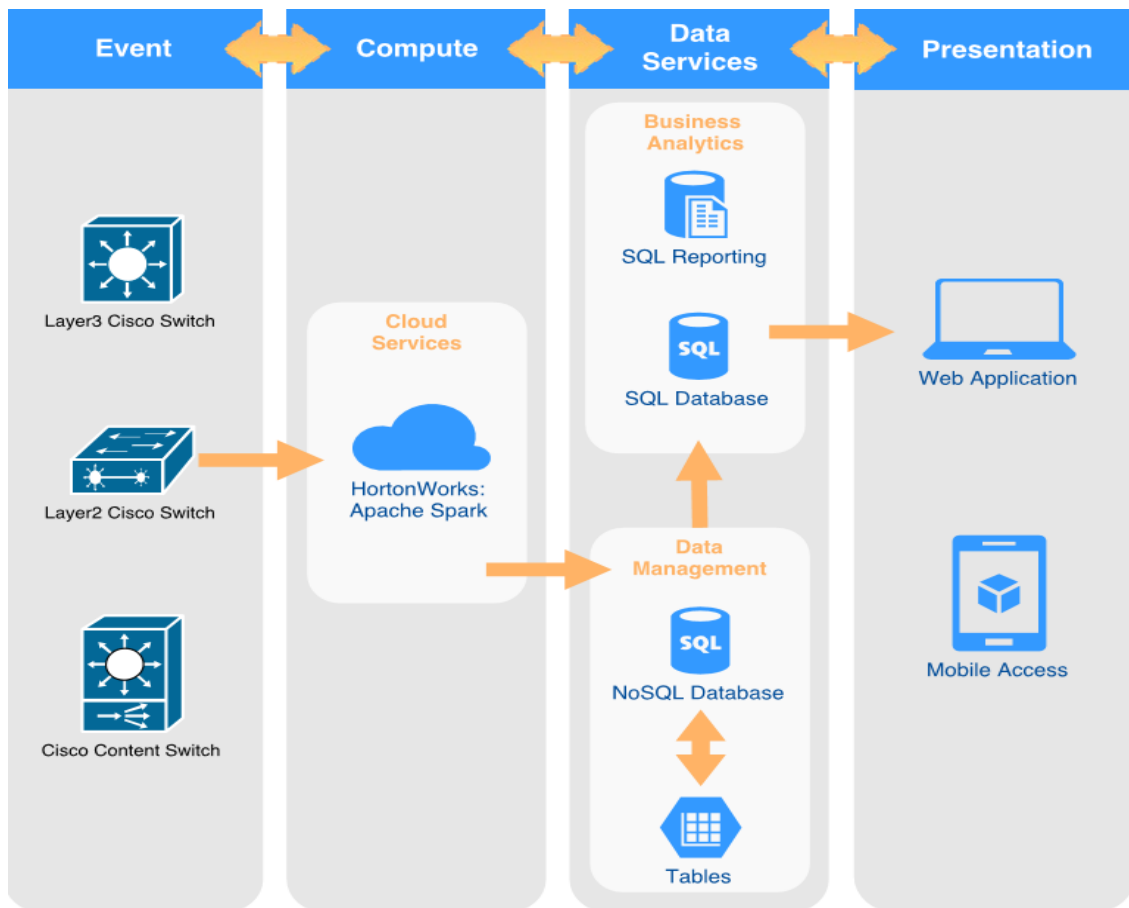


Fig52: Future Solution Architecture

The key difference here can be seen in the 'Compute' and 'Data Services' sections. For handling the large-scale data, an implementation of the Horton Works Data platform (HDP) would be utilised. "The Hortonworks Data Platform (HDP) product includes Apache Hadoop and is used for storing, processing, and analysing large volumes of data." (Wikipedia, 2017). At a high level, the device would be configured to send its NetFlow data to the IP address of the Hortonworks instance over a specific port. The Apache Spark application would connect to this system and perform the relevant operations on the data. Due to the size of the data a

NoSQL database (HBase) would be considered for the subsequent data storage i.e. the 'Spark output. A method of migrating the data from NoSQL to a SQL server instance would then also be required. It would be preferable to go straight from 'Spark to SQL server but my research says that this could be difficult due to the structure and volume of the (Big) data. SQL remains part of the infrastructure due to its simplistic integration with ASP.net – which again would be utilised for the front-end application. Another overview of the infrastructure – hosted within a Microsoft Azure cloud, would be as below:

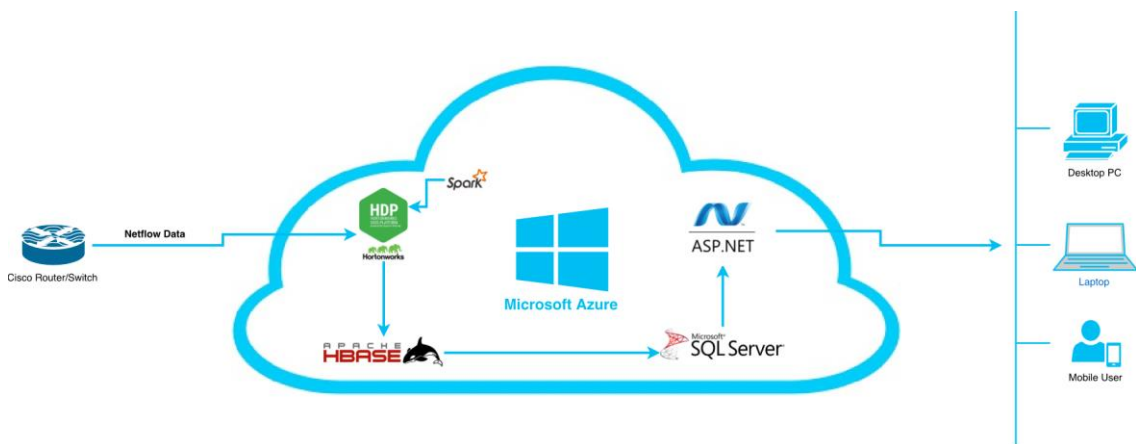


Fig53: Future Solution Topology and Technologies

If this project was to be further developed and productized, the following core requirements would need to be met:

- Purchase an appropriately spec'd cloud based infrastructure with logical segregation of virtual machines (VM's) for each individual customer
- Each individual VM would need to be spec'd in line with customer requirements
  - Number of Devices
  - Proposed volume of data
  - Etc.
- Enhanced security controls based on the principle of least accessing provisioning i.e. users should only have access to the data/screens/streams to which they are authorized
- Web Application Development
  - Front-End Development
  - Mobile Capabilities
  - Additional Functionality (JavaScript/JQuery, Ajax etc.)
  - Enhanced Database Design & Security
- Online Marketing
  - Continued Website Development
  - Online Customer Service and Consumer Interaction
  - Increased Social Media Presence

In terms of the end-user one of the key system developments would be the enhancement of user roles and creation of multiple user access levels e.g. administrator, standard user etc. Ceding control of user creation and management would empower the customer and allow them to provision access in accordance with their local information security and access control policies etc. It would also allow them to cater for staff churn and organizational changes without requiring constant interaction with the vendor.

In terms of product development and marketability, the core concepts of this project could be enhanced and evolved to an enterprise tool. Given the right level of time and resources to grow the product, significant inroads could be made into the market. Information & Cyber Security are at the fore-front of enterprise risk for organizations across the world. Many of these organizations are looking to expand their portfolio of management tools whilst simultaneously managing costs. Creating a tool that focusses on the niche market of network activity and delivering this at an acceptable cost would be an attractive prospect for many within the industry.

### 13. Target Market

This product is aimed at security conscious organizations who are looking to protect their network from malicious activity. Effectively it is aimed at progressive IT security professionals within an organization who are looking to really understand what is occurring on their network on a day to day basis. This activity ranges from security threats from known IP ranges or ports, to (with future development) a volumetric trend analysis of what is occurring. The targeted organizations can range from a variety of different industries i.e. commercial, financial, government etc. The intended audience is any organization that is actively running a Cisco based network infrastructure, regardless of their front office functions or objectives.

#### 13.1 Market Rivals

There are without doubt several rival products on the market that will do similar operations e.g. products from Solar Winds, Paessler and others such as MachineEngine. SOS Threat Analytics looks to create an open-source product that will encourage better monitoring of network activity and empower security professionals. Regardless of rival products or vendors in this space, I believe that this is a challenging initiative and it is also something that I am (and have been!!) passionate about developing.

## 14. References

Techopedia.com. 2017. What is NetFlow? - Definition from Techopedia. [ONLINE] Available at: <https://www.techopedia.com/definition/28315/NetFlow>. [Accessed 06 May 2017].

Simulates NetFlow data streams for testing purposes. 2017. Simulates NetFlow data streams for testing purposes. [ONLINE] Available at: <https://www.paessler.com/tools/NetFlowgenerator>. [Accessed 06 May 2017].

What Is Apache Spark? Webopedia Definition. 2017. What Is Apache Spark? Webopedia Definition. [ONLINE] Available at: <http://www.webopedia.com/TERM/A/apache-spark.html>. [Accessed 06 May 2017].

SearchAWS. 2017. What is Amazon Simple Storage Service (Amazon S3)? - Definition from WhatIs.com. [ONLINE] Available at: <http://searchaws.techtarget.com/definition/Amazon-Simple-Storage-Service-Amazon-S3>. [Accessed 06 May 2017].

Wikipedia. 2017. ASP.NET MVC - Wikipedia. [ONLINE] Available at: [https://en.wikipedia.org/wiki/ASP.NET\\_MVC](https://en.wikipedia.org/wiki/ASP.NET_MVC). [Accessed 06 May 2017].

Mark Otto, Jacob Thornton, and Bootstrap contributors. 2017. Bootstrap · The world's most popular mobile-first and responsive front-end framework. 3.3.6 Documentation - BootstrapDocs. [ONLINE] Available at: <http://bootstrapdocs.com/v3.3.6/docs/>. [Accessed 2 May 2017].

SearchWinDevelopment. 2007. What is C#? - Definition from WhatIs.com. [ONLINE] Available at: <http://searchwindevelopment.techtarget.com/definition/C>. [Accessed 6 May 2017].

reqtest.com. 2012 No page title. [ONLINE] Available at: <http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>. [Accessed 4 May 2017].

GUI Definition. 2004. GUI Definition. [ONLINE] Available at: <http://www.linfo.org/gui.html>. [Accessed 10 May 2017].

Digital Guardian. 2016. What are Indicators of Compromise? | Digital Guardian. [ONLINE] Available at: <https://digitalguardian.com/blog/what-are-indicators-compromise>. [Accessed 10 May 2017].

Java Programming Language. 2014. Java Programming Language. [ONLINE] Available at: <http://docs.oracle.com/javase/7/docs/technotes/guides/language/>. [Accessed 10 May 2017].

SearchMicroservices. 2014. What is JavaScript? - Definition from WhatIs.com. [ONLINE] Available at: <http://searchmicroservices.techtarget.com/definition/JavaScript>. [Accessed 10 May 2017].



What is SEO - Search Engine Optimization? Webopedia. 2017. What is SEO - Search Engine Optimization? Webopedia. [ONLINE] Available at: <http://www.webopedia.com/TERM/S/SEO.html>. [Accessed 10 May 2017].

Essential SQL. 2017. Learn about Stored Procedures - Essential SQL. [ONLINE] Available at: <https://www.essentialsql.com/what-is-a-stored-procedure/>. [Accessed 10 May 2017].

## 15. Appendices

<b>SOS Threat Analytics Project Plan</b>	 SOS Threat Analytics - Project PI
<b>Project Proposal</b>	 x13117084 - Project Proposal - 4th Year S
<b>Solution Architecture (draw.io)</b>	 <b>Final Solution Architecture.png</b>
<b>Use Case Diagram – System (draw.io)</b>	 <b>Full System.png</b>
<b>Database ERD (draw.io)</b>	 <b>Final Project DB Structure.png</b>
<b>Project Journal Entries</b>	 x13117084 - Software Project Ma
<b>Primary Research – Completed Questionnaires</b>	 Completed Questionnaires.xlsx
<b>Project Poster</b>	 <b>SOS Threat Analytics Poster - Fir</b>
<b>Testing &amp; Evaluation Sheet</b>	 Testing & Evaluation.xlsx
<b>Supervisor Meetings</b>	 4th Year Student - Learning Agreement

Due to issues with Moodle and lecturers being unable to access some embedded appendices as part of the upload, I have included the following appendices below to ensure that they can be seen and reviewed:

- Journal Entries
- Project Plan
- Completed Questionnaires
- Testing & Evaluation Sheet (See Section 10)

### **15.1 Journal Entries**

Below are the extracts from my 'Master Project Journal'. This covers all journal entries for the duration of the 'Software Project' module.

#### **15.1.1. September**

I had great intentions of starting my final year college project almost immediately after finishing my 3<sup>rd</sup> year exams in January. There was a certain amount of relief in completing semester1 and then realizing that I had the next 7 months off due to a work place exemption. Following the exams, I decided to give myself a 2 month break just to relax and focus on work & home life – both of which had been neglected to a certain amount due to exam season.

As those 2 months passed I began to think less and less about college and any potential final year project. Coupled with the fact that the other half had now placed a ban on all studying/working at home, it had made it increasingly difficult to find time to get my head into that space. To be fair to her, college had taken a lot of my time and that pressure was only increased by the birth of our daughter in late 2014. Doing my January exams that year was quite a taxing process!!! I agreed that there would be no talk of college until September and that I would simply enjoy the time off. However, I do have an approx. 1 hour commute to work in the morning so I did start to think about what sort of project I could put together...

As I work in IT currently, my initial thoughts were around a web application for call logging & incident management. What we currently use in work is not exactly fit for purpose and I could see how a more simplified and focused design would assist my team. I started to think about the technologies I would use i.e. ASP.net, SQL server, storage requirements etc. and I felt that this could actually be a runner. I spoke to some of the more senior technical guys in work and to be honest they were a little underwhelmed. They spoke of trying to differentiate myself and to push myself a little harder. They said that the project concept was fine and I would probably get a good mark, however there was a strong likelihood that the majority of projects would be fundamentally similar in design i.e. web applications talking to databases.

I had a long think about this and decided to revisit the idea and take another look around the day to day tasks I perform in work and see if there was something else I could focus my attention on. Operating in the financial services sector, there is a strong focus on the concept

of information security. I am part of the firms' information security team so I decided to see if there was something worthwhile in that space instead.

Security is a very broad topic, but an incident we underwent in late July / early August gave me an idea around a niche area that I believe offered and opportunity for further investigation. My firm was subjected to a DDOS attack and subsequent ransom demands and this really opened our eyes in terms of vulnerabilities in our infrastructure and the need for pro-active management/alerting. Many organizations appear to tick boxes in terms of their approach to security and very rarely do they get under the hood and really take a look at what is going on within their infrastructure and across their network. Our network infrastructure is entirely Cisco based and from this came my idea...

Cisco end points produce proprietary information called 'NetFlow' that contains network activity details such as source IP address, port, destination IP, connection times and duration etc. The core concept of my idea was to assimilate this data into a database of some kind and to interrogate and interpret that data in order to perform some pro-active analysis/alerting and potentially identify networks threats and spurious activity. Ultimately this data would be displayed to end users (IT Staff member) via a web application/dashboard. I gave this idea serious thought and again ran it by some of the technical team in work and the feedback was that this seemed a much more worthwhile endeavor. I understand that what I have described in this paragraph again sounds like a web application simply talking to a database but please hear me out!!! The real challenge to this project (not that the visualization piece won't be a massive struggle anyway!!) is at the back end i.e. generating a mass amount of data and identifying the technologies that will not only store this data, but filter it down into the requisite bits I require for interrogation/visualization.

Following some initial research in this space, I have decided that I will be using the Hortonworks data platform (eventually hosted in the cloud) with Apache Spark to stream and filter the data into a NoSQL (probably HBASE) database. I have chosen NoSQL due to the volume of data and also as I believe it offers another element of learning and complexity to the project. From here I may look to import the filtered data into a SQL database but I am as yet undecided. To get myself more acquainted with the core concepts of the project (NetFlow, Big Data etc.), I have started to take some courses on plural sight – the notes from my first course can be seen in appendix1.

On October the 4<sup>th</sup>, I am due to present my idea to a trio of lecturers (Dominic Carr, Eamon Nolan & Anu Sahni) in order to receive the requisite approval. From reading the brief, we are allowed no slides (although we must submit a short slide deck which to be honest now seems pointless!!) and must simply convey the idea by giving a high-level description. In order to get this right, I plan to make my presentation short and to the point, covering the following:

- The Core Idea
- Reasons for my choice
- Technologies that I will utilize

I firmly believe that the topic is relevant, current and applicable to today's market needs. Furthermore, from my own point of view it will allow me to utilize some of the skills I have learned thus far in college whilst also engaging in and learning about new some new technologies. Most importantly, I believe that this project will indeed differentiate me from the norm and potentially offer something different come final submission. Here's hoping the lecturers feel the same way and give me approval to proceed.

### **15.1.2. October**

On the 4<sup>th</sup> of October, I was scheduled to present my project idea to a team of 3 lecturers in order to have it ratified and to allow me to proceed. I had handed up my short slide deck which gave a very high level overview of the core project concepts, technologies etc. and I was very comfortable in terms of my level of understanding. I had been doing some research already on things like NetFlow, Hadoop and was just touching on the Horton Works platform so I had very little concerns in relation to the project being rejected. In preparation, I went to the library for 6pm and wrote out some short notes to structure what I was going to say. Regardless of my understanding, I am very aware that how something is communicated and how you get it across to other people is key.

My presentation was scheduled for 19.30 and on entering the room I was introduced to the three lecturers Eamon Nolan, Dominic Carr & Anu Sahni. Although Eamon is the project supervisor this year, I had never had a class with any of these lecturers previously so they had nothing to base my abilities/capabilities/weaknesses around – this meant I needed to ensure to convey the idea with confidence and an appropriate level of detail. I explained the core concepts, technologies and my reasons for choosing the project in a little under 10 minutes. Feedback was very strong and all 3 recognised that I had put in a bit of preparation and that they were comfortable that I could press ahead. On leaving the room, Anu mentioned that rather than using a software application to generate my NetFlow data, I should look to get access to a Cisco switch to give the project a more authentic base. I replied by saying that if NCI were willing to buy me a switch I would be more than happy to!! However, considering that I am paying for a wedding at the moment I doubt I could stretch to that myself!!

Now that the project was ratified, the next major deliverable for the month was the Project proposal which was due for submission on October 21<sup>st</sup>. I had already made a start on this as I am keen this year to try to get things completed ahead of the submission dates. There was a project proposal template added on Moodle as well as some examples of previous project reports etc. This was to give us an idea of what was expected. I write technical documentation in my day to day job so am quite confident in my ability to deliver on the suite of documentation required. My marks from previous years in this space have been relatively good so this is something I was keen to maintain. As opposed to sticking with the pre-subscribed templates and actually went back and looked at previous project proposals I had done since 2<sup>nd</sup> year. I took all the headings from each as well as those from the Moodle template and put them into an excel document (see Appendix II). From here I decided to take the best headings and combine them to form the template structure for my proposal document.

Doing this bit of preparation helped focus my mind on what I intended on putting to paper and submitting so I found this a useful exercise. Once this was completed I set to populating the document as required and fleshing out each section. In truth, this took me longer than anticipated and I probably put more effort into this document than I should have – considering it was only worth 5% of the overall mark!! However, I also wanted to hand up something that would come close to full marks, gave a strong explanation of my intent and looked professional. I created and inserted a Microsoft Project Gantt chart that pretty much spelt out the various project phases (as outlined by our project coordinator) and drew out a topology of the solution using an online tool called draw.io. I found this tool very useful as not only were the icons and the various options of a high quality, but the work and associated changes you create can be automatically sync'd to Dropbox – which was very handy. This document was due for submission on the 21<sup>st</sup> of October and was submitted accordingly.

The next key point of the project appeared to be the assignment of our project supervisor. Once the college had received the full complement of proposals a list was placed on Moodle assigning a project supervisor to each student. There was plenty of discussion among the students around the actual function of project supervisors and the potential value they could/would add. In truth, the main discussions centered around who people were hoping they didn't get!! I was assigned Manuel Tova Izquierdo and promptly sent him a mail requesting an initial sit down. I was careful to list the days I would be available i.e. the days that I would be in the college, as some of my colleagues had mentioned that their supervisors got in touch and requested that they come to meetings on Mondays, Fridays etc. I felt this was unreasonable for evening students and that the supervisors should work around our schedules as opposed to the other way around. Thankfully when Manuel got back to me he was more than happy to facilitate our meetings on a Thursday, which was greatly appreciated. Our initial meeting was originally scheduled for Thursday October 27<sup>th</sup>, however Manuel dropped me a note to advise that it would need to be postponed due to personal reasons. This was completely understandable and I hope to sit down with him following the reading week. Having not met Manuel previously I am interested to get his feedback on my project proposal document and his personal views on my project concept. Hopefully he can offer some constructive feedback and ideas/structure(s) that I can potentially incorporate.

In terms of learning and research this month I completed an introductory video on Apache Spark which gave me some good insights into how the transformation and action processes on data can function. The demos showed this implemented using both Python and Scala. Whilst the data platform used was not Horton Works – it was an alternative known as Databricks – the fundamental concepts were the same. The presenter in the videos was very knowledgeable and broke things down well. This gave me a good starting point and some further confidence around the project concepts. My notes from this video can be found attached within the appendices.

### 15.1.3. November

On the whole November was an exceptionally difficult month. Not only was I under severe pressure at work, but November seemed to be the month in which most of our CA deliverables were due. We had to hand in a Strategic Management project right at the end of October, our Project requirements Specification was due on 18/11, we had a CA in Business Data Analytics and finally we had to produce a quite exhaustive CA for Business Intelligence and Data warehousing. That's not mentioning the Academic Paper for Security Principles and the Mid-Point project deliverables for 'Software project' – both due early in December. Needless to say, time and priority management were key to getting to the end of the month!

In terms of this project directly, the main deliverable was obviously the 'Requirements Specification' document. To be honest, to this point I had mixed feelings on what was actually expected from this deliverable. I had completed multiple RS documents for previous projects and had always received strong feedback, however there seemed to be a focus here on interviews and marketing strategies that I didn't really feel was relevant. However, not wanting to lose marks, I needed to take this into consideration. At this point I had still yet to meet my project supervisor (which itself was quite concerning as we were already past halfway through the semester). Thankfully our first meeting was scheduled for Thursday the 10<sup>th</sup> of November.

I met with my project supervisor, Manuel Tova Izquierdo and my expectation was that I would get some feedback on my project proposal and some advice around the best practices for completion of my RS and planning for the mid-term report submission. This meeting did not go as planned. Manuel explained that he had been very busy and had not had a chance to review my proposal, furthermore he was unable to offer much in terms of insight for my RS. He did however re-iterate the requirement for me to conduct the previously mentioned interviews to display a method of primary research. I argued against their value considering it is not likely that I will ever try to sell the end-product, however Manuel pretty much indicated that this was a necessity – adding another task to my already long list for November. We discussed the requirements for the mid-point presentation and again there seemed to be an expectation above what I was planning to deliver. However, I took this on the chin and said I would plough on. We agreed to meet again in one week to review my 'RS' prior to submission. Unfortunately, one week passed and Manuel let me know that he was unavailable to meet. I had put a lot of work in the 'RS' and would have liked some feedback, however I was forced to submit the document 'Supervisor unseen'.

The following Tuesday, I raised a few concerns with Eamon Nolan, the project coordinator, in relation to the Project Supervisor role and the difficulty I had experienced in getting some time from Manuel. I explained that I am not overly anxious about the lack of meetings however I wanted to ensure that I would get the relevant marks for attempting to arrange the meeting. He said this would not be an issue. He was also good enough to step through my RS document and give me some feedback and advice for the mid-term submission. Coincidentally I received a mail from Manuel that later that week asking if I was available for a conference call. We agreed and I took some time out from work to call him @10am. Thankfully we had a very good meeting and stepped through my 'RS' document as well as holding a detailed discussion on expectations for the mid-term presentation which had not been scheduled for Tuesday

13/12. We were able to agree the relevant details and discussed the wider scheduling issues. I was happy that any concerns I had were ironed out and I could proceed as planned.

In terms of research in November I continued my work around the Apache Spark platform and considered my data requirements a little further. I tested some basic functionality on the Hortonworks environment and made some hardware changes on my laptop to support the environment at an optimum level. I also made an initial attempt at a baseline asp.net front-end that I could potentially utilise for the proto-type as part of the presentation in December. Overall, November was an exceptionally challenging and stressful month, however at the end of it I had achieved what was needed and had made some good progress in terms of the project and the upcoming deliverables.

### 15.1.4. December

As previously mentioned, November was quite a taxing month. This continued into early December with my Security Principles Research paper and BI&DW project due on the 4<sup>th</sup> and 10<sup>th</sup> respectively. Whilst I had no problem with the security paper, I never want to see anything to do with that BI&DW course again. What I initially thought would be an excellent and relevant course was both poorly planned and even more poorly executed. Getting that project done was of significant relief and a large weight off my shoulders. However, at that point I realised that I needed to start to plan-out the next steps in my project and do some preparation for the mid-term presentation. Initially I had been told that the presentation dates were the 19<sup>th</sup> & 20<sup>th</sup>; however due to lecture availability (or lack thereof) my own was moved to the 13<sup>th</sup>.

Having discussed this in detail with Manuel, I essentially had to deliver three things:

- 1) A strong presentation clearly outlining the Project
- 2) A prototype of the end product
- 3) Marketing Website (Draft)

Out of the above three, the most challenging would be the prototype itself because at this point I had nothing to really 'prototype'. All of my work thus far had been focussed on getting a strong understanding of the back-end infrastructure and constructing the relevant documentation in terms of my deliverables. My discussion with Manuel had focussed a lot on this and he agreed that once the prototype represented a vision of the end deliverable, it was ok to use dummy data as opposed to a working solution. This point made the process easier, as effectively all I need to produce was an ASP.net front end with some basic functionality. To compliment this, I wanted to demonstrate a draft of my marketing website in order to show something additionally tangible on the day. In terms of the delivery of the presentation, my understanding of the end goals and the work completed so far (project proposal, SRS, technical report) meant that I was relatively comfortable in this space.

I set out a few days and began the process of working through my application build. I have to say, compared to documentation and my more recent deliverables in other subjects, I found this quite refreshing. To sit back and actually build something took away some of the tedium from the past few weeks. Unquestionably, however this process was equally draining. There were a number of long nights in getting the application to where I wanted it to be as well as

putting in a few hours here and there to create the accompanying website. Those few days did not sit well with my other half but in terms of preparation, I was confident that I now had enough to present and hopefully gain a good mark from the reviewers. In the end I had produced the following for submission/review:

- Technical Report
- ASP.Net Web Application
  - Login/Registration
  - 4 Navigable Pages
  - SMS Functionality (Click to Text)
  - Sample Graphics
- Marketing Website (Bootstrap)
- Mid-Term Presentation (PowerPoint)

On the day of the presentation, I arrived early in order to get everything setup on the local classroom PC and to ensure that all of the above were accessible. Thankfully, I had no issues. Aside from a small annoyance around the resolution (the website did not render as nicely on the college PC), I was able to produce everything fairly easily. I had everything ready to go when Manuel arrived but it was only at that point that I realised I had forgotten the printouts of both by report and presentation! These were in no way essential, however I always try to bring printouts as they tend to look professional and add an additional element to any presentation. Unfortunately for me, they were now sitting in the print room of my place of work!! I made a point of mentioning this once Eugene McLaughlin (the second reviewer) had entered the room; however neither he nor Manuel were overly worried about that aspect.

I started things off with the presentation and ran through each section in quite a bit of detail. Both reviewers listened intently and took notes as I went through each slide. There was not much interaction apart from the odd question as I moved through and I took this as a good thing. I spoke for a good 20 minutes before Manuel asked to see the application. I made of point of running the application from Visual Studio in order to demonstrate that it was indeed my own code and thankfully, the application started up without issue. I walked through the various pages and demonstrated some of my base functionality i.e. printing reports from screen, sending a text to my phone and I logged in under my own account. We then moved through the marketing website, where both Manuel and Eugene got a laugh out of me including them in the 'Contact Us' page – complete with their pictures stolen from LinkedIn! At this point I completed the presentation and opened the floor to questions.

Eugene asked a pointed question relating to the legal side of putting such an application into a live environment and the direct repercussions of a failure. I responded by speaking to the requirement for a detailed contract, master services agreement and SLA which would map out in no uncertain terms the functionality, penalties etc. I thought this was an excellent question and something I had not prepared for. However, through my job, I have experience in this space and was happy with my retort. Manuel focussed on my future plans how I was structured in relation to implementation. I spoke around my requirement to get a single stream of information flowing and to work from that point. I truly believe that once I have one stream working, the rest will follow quite quickly. From a feedback perspective, that was it. Both were



very complimentary and thankfully, that was reflected in the strong mark that I received. I took a break from the project for the remainder of the month as it was now time to focus on birthdays (my own and my daughters on the 18<sup>th</sup>), Christmas, as well as the upcoming January exams.

### 15.1.5. January

Following my presentation, I only had my second Business Data Analysis CA to submit before I finished the semester. I got this finished and submitted prior to the deadline so was happy to put my college work to bed until after Christmas. January began as it always does – Exams! Leaving all college work until after Christmas was pretty much a necessity (I was shattered) but coming into January that feeling of dread was kicking in regarding the exams. I had focused purely on my day job in the run up to Christmas and now looking through notes etc. I realized I had an uphill battle to get through everything. In truth, this has been the same every Christmas time. I find it difficult to get back into college work after Christmas and it is almost cruel to have exams so quickly after the break.

We had received our timetables well in advance and in truth I was not overly happy with what I saw! We had the dreaded Business intelligence and Data Warehousing up first and then the following week we had 3 exams in 4 days! You would think the college could have spaced them out a little better, however after 4 years I have given up on looking for method to their madness!! The exams were not easy, actually they were more difficult than I could have expected. I got through them but was thoroughly exhausted throughout. I genuinely struggled with a post Xmas hangover and studying was more of a chore than any other year I can remember. It didn't help that I was not a big fan of the subjects but I genuinely found it difficult to get the right level of motivation.

On getting through the exams you always get a welcome break before returning to class. This year it was only a week, but it still felt good to take some time to decompress. Soon after the exams the timetables arrived and I noted that we had a total of 3 modules alongside the project. This was one less than last semester so in theory should present me with a better opportunity to focus an appropriate amount of time on my project. Bar one subject (Business Data Analysis II), the semester looked heavily theory based and after taking the other two subjects (Business process Management & Information Systems Management) I could definitely confirm that. I was shocked at the IMS module as it seemed a very strange one for 4<sup>th</sup> year students. We had a similar module in first year called 'Management Foundations of information Systems' and I feared this was going to be a repeat of same. Thus, far I believe that will be the case. The content appears to be quite basic and focused on things we have learned previously. Laughably, the lecturer actually asked us if we were 1<sup>st</sup> years when I attended the first class!! I decided that rather than sulk, I would just put the head down and get through everything. After all, I can see the light at the end of the tunnel now. 4 more months of hard work and it's all over!! This is a shorted entry than usual and reflects the level of work completed in January outside of exams!! Let's hope February is more productive!!

### 15.1.6. February

February rolled around and to be honest it was at this point that I really started to think about my project. Although I was happy with my progress before Christmas – and the marks I had received thus far – I had a nagging feeling that there was a lot of work to do in order to successfully deliver on my commitments!! The month started with us getting our results for Semester1 – as usual as I was a little disappointed. Some of my results were expected but I was shocked at my Business Data Analysis exam result as I had done very well in the CA's. My overall average was still close to where it needed to be but that result really grated me. I spoke to a few people in the college about a review but didn't seem to be getting anywhere. Rather than focus on the negative I decided to move forward and just focus on the coming months and the new subjects. If necessary I could look to get this reviewed at the year's end.

I started out this month with best intentions and some real determination to get my project moving. Things unfortunately didn't really go to plan. We were given a tough CA in 'Advanced Business Data Analysis' and this required a significant amount of time and effort. Coupled with an increasing workload in my day-job, finding time to really work on my project was proving increasingly difficult. Furthermore, the whole project supervisor idea was causing its own issues! Neither myself nor Manuel could agree on a time/date to hold some catch-up meetings. Times that suited him were typically during my working hours and times that suited me were typically when he was not in the college. In the end, we agreed – over email – that rather than hassle each other for meetings, I would instead send him some regular email communications updating him on my progress. So long as this did not affect the project marks I felt very comfortable with this approach! It was a bit of a relief to get this one sorted as these engagements are worth 5% and at this stage of the year, every mark counts.

As mentioned, I spent a significant amount of the month working through another CA, however the one area I did focus on in terms of my project was planning out some tasks and getting very clear in my head what the associated timelines for each one amounted to. With only a few months left I felt that actual tasks breakdowns and prioritisations were essential to me actually structuring out the required workload. Items that I took into consideration were as follows:

- Configuring the IDE environment (Eclipse)
  - Installing the application
  - Configuring environment variables
  - Testing
  - Etc.
- Creating a dedicated GitHub Repository
  - Java Code
  - Bootstrap Code
  - Version Control
- Creating a localised Spark Streaming Session
  - Generating of RDD's
  - Filtering Operations
  - Transformation Operations

- Transferring localised Sessions to HortonWorks
- Marketing Website
  - Finalise Look/Feel
  - Create Marketing Video
  - Test Amazon S3 Hosting Service

The above are some of the high-level items I collated and towards the end of the month, these tasks (and more) were uploaded to an online Trello account. Trello is an open source, online tool for the KanBan project management methodology. As mentioned, I felt that the prioritisation of tasks was essential and uploading them into Trello allowed me to categorise and manage tasks in real-time.

Before the month ended I successfully created my development environment (I had never actually used Eclipse before!!!) and tested several sample Spark Streaming applications on my local environment. At this point, my aim is to complete and end to end stream of NetFlow data before the end of the month. If I can complete this task I will be very well position to deliver on my initial requirements specification. Till next month.....

### 15.1.7. March

Coming into this month I had a bit of momentum following some solid work in the previous week. It had been a long time since I had coded in Java – and even at that we were only really doing the basics – so to get some basic code up and running on Apache Spark was certainly a positive step. The next key milestone was to apply this to my project and actually write some code to read in and manipulate/filter a live 'NetFlow' packet. As usual with programming, my initial enthusiasm was soon drained out of me!!!! I had done a lot of studying around NetFlow so was comfortable that I understood the layout of the packets and the information that I needed to acquire. Parsing the live packet however became quite a task. I have mentioned previously that in order to simulate NetFlow traffic I would be using a software based NetFlow generator (Paessler NetFlow Generator). I set about creating an instance of this generator and using Wireshark to identify the traffic as it was being generated. I ran into the following issues – which I thankfully resolved:

- 1) Aiming packets at the localhost will not show up on Wireshark!!
- 2) Selecting UDP packets will only show you specific UDP data (not necessarily NetFlow)
- 3) NetFlow packets within Wireshark must be decoded using 'CFLOW' protocol

Below are some screen shots of NetFlow packets successfully being picked up within WireShark:

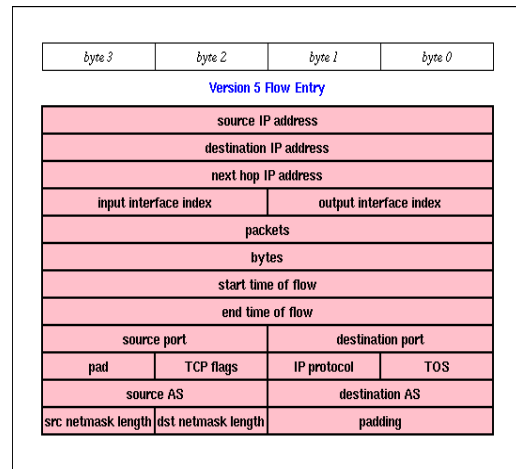
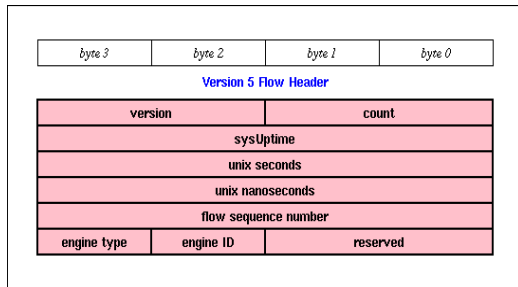
No.	Time	Source	Destination	Protocol	Length	Info	Dst
521	17.632425	192.168.223.76	192.168.223.77	CFLOW	114	total: 1 (v5) flow	9996
773	27.632983	192.168.223.76	192.168.223.77	CFLOW	114	total: 1 (v5) flow	9996
1017	37.636244	192.168.223.76	192.168.223.77	CFLOW	114	total: 1 (v5) flow	9996
1483	47.640586	192.168.223.76	192.168.223.77	CFLOW	114	total: 1 (v5) flow	9996
1775	57.645869	192.168.223.76	192.168.223.77	CFLOW	114	total: 1 (v5) flow	9996

Below is a deeper breakdown a one of these packets (also within Wireshark):

```

> Frame 521: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
> Ethernet II, Src: IntelCor_74:b6:c0 (c4:d9:87:74:b6:c0), Dst: Cisco_d9:87:c0 (00:25:b4:d9:87:c0)
> Internet Protocol Version 4, Src: 192.168.223.76, Dst: 192.168.223.77
> User Datagram Protocol, Src Port: 62598, Dst Port: 9996
v Cisco NetFlow/IPFIX
  Version: 5
  Count: 1
  SysUptime: 79814.234000000 seconds
  > Timestamp: Jun 6, 2002 18:16:48.-2133291404 GMT Daylight Time
  FlowSequence: 3268667600
  EngineType: Unknown (254)
  EngineId: 255
  11.. .... .... = SamplingMode: Unknown (3)
  ..11 1111 1111 1111 = SampleRate: 16383
  > pdu 1/1
0000 00 25 b4 d9 87 c0 c4 d9 87 74 b6 c0 08 00 45 00  .%. .... .t...E.
0010 00 64 34 30 00 00 80 11 c6 6d c0 a8 df 4c c0 a8  .d40.... .m...L..
0020 df 4d f4 86 27 0c 00 50 26 a9 00 05 00 01 04 c1  .M...'..P &.....
0030 de 5a 3c ff 99 00 80 d8 8e 74 c2 d3 e8 d0 fe ff  .Z<.... .t.....
0040 ff ff 0a 00 00 01 0a 00 00 02 00 00 00 00 00 00  .....
0050 00 00 00 00 00 00 00 00 03 e8 04 c0 f3 fa 04 c1  .....
0060 de 5a 03 e8 00 50 00 1f 11 fa ff ff ff ff 00 00  .Z...P.. ....
0070 00 00  ..
  
```

Whilst the above seem like small items, understanding the issues, validating them and finally getting to a point where I could identify and visualise a successful packet delivery (as above) was quite tedious! Once I got over that hump however it did allow me to understand the packet structure, the byte layout etc. This gave me my initial starting point for how I could write code to read and stream the packets in an appropriate manner. One thing I did already know was that there is plenty of information out there in relation to Apache Spark streaming so I set about googling similar tasks/problems and reading through 'Stack Overflow' etc. I was able to find some coding examples that were quite similar in terms of receiving stream of data. I then set about manipulating these and writing my own code the tailor it to the type of data I would be sending through. A basic NetFlow packet is broken up into a 'Header' and a subsequent 'Flow data' as shown below:



The challenge was to read in a stream of this data and to ensure that I could break it up into its constituent parts. As below (taken from Eclipse), I have successfully ingested a stream from the NetFlow generator, split it into its relevant bytes and printed the information:

```

<terminated> SOSThreatAnalytics [Java Application] C:\Java\jdk1.8.0_121\bin'
Version 5
Count 1
system up time 80368937
Epoch 1023383808
Nano 2161675892
Flow 3268667600
Engine Type 254
Engine Id 255
Sample SD 510
Src IP 10.0.0.1
Dest IP 10.0.0.2
Next Router Hop 0.0.0.0
Inbound snmpIFindex 0
Outbound snmpIFindex 0
Packet Count 0
Byte Count 1000
Time at Start of Flow 80308937
Time at End of Flow 80368937
Source Port 1000
Destination Port 80
Padding 0
TCP Flag 1f
Protocol 17
    
```

The next challenge in this space will be to filter the information according to what I want to take out, inject this data into an object and successfully insert an entry into a SQL database (that should be challenging!!). To this point I have a working connection from Java to SQL Server using a 'JDBC' connector and will be focused on developing this in the coming weeks.

In parallel I set about building and launching my proposed Marketing Website. I had created the site locally for the purposes of my mid-point presentation and was happy with this as a starting point. However, I had not really looked at it since and aside from wanting to update

it (social media, color schemes etc.) I also needed to look at migrating it to a live environment. I had set out in my mid-point report that I would use Amazon S3 to host the site – however in truth I had done very little investigation into how I would go about this. I had heard however that it was quite intuitive. I set about creating an AWS account within the Amazon environment and followed a detailed YouTube video around the ‘static website creation’ process. Thankfully, this was indeed quite trivial. Apart from the tedium of uploading every single file (css, images, bootstrap etc.), once I navigated my way through the site, the layout and the concept of AWS ‘buckets’ I was able to generate the site online and create a home page:

<https://s3-eu-west-1.amazonaws.com/www.sosthreatanalytics.com/Index.html>

Whilst not the most attractive web address it will do for the moment. There are options to change this and create my own DNS and I will look to do this once I have finished updating the site. Whilst it was quite handy to do in the end, this was another good milestone to get through. Regardless of updating the sites look & feel, the next remaining challenge in this space is around the concept of SEO. I am probably not going to look into this too much until next month; however, it will be another good addition to the site and is something I certainly plan on completing.

In terms of ongoing project management, I mentioned last month that I had created a ‘Trello’ account for project management and task prioritization. Whilst I was comfortable with Trello I was able to acquire a free license for a better application – Kanban Flow – so I have since migrated my task list across. Again, I feel it is important to visualise my tasks and appropriately manage their priorities in order to ensure that my focus is right each day/week. There are only a few weeks left and with another outstanding CA and 3 exams – it would be easy to lose focus.

Aside from all of the above, I attended a project class with Eamon Nolan in which we were directed to updating an online profile giving some high-level information about our projects. We also had our pictures taken and were asked to create a LinkedIn page. I already had an existing page for work purposes, so was able to re-use this one. I updated the profile page almost immediately in order to get this out of the way. Furthermore, we are now required to create a marketing poster for the project – this is due 08/05. I will start to think about that at some stage next week. All in all it has been a successful month in relation to the project and I feel that I have made some good progress. There is still an incredible amount of work to be done and it’s going to be very tight to deliver a final project that I believe is in line with expectations. I am however determined to put as much effort as possible into the remaining weeks and to put my best foot forward. After 4 years of effort to this point, it would be stupid not to!



### 15.3 Requirements Elicitation – Questionnaires

User Interview - Greg Shelly: IT Service Delivery Specialist		
Question	Response	Comments
Can you describe your current job role & responsibilities	Yes	I am currently employed in Goodbody, where I work as a Service Delivery specialist. This is a challenging and broad role where I have responsibilities in Vendor management, Incident management and take a lead role in internal business Relationship Management program. I am also part of the firm's BCM team. I have been operating in this role since mid 2011.
Can you give a brief overview of your previous experience in the industry?	Yes	I was originally employed in Accenture as and IT analyst, I moved to AIB to perform a similar role. I worked in support in both the ROI and Capital Markets groups respectively. I moved to Goodbody in 2011.
Do you have experience in networking - security, monitoring or otherwise	Yes	I am responsible for the oversight and governance of our infrastructure managed service provider. Part of that involves the management and availability of the underlying WAN & LAN infrastructure. It is critical that I understand the various components of this configuration as well as the risks associated with it. Network monitoring is something we work closely with the vendor on.
Have you read the Project Proposal?	Yes	I completed a review of the project proposal in advance of this meeting as requested.
Do you understand the core concept and goals of this project?	Yes	I understand the core concepts and overall aim of the project.
As a user what are the fundamental requirements/functionality you would expect to be delivered	N/A	User Login, Alerting and notifications, charts and graphics if possible. I would also like the ability to contact support if I have an issue
Do you understand the technologies?	Yes	I am not familiar with the back-end infrastructure and have no development background to speak of. The technologies are familiar to me by name through my work in the industry.
Have you any previous experience with these technologies?	Yes	I have no previous experience in using these technologies - from a development perspective. However from a user perspective I would have encountered many ASP.net front-end applications
Do you agree with the approach from a technology perspective?	Yes	Not being overly familiar with the technologies hampers me a little in terms of challenging the approach, however the proposal outlines the purpose of the technologies well and the overall approach appears to be sound and well thought-out.
Do you feel there is a market for this project?	Yes	The proposal document makes a clear case for the requirement of monitoring on the underlying network and as such I believe the market would be there. I look forward to getting a look at the product and evaluating whether or not it would be something that could be put to use in a functioning enterprise environment.
Is this something you would purchase / be interested in?	Yes	Potentially, but it will depend on the level of detail provided.
Would you be willing to be a 'software tester for an early version of this application?	Yes	Yes, I would be very interested and would be strongly engaged if I get an opportunity to evaluate



User Interview - Ian Dempsey: Network Management		
Question	Response	Comments
Can you describe your current job role & responsibilities	Yes	I work in a network operations team in BNY Mellon. I am responsible for a team of people that cover the network operations centre. Responsibilities include incident management and service availability reporting. I also work closely with the network security team.
Can you give a brief overview of your previous experience in the industry?	Yes	Prior to this I worked in operations in Pershing - a BNY mellon company. My roles there included IT Relationship Management and IT Support
Do you have experience in networking - security, monitoring or otherwise	Yes	This is the basis of my current role in BNY Mellon. Managing the integrity of our networks includes service monitoring, liaising with security teams and creation of relevant service statistics etc.
Have you read the Project Proposal?	Yes	Completed in advance of this questionnaire
Do you understand the core concept and goals of this project?	Yes	Completely, again this type of activity/technology is something I am around on a day to day basis
As a user what are the fundamental requirements/functionality you would expect to be delivered	N/A	As this is an online site I would obviously like a URL that is accessible and always available. I would like my own user account and ability to change mobile numbers etc. for notifications. The application should be easy to use and present the relevant details to the user in a clear manner. I would be interested in seeing volumes or port hits (particularly unsecure ports) and volume reports that would alert after a certain threshold is hit.
Do you understand the technologies?	Yes	I am unfamiliar with the HortonWorks data platform and have little knowledge of database technologies, however I understand the role of each component in the overall solution
Have you any previous experience with these technologies?	Yes	I am familiar with network monitoring software as I would come across various types through my role in BNY Mellon. We have previously used technologies such as solarwinds, Cat Tools etc. I was also aware of the NetFlow protocol used by Cisco devices
Do you agree with the approach from a technology perspective?	Yes	The proposal breaks it out pretty clearly and provided the technologies can talk to each other within the Azure Cloud, the platform as a whole seems to make a lot of sense.
Do you feel there is a market for this project?	Yes	I would be interested to see the final product as I do believe there is a market for this if it was to be marketed in the right manner. A lot of these tools tend to be expensive and overly complex. A lightweight and cheaper alternative could corner a large piece of the market if promoted right.
Is this something you would purchase / be interested in?	Yes	See above, not sure if it would suit a large scale environment but the idea is something I would certainly recommend and encourage. Longer term development and funding could enhance functionality and change all of that.
Would you be willing to be a 'software tester for an early version of this application?	Yes	Absolutely

User Interview - Chris Moran: IT Project Management and Application Support		
Question	Response	Comments
Can you describe your current role & responsibilities	Yes	I am currently employed in London doing contract work on various projects with 'Transport for London'
Can you give a brief overview of your previous experience in the industry?	Yes	I've held various roles over the years including application support in AIB, IT Service Desk in Norwich Union, IT Service Delivery in Goodbody, application development in Aegon and this is my 2nd stint in Transport for London. I also worked as IT Manager for Arriva in London.
Do you have experience in networking - security, monitoring or otherwise	Yes	The only role that I was directly involved in Network security and monitoring when during my time in Arriva. As IT manager I was a jack of all trades and so became quite familiar with the networks team and the tools they utilised.
Have you read the Project Proposal?	Yes	Completed.
Do you understand the core concept and goals of this project?	Yes	I do.
As a user what are the fundamental requirements/functionality you would expect to be delivered	N/A	I would like the ability to create multiple accounts so that the application could be accessible for a number of users in a team. If certain users could only see certain elements that would also be preferable. The application should be user friendly and have a consistent look and feel. As a team leader I would like to be able to create and manage each account
Do you understand the technologies?	Yes	The proposal details a list of technologies and I have done some quick research on what I was not familiar with. The front-end application and underlying database elements would be very familiar to me. In saying that I have not worked in an organisation that used a valid NoSQL implementation.
Have you any previous experience with these technologies?	Yes	I have extensive experience with ASP.net as I have worked in development teams that produced internal web applications utilising this MVC framework. Similarly I would have a lot of experience in relational database via SQL server. The HortonWorks platform and Apache Spark elements I would not have encountered.
Do you agree with the approach from a technology perspective?	Yes	Having not worked with a NoSQL database it would be great if that part could be cut out and the data could be streamed directly to SQL server. However not being familiar with the HDP platform I am unsure of how this would work. Aside from that the approach appears to be very well structured.
Do you feel there is a market for this project?	Yes	Yes, security and network security in particular is becoming ever more topical in all aspects of business. The more information we can glean for the network the more aware we become to threats etc.
Is this something you would purchase / be interested in?	Yes	Yes, the notifications are particularly of interest to me.
Would you be willing to be a 'software tester for an early version of this application?	Yes	No problem

User Interview - Donal Mackey: SQL Developer		
Question	Response	Comments
Can you describe your current role & responsibilities	Yes	I currently operate as a SQL Developer for Fidelity Investments, located in Galway. My role is to work with the DBA and local development team to manage and maintain the database estate whilst performing extensive MIS reporting in line with internal business requirements
Can you give a brief overview of your experience in technology business to this point?	Yes	My first role in technology was in Goodbody. I was part of the IT Operations team, responsible for the administration and governance of all application access levels. From there I moved into the development team within the same IT department. My focus was on MIS reporting and I worked with the local DBA. I left in 2013 to join Fidelity in a similar role
Do you have experience in networking - security, monitoring or otherwise	Yes	Not directly but working in IT I am aware of the terminology and necessity to manage the network
Have you read the Project Proposal?	Yes	I have.
Do you understand the core concept and goals of this project?	Yes	Yes, this is quite clearly explained.
As a user what are the fundamental requirements/functionality you would expect to be delivered	N/A	<ul style="list-style-type: none"> <li>- User Registration &amp; Login</li> <li>- Alerting &amp; Notification (text would be ideal)</li> <li>- Ability to change my profile (alerting details)</li> <li>- A page per category i.e. ports, volumes, protocols etc.</li> <li>- A clear and navigable layout</li> </ul>
Do you understand the technologies?	Yes	I am very familiar with the database technologies - more so the SQL server element but I would not be familiar with the others
Have you any previous experience with these technologies?	Yes	I've used Microsoft Azure as we have access via our organisational accounts and as above I use SQL server and other such technologies quite regularly.
Do you agree with the approach from a technology perspective?	Yes	The approach appears to be thought through and there is a logical flow to the end goal. The back-end is quite intriguing from a technology perspective
Do you feel there is a market for this project?	Yes	Yes, I believe there could be.
Is this something you would purchase / be interested in?	Yes	Not personally but I would certainly know of a number of companies that are not managing their internal network and that could utilise this level of monitoring
Would you be willing to be a 'software tester for an early version of this application?	Yes	Yes

User Interview - Danny O'Callaghan: Java Developer & Solutions Architect		
Question	Response	Comments
Can you describe your current role & responsibilities	Yes	I'm currently employed in Goodbody as senior Java Developer and Solutions Architect within the IT department. I am responsible for the online trading website and all interfaces between our OMS Trading system and back-end settlement and clearing system. I also have lead roles in Security and mobile technologies.
Can you give a brief overview of your previous experience in the industry?	Yes	I have worked as a contractor for many years developing java applications and managing front-end enterprise websites. I have worked in secure development focussing on web security standards such OWASP and Mitre. I am also a certified ethical hacker.
Do you have experience in networking - security, monitoring or otherwise	Yes	I have extensive experience in technology and whilst not working in a network management role I am very well up to speed on network security and the associated pitfalls. I have worked with tools such as Alert Logic in the past
Have you read the Project Proposal?	Yes	I reviewed the proposal in detail as this is an area in which I would have a keen interest
Do you understand the core concept and goals of this project?	Yes	I have a strong understanding of the project concept and end goals
As a user what are the fundamental requirements/functionality you would expect to be delivered	N/A	<ul style="list-style-type: none"> <li>Administration: Create an Account, Login, Update user details</li> <li>Navigation: simple &amp; intuitive, multiple pages (per protocol, port etc.)</li> <li>Alerting: text messaging as well as email alerts</li> <li>Monitoring: traffic analysis by volume, port, malicious IP</li> <li>Reporting: I would like to see an ability to create reports to excel i.e. maybe in '.csv' format</li> </ul> <p>I'm not sure what the time limitations will be considering this is a college project but to me the above is desirable. If you could have different access levels for users that would be a plus.</p>
Do you understand the technologies?	Yes	I am familiar with all aspects of the technologies in use and have previously researched 'Apache Spark' as part of a NoSQL project I was engaged in. I was not initially familiar with the Horton Works data platform, however on reading the proposal I have researched and understand its function in the overall environment. I am familiar with ASP.net however I would not be an active developer in this space.
Have you any previous experience with these technologies?	Yes	As a developer, I have obviously engaged with front-end applications and back-end databases extensively. I have experience in the majority of current programming languages e.g. java, c#, ruby on rails, python, Scala etc. Whilst not a massive fan of the ASP.NET MVC platform - or MVC's in general, I can understand the choice of tool here considering your experience levels and previous college work. As I am responsible for the Goodbody website, I have previously evaluated NoSQL databases with a view to planning for the future and presumed increases in traffic & data consumption.
Do you agree with the approach from a technology perspective?	Yes	Broadly I would agree with the structure and in particular the Cloud Hosting aspect. As previously mentioned I am not a fan of ASP.net but once you can achieve your goals it should be more than fit for purpose
Do you feel there is a market for this project?	Yes	It will be interesting to see how the project develops but the core concept is certainly something that would be attractive to companies looking to learn more about their network activity
Is this something you would purchase / be interested in?	Yes	There are alternatives on the market and I appreciate that time is limited in terms of turning around an enterprise ready version of this application! With further development and a focus on security, user segregation etc. this could be a very viable product. However it would require significant investment and resourcing to reach that point.
Would you be willing to be a 'software tester for an early version of this application?	Yes	