# IDENTIFYING AND SCHEDULING DDOS-APPLICATION LAYER ATTACKS IN MARINE, OIL & GAS EXPLORATION ASSETS ON-BOARD SYSTEMS LIKE PMS, SENSOR BASED DATA SYSTEMS

(DOI No: 10.3940/rina.ijme.2019.a3.539)

**Shaik Yaseen Baba** and **P S Avadhani**, Andhra University, India

## SUMMARY

For the cost effective & safe operation of ships and other marine assets it is mandatory to develop software solution tool which helps in timely maintenance with priority based to avoid both financial losses and operational downtime. Our idea is to propose concept to develop combined protection mechanisms system for Planned Maintenance system. The research is IDENTIFYING AND SCHEDULING DDOS-APPLICATION LAYER ATTACKS on onboard systems. Countering Distributed Denial of Service (DDOS) attacks are becoming ever more challenging with the vast resources and techniques increasingly available to attackers. In this paper, we consider sophisticated attacks that are protocol-compliant, non-intrusive, utilize legitimate. Application-layer requests to overwhelm system resources. We characterize application layer resource attacks on the basis of the application workload parameters that they exploit. Request flooding, asymmetric, repeated one-shot. To protect marine software-based servers from these attacks, we propose a counter-mechanism that consists of a suspicion assignment mechanism and a DDOS-resilient scheduler, DDOS Shield. In contrast to prior work, our suspicion mechanism assigns a continuous value as opposed to a binary measure to each client session, and the scheduler utilizes these values to determine if and when to schedule a session's requests. This will be done through an integrated working of PMS and Inventory. PMS and Inventory, while performing definite tasks independently, will seamlessly integrate with each other. Further the installations will reside in the vessel, office and other office nodes, where information can be viewed and updated depending on your network of vessels. In office, the Inventory-PMS package will function in a client –server mode and in a single terminal on the ship with LAN for the purpose of accessing Internet. All the database updating and back-up maintenance shall be shown into the system to enable the user to do the database management without incurring exorbitant annual maintenance bills, which normally comes with all similar systems in the market.

## NOMENCLATURE

| | |
|---|---|
| *WSN* | Wireless Sensor Networks |
| *DDOS* | Distributed Denial of Service |
| *PMS* | Planned Maintenance System |
| CAD | Computer Aided Design |
| CAM | Computer Aided Modeling |
| IP | Internet Protocol |

## 1. INTRODUCTION

In absence of Decision Tool and Maintenance Management

- Poor Information Availability & difficult to control in office
- Decision tool helps in prioritizing the activities and considering external factors
- Un necessary Expenditure
- Operational downtime causes both production and financial losses
- Ships will lose attention in addressing classification societies requirements
- Loss of man-hours and unutilized resources

All Marine and offshore assets are designed as per the Rules & Guidelines of any IACS society [1]. Note that there are four main categories of regulations: International regulations, Regional regulations, National regulations, Local Regulations. Let us try to make a list of regulations in reference list [2, 3, 4, 5, 6, 7, and 8], conventions, and guides which should be followed across the world by assets operators.

In ships and marine assets, PMS will keep a record of equipment, their maintenance schedules, spares, and all required database, which is relevant to carry out the planned Maintenance and unplanned ones as well.

Denial-Of-Service (DOS) and Distributed-Denial-Of-Service (DDOS) attacks pose a grave danger to Internet operation. They are, in essence, resource overloading attacks. The goal of the attacker is to tie up a chosen key resource at the victim, usually by sending a high volume of seemingly legitimate traffic requesting some service from the victim. The over consumption of the resource leads to degradation or denial of the victim's service to its legitimate clients.

In the absence of effective defense mechanisms, the Denial-Of-Service effect lasts for the entire duration of the attack (i.e., as long as key resources are being tied with malicious traffic), and vanishes quickly once the attack is aborted. Since machine resources are usually shared among many applications, the DOS effect inflicts significant damage not only on client transactions with the victim, but on the victim's total operation. The victim experiences a significant slowdown in all applications sharing the targeted resource, and frequently also connectivity disruption.

Both DOS and DDOS attacks are seemingly simple in design and operate with-out requiring any special skill or resource for their perpetration. The attack tools can be obtained easily online and the attack goal (resource exhaustion) is attained whenever a sufficiently large amount of malicious traffic is generated. The targeted resource dictates the type and contents of attack packets.

The main difference between DOS and DDOS attacks is in scale DOS attacks use one attack machine (to generate malicious traffic) while DDOS attacks use large numbers of attack machines. The scale difference also invokes differences in operation modes. The large number of attack machines allows DDOS perpetrators certain recklessness – they frequently trade sophistication for brute force, using simple attack strategies and packet contents to overload victim resources. However, the simplicity in both attack types arises from convenience, not necessity. The lack of effective defence mechanisms, even for simple attacks, offers no motivation for perpetrators to design more sophisticated ones. Once defences successfully counter one attack class (e.g., like ingress filtering [FS00] has countered random IP source spoofing), attackers quickly deploy slight modifications in their attacks to bypass defensive actions.

There are several features of DDOS attacks that severely challenge the design of successful defences:

- Use of IP source spoofing.
  Attackers frequently use source address spoofing during the attack they fake information in the IP source ad dress field in attack packet headers. One benefit attackers receive from IP spoofing is that it is extremely difficult to trace the agent machines. This, in turn, brings several dire consequences. Since agent machines run a very low risk of being traced, information stored on them (i.e., access logs) cannot help to locate the attacker himself. This greatly encourages DDOS incidents. Furthermore, hiding the address of agent machines enables the attacker to reuse them for future attacks. Last, as attack packets carry a wide variety of addresses, they appear as if they come from many disparate sources; this defeats fair sharing techniques that are a straight forward solution to resource overloading problems. The other advantage that IP spoofing offers to the attackers is the ability to perform reflector attacks [Pax01]. The attacker requests (in the victim's name) a public service that generates large replies to specific small size requests (amplification effect). The attacker generates as many requests for service as his resources permit, faking the victim's source address, and sends them to public servers. These servers direct a many fold volume of replies to the victim (thus reflecting and multiplying the attack force) and overload its 11 resources. A common case of reflector attack is described. The attacker sends a large number of UDP based DNS requests to a name server using a spoofed source IP address of a victim. Any name server response is sent back to the spoofed IP address as the destination. Because name server responses can be significantly larger than DNS requests, there is potential for bandwidth amplification. Even if the trace back problem1 were solved, it would not help to address reflector attacks. The public servers are unwitting participants whose legitimate service is misused in the attack. They possess no information about the attacker. Also, their service cannot be disabled (i.e., to stop the attack) as this would inflict damage on numerous other clients. Depending on these servers' resources and the request volume, they could prevent reflector attacks by limiting the number of replies they are willing to generate to a particular IP address. This approach would require servers to cache requesting addresses, thus potentially consuming significant memory resources.

- Large number of agent machines
  Even if trace back could be successfully performed in the face of IP spoofing, it is difficult to say what actions could be taken against hundreds or thousands of agent machines. Such a large number prevents any but crude automated responses aimed at stopping attack flows close to the sources.

- Similarity of attack to legitimate traffic
  Any type of traffic can be used to perform a successful denial-of-service attack. Some traffic types require a higher attack volume for success than others, and attack packets of different types and contents target different resources. However, if the goal is simply to cripple the victim's operation, it can be met by sending sufficiently large volumes of any traffic and clogging the victim's network. Attackers tend to generate legitimate like packets to perform the attack, obscuring the malicious flow within legitimate traffic. Since malicious packets do not stand out from legitimate ones, it is impossible to sieve legitimate from attack traffic based purely on examination of individual packets. A defence system must keep a volume of statistical data in order to extract transaction semantics from packet flows and thus differentiate some legitimate traffic (e.g. belonging to lengthy well behaved transactions) from the attack traffic.

  There are many attack variations and many dimensions in which attacks can still evolve while preserving the ability to inflict damage on the victim. This feature makes it very challenging to design successful defences. Due to attack variety, defence systems must maintain a volume of statistical data in order to detect attacks and sieve legitimate from attack traffic. This incurs high operation costs.

  On the other hand, attackers can easily bypass or trick defences with slight modifications to their attacks. Any such modifications require added complexity in defence mechanisms (in order to handle the new attack class), thus skyrocketing the cost.

This Project is to propose a mechanism for protecting the servers from Distributed Denial of Service (DDOS) attacks. We propose a counter mechanism that consists of a suspicion assignment mechanism and a DDOS resilient scheduler, DDOS Shield. In contrast to prior work, our suspicion mechanism assigns a continuous value as opposed to a binary measure to each client session, and the scheduler utilizes these values to determine if and when to schedule a session's requests.

## 2. DETAILED SYSTEM STUDY

### 2.1 EXISTING SYSTEM

The Internet was designed with functionality, not security, in mind, and it has been very successful in reaching its goal. It offers participants fast, simple and cheap communication mechanisms at the network level that provide "best effort" service to a variety of protocols.

The end-to-end paradigm enabled end users to manage their communication any way they desired, adding complexities such as encryption and authentication, while the intermediate network remained simple and efficient. Problems arise when one of the parties in the end-to-end model becomes malicious and acts to damage the other party. In that scenario, end-to-end protocols are violated and provide no more guarantees. At the same time, the end-to- end paradigm prevents the intermediate network from stepping in and policing the violator's traffic. Instead, it continues passively forwarding packets to their destination, where they overwhelm the victim's resources. End-to-end flow management was unable to ensure a fair allocation of resources in the presence of aggressive flows (i.e., those that would not deploy congestion control). This problem was recognized and finally handled by enlisting the help of intermediate routers to monitor and police bandwidth allocation among flows to ensure fairness. There are two major mechanisms deployed in today's routers for congestion avoidance purposes – active queue management and fair scheduling algorithms. A similar approach that engages intermediate routers in flow management may be needed to completely solve the 14 DDOS problem.

### 2.2 PROPOSED SYSTEM

The challenges to designing DDOS defense systems fall roughly into two categories: technical challenges and social challenges. Technical challenges encompass problems associated with the current Internet protocols and characteristics of the DDOS threat. Social challenges, on the other hand, largely pertain to the manner in which a successful technical solution will be introduced to Internet users, and accepted and widely deployed by these users.

The main problem that permeates both technical and social issues is the problem of large scale. DDOS is a distributed threat that requires a distributed solution. Attacking machines may be spread all over the Internet. Clearly, attack streams can only be controlled if there is a point of defense between the agents and the victims. One approach is to place one defense system close to the victim so that it monitors and controls all of the incoming traffic. This approach has many deficiencies, the main one being that the system must be able to efficiently handle and process huge traffic volumes. The other approach is to divide this workload by deploying distributed defenses. Defense systems must then be deployed in a widespread manner to ensure effective action for any combination of agent and victim machines. As widespread deployment cannot be guaranteed, the technical challenge lies in designing effective defences that can provide reasonable performance even if they are sparsely deployed. This paper also provides a comprehensive work of development of safety mechanism for Integration of PMS/Decision matrix tool. The rest of the paper is organized as follows: Section 1.1 briefly describes PMS Integrated combined with Analytical [9] decision Tool. Section 2 briefly describes System Design Scenarios; Section 3 discusses Resident features and Database for office in PMS & Protection mechanism system. Section 4 discusses about replication manager for PMS, Section 5 discusses about Communication between ship system and office systems. Section 6 PMS Salient features. Section 7 Client Server Architecture for PMS Integrated software. Section 8 Test Cases for the safety System. Section 9 Written Sample code. Section 10 listed conclusions.

### 2.3 PROPOSED SYSTEM

The PMS system will have module of inventory to handle detailed spare database, their indenting as per the requirements, and their receipt as per the dispatch from office and tracks the cost of spares on an equipment basis and on the job basis. Which is interfaced with Analytica decision tool. Job details and job schedule records are sent to office and the office system is kept updated using email with the help of Replication Manager. It is important to understand that why PMS is required. Figure 1. Illustrates PMS and decision matrix tool integration discussed for this research paper.

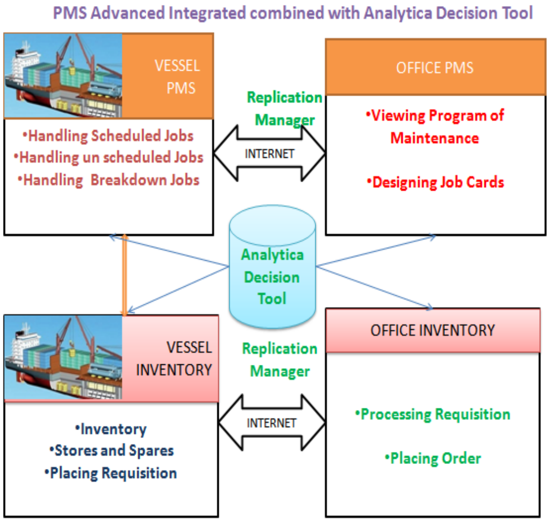


Figure 1. PMS and decision matrix tool integration

## 3. SYSTEM DESIGN SCENARIOS

A scenario is an instance of use case describing a concrete set of sections. Scenarios are used as examples for illustrating common cases. The following are the scenarios for all possible use cases in “IDENTIFYING AND SCHEDULING DDOS-APPLICATION LAYER ATTACKS”

**Scenario_1:**
Scenario Name: Send Request to the server
Participating Actor: Clients
Entry Condition: Send Request
Flow of Events:

1. Send Request
2. Waiting for Response

Exit Condition: Receiving Response

**Scenario 2:**
Scenario Name: Generate Response
Participating Actors: Main Server
Entry of Event : Receive Request Details from Proxy Server.
Flow of Events:

1. Receive Request Details From Proxy Server.
2. Generate Response to the Clients

Exit Event: Generate Response to the Clients

**Scenario 3:**
Scenario Name: Check Request details
Participating Actor: Proxy Server
Entry Condition: Receive the Details From the client
Flow of Events:

1. Receive the Details From the Client.
2. Check Whether attack or Not
3. If Attack the Calculate suspicion Value.
4. Store the request details into Queue.
5. Retrieve the details from queue using scheduling some algorithm
6. Send the Request Details to Main Server
7. Remove the Details from queue after generate Response

Exit Event: Remove The Details from Queue after generate Response.

## 4. RESIDENT FEATURES AND DATABASE IN OFFICE FOR PMS & PROTECTION MECHANISM

The system residing in the office will be a replica of the system existing in the vessels with the following extra features:

A spares inventory system having the following capabilities:

- Indenting from vessel.
- Indents receiving in office from vessel.
- Receipts generated by the vessel based on the system generated indents.
- Updating of the receipts from the vessel to the office.
- Auto update of material database, which contains the inventory status (Material quantities) from vessel to office on running the “Replication Manager”.

### 4.1 DATABASE DESIGN

### 4.1 (a) Data Dictionary

A Data Dictionary is a collection of metadata that is data about data. In addition to storing catalogue information about schema and constraints, the data dictionary stores other information, such as design decision, usage standards, application program descriptions, and user information.

Table 1: Attacks

| Name | Null? | Type |
|---|---|---|
| ATTACK_CODE | NOT NULL | NUMBER(2) |
| ATTACK_NAME | NOT NULL | VARCHAR2(20) |

Table 2: Status

| Name | Null? | Type |
|---|---|---|
| STATUS_CODE | STATUS_CODE | NUMBER(3) |
| STATUS_NAME | NOT NULL | VARCHAR2(20) |

Table 3: Session details

| Name | Null? | Type |
|---|---|---|
| CLIENT_IP | NOT NULL | VARCHAR2(16) |
| SESSION_ID | NOT NULL | VARCHAR2(50) |
| NO_OF_ REQUEST | NOT NULL | NUMBER(5) |
| STATUS_ CODE | NOT NULL | NUMBER(3) |
| SESSION_NO | NOT NULL | NUMBER(6) |
| SUSPICION_ VALUE | NOT NULL | NUMBER(18,16) |

Table 4: Request_details

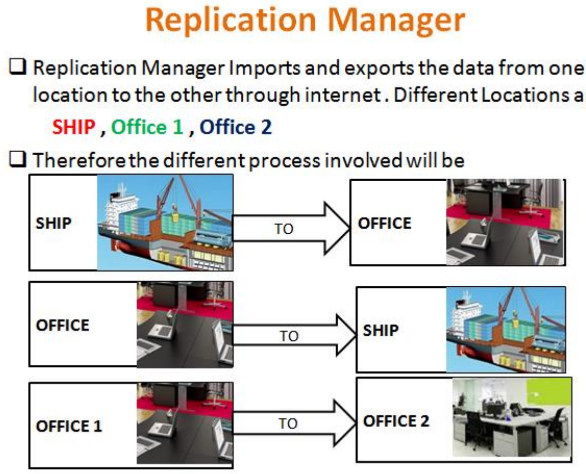
| Name | Null? | Type |
|---|---|---|
| CLIENT_ IP | NOT NULL | VARCHAR2(16) |
| SESSION_ ID | NOT NULL | VARCHAR2(50) |
| REQUEST_NO | NOT NULL | NUMBER(5) |
| REQUEST_ARRIVAL _DATE | NOT NULL | DATE |
| REQUEST_ARRIVAL _TIME | NOT NULL | VARCHAR2(9) |
| ATTACK_ CODE | NOT NULL | NUMBER(2) |
| REQUEST_ARRIVAL _TIME_IN_ MIS | NOT NULL | VARCHAR2(20) |
| REQUEST_STATUS_ CODE | NOT NULL | NUMBER(3) |
| REQUEST_RESOURSE | NOT NULL | VARCHAR2(150) |

Table 5: Queue_details

| Name | Null? | Type |
|---|---|---|
| CLIENT_IP | NOT NULL | VARCHAR2(16) |
| SESSION_ID | NOT NULL | VARCHAR2(50) |
| ATTACK_CODE | NOT NULL | NUMBER(3) |

## 4.2 REPLICATIONS MANAGER

Replication manager imports and exports the data from one location to other location through internet. The locations explained in Figure 2.



Replication Manager

❑ Replication Manager Imports and exports the data from one location to the other through internet. Different Locations are SHIP, Office 1, Office 2

❑ Therefore the different process involved will be

Figure 2. Replication manager data import/export

## 5. RESIDENT FEATURES AND DATABASE IN OFFICE FOR PMS & PROTECTION MECHANISM

Ship system will communicate with Office system through email. An online updating of both systems is done in this environment. Our system effectively uses internet mail for updating ships data to office and vice versa. The updating happen as mail attachments of the size of 10 kb. All the actions pertaining to database updating are done automatically by our application "Replication Manager" and no manual intervention is required for the same.

## 6. PMS AND DECISION TOOL INTEGRATION SALIENT FEATURES

1. Integrated with Decision matrix tool
2. Classification of all the Equipments and their components based on their operation.
3. Creation of "Job Cards" which contain the detailed instruction as to how a "Job Order" needs to be carried out.
4. Creation of a "Job Order" (work Order) based on a "Job Card" and scheduling it based on Time or Running Hours, and assigning it to a particular designation.
5. Generation of warnings for conditions such as:
   (a) A "Job Order" approaching its Due Date
   (b) "Job Orders" that are Due for the day
   (c) Job Orders which are Overdue
6. "Job Orders" that are suspended or Rescheduled due to conditions such as unavailability of spares or manpower or any other reason, are highlighted on the screen.
7. Booking of spares at the time of creation of a "Job Order", so that a warning can be generated in advance based on the number of spares that would be consumed by the "Job Orders" that are due within a certain period, and those spares can be immediately be indented, in case they are deficient.
8. Planning of the "Job Orders" is made extremely easy by a user-friendly interface.
9. Information exchange between the vessel and the Office made very simple, and with no hassles, with our add-on product "Replication Manager", keeping the vessel as well as the Office well informed of each other's activities. Data Transfer files through e-mail are as small as 2 – 3 KB.

Replication Manager enables updating date between office and ship at preset time intervals. PMS Inventory integrated flow charts explained in Figure 3.

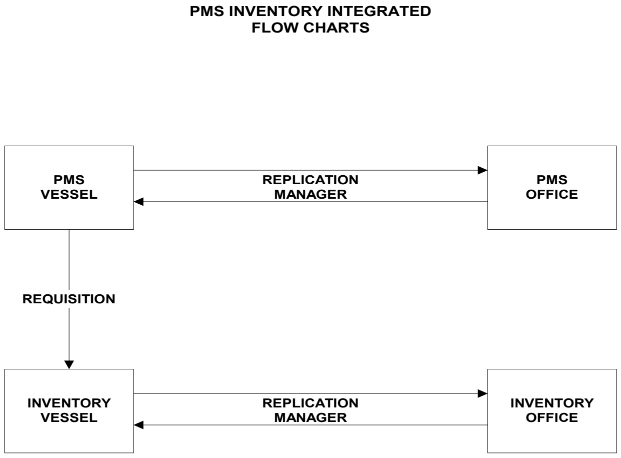


Figure 3. PMS Inventory Integrated flow charts

Flow of PMS that contains defining of equipment tree, defining job cards, scheduling job orders, Job completion explained below.

1. Maintaining a Material Stock database up-to-date by recording and computing transactions such as Receipt or Issue of any material.
2. Maintaining a Suppliers and Manufacturers List for all the Materials to facilitate quick Enquiries to the respective suppliers.
3. Warnings generated when any material falls below a preset "Minimum Level" by means of

color-coding which even indicates if a material has been indented.

4. Generation of "Periodic Indents" by the Office for the supply of the materials periodically to all the vessels.
5. Approval of the Indent by the authorized personnel (password protected) before being sent to the office
6. Sanctioning of the materials in an Indent by the Office personnel (password protected)
7. Receipts can be generated against the materials received (in Office or on the vessel).
8. The office can be kept updated of all the transactions inside the vessel via program-generated files that are sent as e-mail attachments to the specified Location.
9. User based Authentication for Approval, Sanction and Issue
10. User based Authentication for Approval, Sanction and Issue
11. User Defined Report generation which gives the flexibility to user to generate Report Templates as per there needs for. E.g. summarizing the total price for particular period, selecting fields the necessary fields to be displayed on the report etc.
12. Inventory Cost Evaluation
    a. Budget
    b. Material Specific Cost Value
13. In Order, Material specific additional Charges evaluation
14. Issuing process without undergoing Requisition

PMS over all Flowcharts explained in Figure 4.

The Inventory Flowcharts shown in Figure 5. shows working of a basic module of system with an Inventory database residing in ship and vessel.

## 7. CLIENT SERVER ARCHITECTURE FOR PMS INTEGRATED SOFTWARE

The Client Server Architecture and Configuration specified for Server PC and Client PC are same for Office Environment and Vessel Environment. The Replication of Data is done between the Server of Office and Vessel through Replication Manager. Representation of Client/Server Architecture shown in Figure 6.

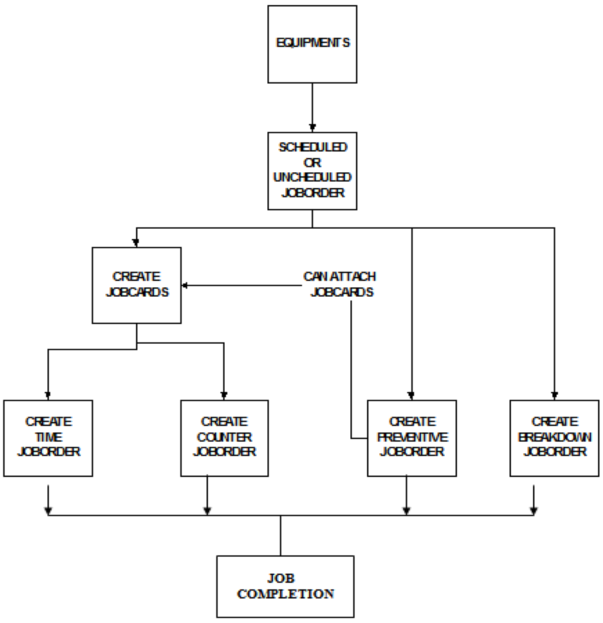


Figure 4. PMS Advanced overall Flowchart

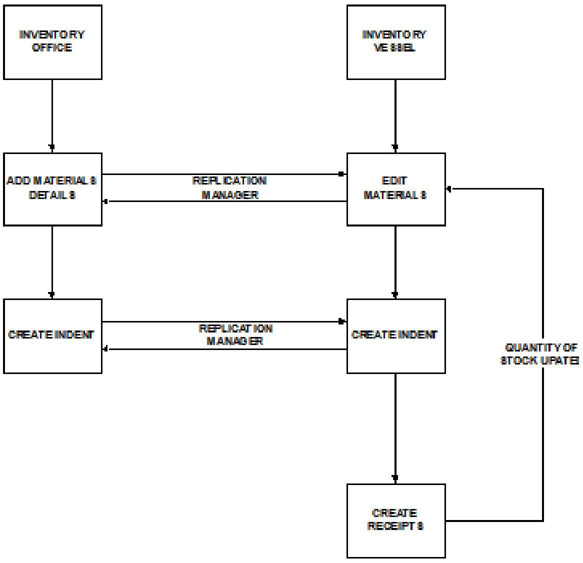


Figure 5. Inventory Flowcharts (shows working of a basic module of system with an Inventory database residing in ship and vessel

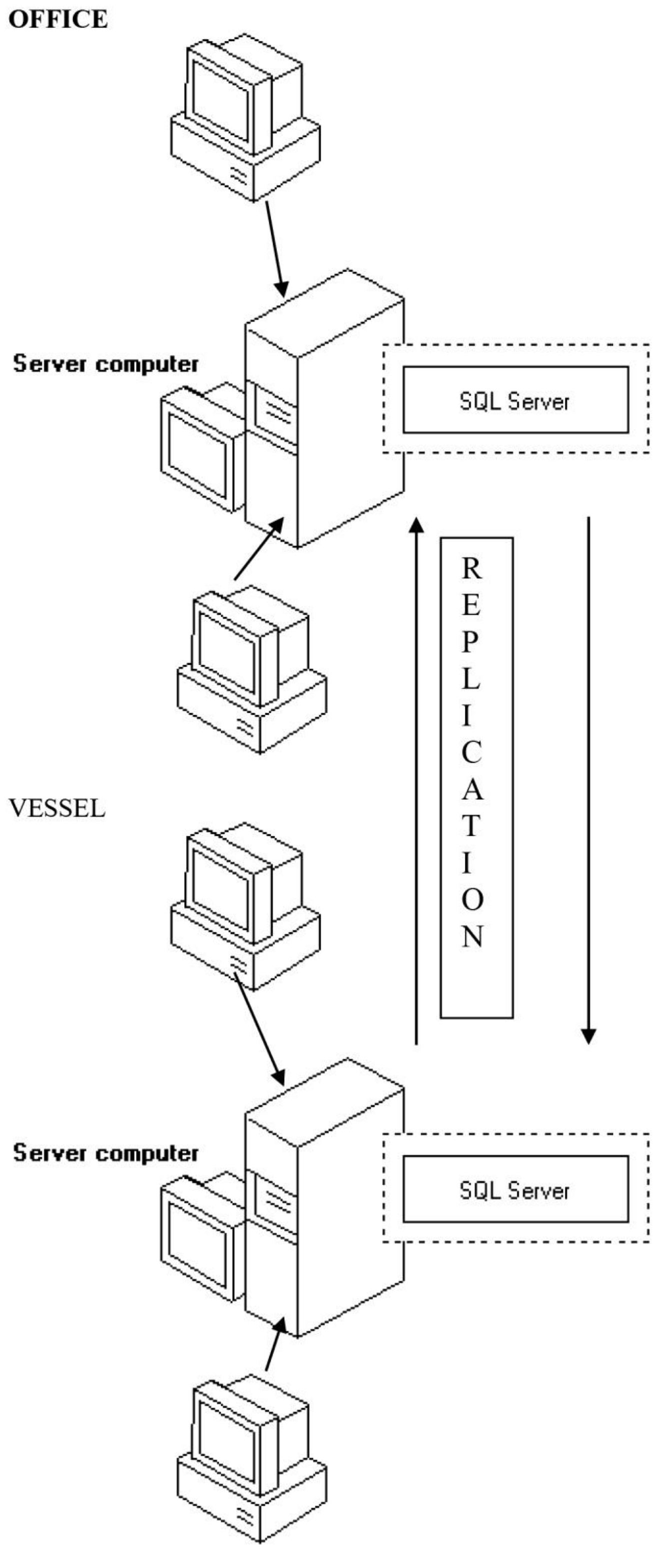


Figure 6. Representation of Client/Server Architecture

## 8. TEST CASES FOR THE SAFETY MECHANISM

Test Cases for the System discussed in this section.

For Unit testing Module is tested separately by the coder himself simultaneously with the coding of module. Unit testing focuses verification effort on the smallest unit of the software, design the module. Unit testing is always white-box oriented, and the step can be conducted in parallel for multiple modules.

For Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors, associated with interfacing.

For Validation testing demonstrates the traces the requirements of the software. This can be achieved through a series of black box tests.

For System Testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. The various tests include recovery testing, stress testing, and perform testing.

| **Aspects to be tested** | **Excepted results** |
|---|---|
| Home page of PMS | The home page should be displayed on port no which server run |
| Home page of Proxy Server | The home page should be displayed on port no which Proxy Server run |
| Login Error page | This page will displayed when Admin enter invalid username or password |
| Introduction page | This page will displayed when user click on the Home Menu. And it contain the Introduction of the all the DDOS-attacks. |
| Attacks page | This page will be displayed when user click on Attacks Menu. And it contains details of the attacks. |
| Details of attacks page | This page will be displayed when user click on the session-id. And it contains details of the each request under the specified session. |
| Solved Attacks | This page will be displayed when user click on the solved attacks Menu. And it will contain list of the all solved attacks. |
| Unsolved Attacks | This page will be displayed when user click on the unsolved attacks Menu. And it will Contain list of the all unsolved Attacks. |
| Percentage of Attacks | This page will be displayed when user click on the percentage Menu. And it contain percentage of the both solved and unsolved attacks for all the requests. |

| **Result** | **Action taken** |
|---|---|
| Home page not found | The Apache server should be started. |
| For a flooding attack the additional requests are not placed in the queue. | Increase the time variant in the request. |
| Attacks information is not displayed | Select one of the menu item form attacks menu. |
| Solved attacks information is not displayed | Select one of the menu item from solved attacks menu |
| Unsolved attacks information is not displayed | Select one of the menu item from unsolved attacks menu. |
| Percentage of attacks information is not displayed | Select one of the menu item from percentage menu |
| Detailed request details are not displayed | Select one of the session-id from the list. |
| Log file length is too big. | Log file length is too big. |

## 9. SAMPLE CODE

Sample Code Presented as a Separate Appendix. Refer APPENDIX- A

## 10. CONCLUSION

In this Project, I explored the vulnerability of systems to sophisticated application layer-7 DDOS-attacks. These attacks mimic legitimate clients and overwhelm the system resources, thereby substantially delaying or denying service to the legitimate clients. Since these resource attacks are un detectable via sub-layer-7 techniques, I developed DDOS-Shield, a counter-mechanism that uses a DDOS-Resilient scheduler to decide whether and when the session is serviced. Using a web application hosted on an experimental test bed, we demonstrated the potency of these attacks as well as the efficiency of DDOS-Shield in mitigating their performance impact.

This solution provides following benefits:
1. Attack free onboard system of PMS
2. No threat to Vessel’s (Ship or any floating body) Specific Software environment
3. Runs on any PC, without any additional cost on high end Database Softwares
4. No cost of hardware additions
5. A single person can handle the data processing in a single terminal
6. Considers the fleet (number of ships) in totality, leading to proper data sharing and resource allocation
7. Complete backing up throughout the implementation
8. Highly flexible packaging and operational procedures

## 11. ACKNOWLEDGEMENTS

This work was supported by CDM & support of Computer Science and Systems Engineering, College of Engineering (A), Andhra University, Visakhapatnam.

## 12. REFERENCES

1. Marine & Offshore classification Rules & Guide Lines by International Association of classification Societies.
2. Petroleum Act 1998.
3. Energy Act 2008.
4. Department of Energy and Climate Change, Guidance Notes - Decommissioning of Offshore Oil and Gas Installations and Pipelines under the Petroleum Act 1998, URN 09D/734, Version 4, August 2009
5. Oil & Gas UK, Guidelines for the Suspension and Abandonment of Wells, Code WEL03, 2009.
6. Offshore Installations (Safety Case) Regulations 2005, SI 2005 No 3117.
7. Oil & Gas UK, Fisheries Sensitivity Maps in British Waters, 1998.
8. Department of Trade and Industry, Guidance Notes on the Offshore Petroleum Production and Pipelines (Assessment of Environmental Effects) Regulations 1999, Version 15, 14 November 2003.
9. Analytica is a visual software environment for building, exploring, and sharing quantitative decision models. This was developed by LUMINA Decision Systems.
10. JACOBSON, I., BOOCH, G., RUMBAUGH J., & RUMBAUGH, J., *The Unified Modeling Language User Guide* (2nd Edition, 2005), The Addison Wesley Object Technology Series;
11. PRESSMAN, R.S., *Object-Oriented Software Engineering* (7th Edition, 2010) - McGraw-Hill;
12. KEOGH, J., *The Complete Reference-J2EE* (Indian Edition, 2015) – Tata McGraw Hill.
13. CRAWFORD, W., and HUNTER, J., *Java Servlet Programming* (2nd Edition, 2010), O’Reilly Media
14. LONEY, K., *Oracle Database 10g: The Complete Reference* (1st Edition, 2004) - McGraw-Hill;
15. *Java Server Programming (J2EE 1.4) Black Book* 2007 (Platinum Edition), Dreamtech Press
16. JOHNSON, R.A., MILLER, E., & FREUND, J.E., *Probability & Statistics for Engineers*, (9th Edition, 2016) Prentice – Pearson