SECURITY AND VULNERABILITY IN GRID MANAGEMENT

FINAL YEAR PROJECT [CM3203]

CARDIFF UNIVERSITY

SCHOOL OF COMPUTER SCIENCE

STUDENT NUMBER: C1740337

AUTHOR: JUDITH MAY

SUPERVISOR: PHILIPP REINECKE

MODERATOR: SURYA THOTTAM VALAPPIL

Abstract

The project's objective is to explore the security and vulnerability of the Wide Area Management Systems (WAMS), especially aspects of data transfer, storage, and the impact of moving from proprietary software to open-source software. Previous studies on the topic have focused mostly on improving some aspects of the WAMS, such as delay integrity checks, Phasor Measurement Unit (PMU) communication, and maintaining the system's reliability using security measures. This research focuses on the security of the software's used in the Control room, from acquisition to storage and analysis of data.

The Open Web Application Security Project (OWASP) testing methodology was used; it is a balanced approach using penetration testing, security and vulnerability scanning etc. This was a BlackBox test approach using the OWASP ZAP, Google, Firefox and Wireshark tool. The OWASP web application security testing was used as a guide throughout the process. The result of the forty-one test cases done from nine categories are; 28 (72.5%) passed the test, 8(17.5%) failed the test, and 4(10%) had issues that did not fail the test but showed vulnerabilities that an attacker could exploit.

The number above does not actually point to a fully protected solution for a software used in an important sector. It shows that extra security measures have to be in place to use this software. This research shows that these systems have been exploited over the years and could cause serious disruption when attacked, such as blackout, rendering usable data useless and service disruption.

Acknowledgement

I am overwhelmed in all humbles and gratefulness to acknowledge my depth to those who have contributed immensely to put these ideas well above the simplicity and into something concrete.

I want to express my special thanks of gratitude to my indefatigable supervisor Philipp Reinecke for his help and support throughout the project. Thanks to the head of school, Professor Stuart Allen, who allowed me to do this topic. I am also deeply thankful to my friend and housemate Louis Marie Noe Ndoki for his support and his vital part in supporting me during difficult time and health.

Finally, I would like to express my gratitude to family and friends for their support and motivation, especially Mother Maria Emeagi for her support and encouragement throughout this project. I am thankful to them for their immense contribution.

Table of Contents

Figures

Tables

1. Introduction

Securing the electric power system is among the highest priorities for important infrastructures protection in every country worldwide. Attacks on these infrastructures are considered an act of war(terrorism) in most cases; attacks in a grid system are considered a cyberwar. Only specific individuals will have the capacity or interest to carry out such attacks; reasons could be political or religious.

While there is plenty of research related to smart grid system, very little is about security in wide-area management. Different systems are used in grid management, such as Energy Management Systems, Distribution Management Solutions, Supervisory Control and Data Acquisition (SCADA) systems, Grid Analytics, Wide Area Management Systems, etc. These systems are put in place for adequate management of the grid system. For this project's purpose, It will be focusing on Wide Area Management Systems (WAMS). This project explores the security and vulnerability in the data transfer, storage, and the impact of moving from a traditional (also known as proprietary) to an open-source software. This research is also aimed at looking into the security and vulnerability of the WAMS software solution and propose a possible solution.

1.1 Research aims & objectives.

This project explores the security and vulnerability in the data transfer, storage, and the impact of moving from a traditional software UI to an open-source software interface. This research is also aimed at looking into the security and vulnerability of the WAMS software solution and propose a possible solution. On the technical side, I will be carrying out an analysis of the software.

- Security in the grid management system (WAMS)
    - Introduction to WAMS
    - WAMS components
    - Data storage (physical and cloud servers)
    - WAMS Software & User interface
- Analyse the vulnerability of the software using a penetration testing tool.

1.2 Significance of the research

This research is significant because it tackles a crucial problem of security and vulnerability within an organisation and therefore a potential benefit to companies, software developers,

security consultants, and students who use the software(open-source). Previous research has mostly focused on improving aspects of the software or looking at security in the smart grid in general but not the software explicitly used. Therefore, this research explored the identified research gap. Additionally, it looks more into the security of the software used in the WAMS systems (stream data, analysis and storage), specifically the Phasor Data Concentrator and Historian. The security test using a balance approach can create insight to help security consultants, information security officer and operations managers decide on the security policies based on the software and components used.

1.3 Section breakdown

This document has several sections: backgrounds, security of the WAMS system, vulnerability scan, evaluation, future works, reflections, and conclusion.

The background sections cover the introduction to WAMS, a brief history of the WAMPAC system, including its components and functionalities. It has a breakdown of how the grid is connected and the supporting components of WAMS.

The security section covers a brief discussion on different types of attacks that happens in a grid system that affects WAMS, including examples of such incidents recorded in the past on the various security breaches. Furthermore, it covers known security measures that could be put in place to prevent these attacks.

The vulnerability section covers the various software used to analyse the openHistorian and why that software was chosen. It also covers the test cases that were carried out and the configuration of the system. The system configurations cover the different parts of the system ports and aspects that were changed to make the system work. The test cases are also broken down into various areas indicating the tools used for each issue addressed. The penetration test result with a detailed explanation of the problems identified is also included in this section.

The evaluation covers the summary and analysis of the result from the penetration test. Showing the calculated percentage of the different pass, failed and issues rating, with the majority of the test being passed, it shows the system's security level. Also, Included in the evaluation are the solutions to the raised problems in each OWASP web application test sub-categories. Most importantly, this section gives the overall evaluation of the assessment of the WAMS system based on the research and result of the test.

The conclusion section wraps up the summary of the entire research. It includes the findings from the research and solutions to the problem as listed in the research proposal, emphasising the system's complexity.

The future work section explained the continuation of the work that would have been covered in this research due to the short time frame not being able to achieve them, such as getting commercial software, analysing the hardware, and making a comparison between different vendors. Also, detailing the software that would be used have been used if there was more time left to finish off the project.

The reflection section covers my experience during this research. It entails what went well during the research and things I was able to achieve, also details of things that didn't go well in the research, such as the setback in the response from companies and how I got around the issues. Furthermore, it covers what was learned so far and skills developed during the three months while doing the research.

2. Related work

Luigi Coppolino et al. 2014 researched into Exposing vulnerabilities in electrical power grids using an experimental approach. The research was about finding the system vulnerabilities and exposure to attacks and unintentional errors due to the complexity of the grid, weakness caused by disruption of the communication network and introduction of malicious software or hardware, and vulnerabilities introduced by the use of new technologies. The result of the experiment shows the vulnerabilities in the components and how they can be exploited if adequate security measures are not implemented. In particular, security threat can be caused by a lack of encryption in communication, a lack of input validation, and weak password policies. The vulnerabilities can enable system attacks such as SQL injections to be launched against the electric power grid (Luigi Coppolino, 2014).

Premkumar S. et al. 2017 researched the impact of denial-of-service (DoS) attack in smart distribution grid communication networks. They simulated the WAMS communication system using GNS3(Graphical Network Simulator 3), PMU connection tester, Wireshark and LOIC (Low Orbit Ion Cannon) tools. According to the author, it is the first time in literature that the modelling and simulation of DoS detection in a smart grid communication network were done. The result of the simulation shows the vulnerabilities of the DoS attack in the power system monitoring. The proposed future work into preventing and mitigating this attack using intrusion detection and prevention systems (Premkumar S., 2017).

Aditya Ashok et al. 2013 researched the cyber-physical security of wide-area monitoring, protection, and control (WAMPAC) in a smart grid environment. They emphasised the importance of securing the WAMPAC to maintain the power grid security reliability. As a solution, they proposed using a game-theoretic framework model for cyber-physical security for WAMPAC applications. The game theory framework models a cyberattack (strategic interaction between the attacker and defender), which cannot be modelled using a traditional risk assessment approach.

Fadi Aloul et al. 2012 research the threats, vulnerability, and solution for smart grid security. They identified that the size of the smart grid and the increasing communication capabilities make it more prone to cyber-attacks. As critical infrastructure, it is important to identify the vulnerabilities and sufficiently provide a solution to the issues to reduce risks to an acceptable level.

3. Introduction to WAMS

This section of the projects introduced the WAMS aspect of the grid. It explains what the WAMS stands for and a brief history of the systems, how it works, and how it is linked to other aspects of the grid. It also has the explanations and system functions of the components of the WAMS system and introduces the PDC concepts that will be discussed later in the third chapter.

3.1 Power Grid

Power grids are complex infrastructures that have been implemented during the last few decades; they are much more than just transformers, high voltage transmission lines, power plants, and distribution lines that connect individual customers. The general architecture of the power grid consists of seven distinct components that interact at various levels, namely:

- Digital (information and communication systems),
- Control (includes protection circuits, management, and synchronisation systems),
- Electric (e.g., transformers, power plans, protections circuit, transmission, and distribution lines),
- Convergent networks,
- Coordination,
- Industry (such as utilities, operations, planning, and marketing)
- Regulatory.

These components will determine the power grid's ability to change in response to technical, operational, cybersecurity, market, regulatory, or end-user requirements (Monteagudo, 2020).

The overall architecture of the power grid is highly complex, and numerous moving parts interact with one another and have been constructed over the last few decades. In addition to these complexities, this infrastructure was not built with cybersecurity in mind; at that time engineers were not required to address the issue raised by the new approach "connectivity at all times". With the increase in connectivity comes the problem of cybersecurity. The era of smart grid systems has increased the awareness of the threats and the security requirements for these security challenges.

There are four layers of a smart grid system:

- Physical Layer: includes generation, distribution, transmission, storage.

- Communication Layer: home area, access & blackholes, core, office, and external networks.
- System Integration Platform: computing infrastructure, networks, and security management, application, and data integration.
- Software Layer: meter data analysis, outage management, load control, GIS, wide area management systems.

3.2 What is WAMS?

Wide Area Monitoring Systems (WAMS) represent the future of power system monitoring; they are operated around the world by utilising time-synchronised, high-resolution measurements of the electricity system. WAMS enables real-time monitoring of power systems dynamics by uniting new estimation, computing, and networking improvements. The estimations of voltage and current phasors are logged by phasor measurement units (PMUs) installed across a wide area power framework and time labelled at the measurement point using a normal time reference (for example utilising GPS).

WAMS provides readings and insights which can reveal information about stability, system security, and efficiency, enabling Operators, Analysts, and Planners to respond rapidly, accurately, and appropriately to system conditions. The system combines synchronised angle measurements to common time reference and stores them as a snapshot of the system. The snapshots are refreshed at a pace of up to once per cycle. With appropriate computing and networking resources available, the sequences of the snapshots can be used to visualise system dynamics in real-time. However, with recent advancements in computing resources in the power system sector, the synchronised snapshot can be used for more than just the visualisation of the dynamic systems (Peter Wall, 2016).

The concept of WAMS was first introduced in the 1980s, and the first commercial product and software device got to be accessible within the 1990s. Since then, numerous studies and pilot projects were created in a few utilities. In a few of them, the wide-area systems were completely implemented with the result and return of ventures. However, numerous markets and clients were not ready to complete technology, both from a financial and organisational aspect, although most have understood the potential benefits of WAMS (Luis Fabiano dos Santos, n.d.). The WAMS technology has been improved over the years to include control and protection (WAMPAC).

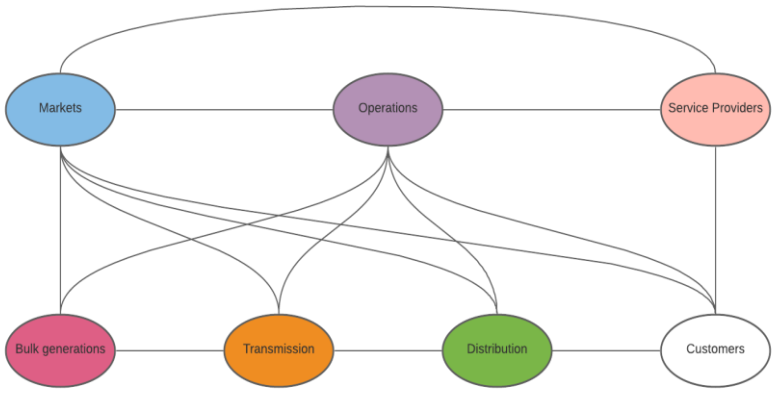


Figure 1: Conceptual diagram of a grid

From the diagram above, WAMS comes under the transmission and operations sector of the grid.

3.3 Components of WAMS

3.3.1 Synchrophasor

Synchrophasor is a time-synchronised figure that represents the magnitude and phase angle of the sine wave found in electricity. The synchrophasor technology is used in real-time operations and offline analysis to improve the efficiency and reliability in the grid and lower grid operating costs. They are measured using high-speed monitors called Phasor Measurement Unit (PMU). The synchrophasor protocol supports TCP and UDP over IP. Both protocols can be used to send data, UDP is the only protocol that can support multicast (one to many nodes), and commands are typically sent over TCP. The data are streamed from all the associated PMU at a reporting rate in a range of 10 to 60 reports per second and accumulated, then time-synchronised and saved by the PDC and served with the least latency to other client application for system alertness analysis, control, monitoring, and protection action (Luis Fabiano dos Santos, n.d.).

3.3.2 Phasor Measurement Units (PMU)

PMU is a device that measures voltage, current, and frequency in terms of magnitude and phasor angle at high speed. This advanced out of the early work on PC-based transferring performed at the American Electric Power (AEP) Corporation. The principal creator joined AEP in 1969 and was before long given the task of building up a computerised PC-based distance hand-off to ensure overhead transmission lines. It was normal that the new transfer will have execution similar to that of the best accessible, simple transfers (A. Phadke, 2018).

Modern synchronised phasor measurement technology dates back to 1983 from an article by Phadke et al., in which the article identified the importance of positive sequence voltage and current phasor measurements and how it has the potential to simplify and improve SE algorithms as well as other real-time analysis programs (A. Phadke, 1983). GPS provides the most efficient method to measurement synchronised phasor in power systems over great distances. In the early 1980s, Virginia Polytechnic Institute and State University in the USA led the effort to build the prototypes of the modern PMU based on GPS. IEEE finished a standard in 1995 and released a revised version in 2005 to standardise the data format used by PMUs (Jian Ma, 2010).

There are two types of PMUs, standalone PMU and Integrated PMU. The standalone PMU performs dedicated high accuracy, time-stamped, precision synchronised measurement task in a standalone device. The integrated PMU is an IED that integrates synchronised precision measurement tasks, such as fault recorders, digital meter, and delay. The focus of this research is on the software, and the hardware is beyond the scope of this research.

Functionalities of a PMU:

- To improve the accuracy of modelling system conditions
- To predict and detect stress and instability on the grid.
- To provide information for event analysis after a disturbance has occurred.
- To identify inefficiencies
- To predict and manage line congestion.

Whereas PMU's give progressively exact situational mindfulness capabilities, their full potential will not be realised unless this estimation information can be shared among other utilities and controllers. Furthermore, control framework applications have to be re-examined to decide the degree to which these improvements can make strides in the grid's effectiveness and unwavering quality. The advancement of progressed control applications will depend on WAMS that can successfully disseminate data securely and solidly.

3.3.3 Phasor Data Concentrator (PDC)

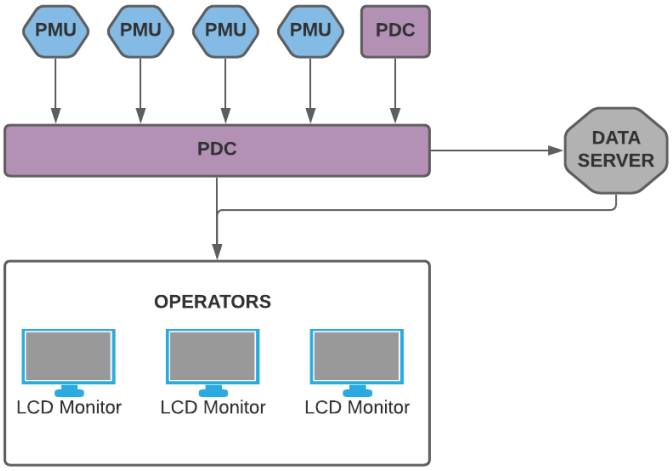


Figure 2: Conceptual diagram for processing data for WAMS

PDC are devices that collect and time synchronises phasor data streamed from PMU from across the grid. A PDC is a critical link between the PMUs that collect the phasor data and the synchrophasor applications which use the data (Paolo Castello, 2018). Aside from streaming data from PMU, PDCs can stream data from other PDCs. The collected data from PMUs are sorted and linked according to the time-stamp value, and this enables comparing real-time monitoring of systems with high precision sampling. The data collected are stored in a large database system (servers or cloud storage) for accurate post-applications analysis, such as loss of mains, blackout, and fault-event monitoring (O.Mohammed, 2017). The time referencing in PDC is important because inaccurate timestamps can cause misdiagnosing and reduce control of the network.

The official IEEE standards for PDCs C37.247 was released in approved in 2019. The standard specifies the requirements that a PDC for power system should have; this includes an aggregation of data, processing of synchrophasors and other data, data interface with other systems, handing of commands (configuration and other metadata), testing and performance (including latency, environment, and throughput). Although the importance of cybersecurity was acknowledged in the overall system and discussed, it was stated that it was beyond the scope of the standard based on the specific functions of the PDC (IEEE, 2019). Before the official standard PDC hardware and software, manufacturers adhere to the IEEE C37.118 1/2-2011 (standard for synchrophasors), C37.242-2013 (IEEE Guide for Synchronization, Calibration, Testing, and Installation of Phasor Measurement Units (PMUs) for Power System Protection and Control) and C37.244-2013 (IEEE Guide for Phasor Data Concentrator Requirements for Power System Protection, Control, and Monitoring). Because there were no

industry standards that existed for the PDC at that time, the performance and functionality of the PDC differ by vendors.

Some functions of a PDC:

- Aggregation of data
- Configuration
- Conversion of data format and coordinates
- Cybersecurity
- Data communication
- Data forwarding
- Data latency calculation
- Data retransmission request
- Data validation
- Handing of redundant and duplicate data
- Performance monitoring
- Phase and magnitude adjustment
- Support and conversion for data transfer protocols

3.3.4 Communication network

The communication network is an important infrastructure of the power system because it ensures a stable and safe operation of the power system. Having high performance, secure and reliable network communication system is the key to support grids. Communication network together with security and stability control of the systems, electrical communication network and dispatching automation systems are considered the pillars of a stable operation in an electrical power system. The communication network of WAMS is an important aspect of the grid system, and it sets a high requirement to its communication network than any other part of the grid system. In other for the security on the grid to be defended efficiently, the real-time operation parameters of a wide-area network with high accuracy and synchronisation are collected, and the distributed data is incorporated with the high-speed communication network, then global dynamics information of the power grid is obtained under space-time coordinates (Cheng Xiaorong, 2012). It is designed based on an open system interconnection (OSI) layer model. The OSI model consists of seven layers: application, presentation, session, transport, network, datalink, and Physical Layer.

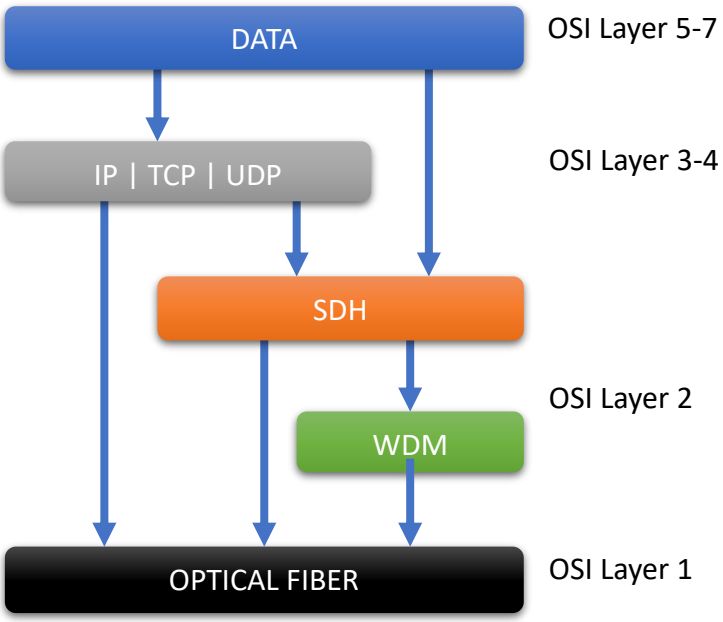


Figure 3:OSI layers of data connection

3.3.4.1 Physical Layer

The preferred means of communication is Optical fibre because it is used as the physical media of the WAMS communication network in other to meet up the requirement. Optical fibre makes use of light pulses as an alternative to electrical pulses to transmit information. For this reason, it can provide thousands of instances of higher bandwidth than ordinary electrical systems. Fibre optic cable can be covered through sheathing and armour to make it resistant to harsh environmental conditions (John, 2012).

0ptical fibre has the traits of frequency bandwidth, low attenuation, small line move section, no radiation, anti-electromagnetic interference capability and high security. It has an increasing number of come to be a desired physical media of the network, particularly the backbone community constructed. Optical Fibre composite overhead Ground Wire (OPGW) is a type of cable which is specially designed and manufactured, considering the one-of-a-kind nature of the strength system. It opened up a transport channel with high-speed, wideband, and low power transmission on the groundwork of preserving the functionality and overall performance warning signs of the ground. It also has a dual characteristic of the transmission of energy and excessive-pace transmission of information.

3.3.4.2 Datalink Layer

Synchronous digital hierarchy (SDH)/ SONET are standardised protocols that transfer multiple digital bit streams synchronously over optical fibre using lasers or tremendously coherent light from LEDs. At low transmission quotes, data can also be transferred by using an electrical

interface. Wavelength division multiplexing (WDM) is a fibre-optic transmission technique that supports the use of multiple light wavelengths to send data over the same medium. Two or more lights can travel on one fibre, and several signals can be transmitted in an optical waveguide at differing wavelengths or frequencies on the optical scale. In WAMS communication network generally uses IP over SDH or IP over WDM technology on the data link layer.

Using IP over SDH directly is ideal for transmission in WAMS communication as it transmits data only. It is the transmission of IP over SDH that IP packet will be used to mapped to the SDH structure through PPP and then followed the SDH transmission rate of each module. The combination of fibre, SDH, and IP is straightforward to execute on the physical separation between different IP applications, which is essential to business such as control and protection of power system. Also, Using IP over WDM is a system that is easy to implement by reducing duplications, network equipment, network complexity, and complexity of network configuration while reducing cost and increasing efficiency in transmission.

3.3.4.3 Network layer

The IP is the protocol that is used in the network layer of the WAMS communication network. It refers to the set of guidelines that regulate how data packets are transmitted over a network. The improvement in the IP technology enables the efficient use of bandwidth by avoiding the allocation of capacity where it is not critical. It used widely generic standards based totally on proven technologies and an excessive degree of interoperability, a very high degree of autonomy to evolve network performance following the strategic wants of the utility. Reliability, due to the impact in IP networks, packets are instantly re-routed if a node or hyperlink fails and scalability to cope with growth (ABB, n.d.).

3.3.4.4 Transport layer

TCP and UDP protocols based on the IP protocol can both be used to transmit real-time data through the network. The TCP protocol is a connection-based communication protocol that ensures data delivery, and it forfeited the speed to a certain degree. When a mistake in transmission happens, TCP will retransmit the information, resulting in delay and cannot meet real-time conditions. UDP protocol does not have to consider the delay due to the retransmission and validation of data; using UDP in WAMS private network with less interference do not result in data loss. When data is lost within the allowed time frame, it can be recovered with the software in the measurement centre. It is the reason why UDP has an advantage over TCP in the WAMS communication network (Cheng Xiaorong, 2012).

	LAYERS	OSI -STACK
5-7	Application	Telnet, FTP etc,
4	Transport	TCP/UDP
3	Network	IP
2	Data link	Ethernet
1	Physical	Fibres, waves

Table 1: OSI Layer 1-4 as a transport vehicle for higher protocols

3.3.5 Data storage

Data from the PMU and PDC are stored in a server called Historian. The Historian server is the central point for managing all the collector and client interfaces, storing and optionally compressing, and retrieving data. The important data storage parameters are storage capacity, access response time, and data nodes such as phasors, digital statues, and analogue value. Each data file represents a specific time period of historical data. You can isolate even more tags and archives into data stores. The main purpose of data storage is to separate the labels by data collection interval. For example, you can place nameplates or static labels on data that rarely changes in value and process labels in another data store; this can improve query performance.

There are two types of storage servers: a physical server or a virtual server. Physical servers refer to the onsite servers that are in the control centres. Operating on electricity and metal, and modern physical servers are often capable of serving multiple users simultaneously. The energy generation company usually own their network of physical servers, and its typically housed in the data centres. There is no limit in terms of storage as they can be expanded when needed. Virtual servers are a partitioned part of a physical server because users rent virtual servers for a fraction of the cost of a physical server that is sold by hosting companies. The security system with the servers is that they are encrypted cloud storage, and data cannot be retrieved without authorisation or having the decryption key.

3.3.6 Application software

The WAMS technology has improved over the years to include Wide Area Protection and Wide Area Control System, which is generally referred to as WAMPAC. The WAMS software has the basic functions that it performs as well as advanced functions that are used in the analysis of the grid performance.

The software is divided into different categories for monitoring (PSGuard from ABB and PhasorPoint from GE), storage (e.g., Historian), post-event analysis (e.g., PhasorAnalytics from GE), grid stability, control systems, and forecast. The forecasting software is a recent innovation with the use of machine learning; the data can be used to predict or forecast the generation of energy in a grid system. An example of the forecast software is the Effective Inertia software from General Electric Digital. Effective inertia measures the combined inertia-like effects for rotating machines, active generator control, and passive load responses. The Effective inertia metering is non-intrusive, with no injection of forced stimulation into the network. WAMS data and analytics measure effective inertia in each regional area of the power system in real-time and combine them to a global value (General Electric , 2019).

According to the publication, the Inertia metering and forecast is agnostic of EMS and PMUs. However, integration with the Advanced EMS generation, network, and WAMS advanced application helps maximise the overall solution value. It is also used to solve the problem of blackout in the transmission and distribution industry. In August 2019, the London blackout occurred, and NGESO said they were working on innovation with General Electric to solve blackout or fault occurrences (GE, 2020).

3.4 Conclusion

Each component of the WAMS has its functions to keep the system running from acquiring the data using the PMU and PDC, storing data (Historian), visualising (vision), and analysing the data to perform maintenance tasks and use of the system. Each component is essential and dependent on each other to operate appropriately and uses different communication systems to communicate. Vendors of this separate component must follow IEEE standards to produce durable and suitable software/hardware for the energy sector.

4. Traditional vs open-source software

Companies use two types of software in this environment, depending on the functionalities or specification needed this allow companies to choose what software they think fits their requirements. The two types of software that could be used are closed source (Traditional/Proprietary) and open-source software.

4.1 Traditional software interface

Traditional or proprietary software is software that can only be bought or gotten through a software license or leased from its developer or publisher, whether it is an individual, company or organisation. The company that owns this software preserves intellectual property (IP) rights and may impose constraints on its utilisation and distribution. The source code is usually treated as a trade secret and cannot be altered by end-users because the source will reveal how the product works, and this prevents their competitors from stealing ideas or getting insight from it. The restriction of the source code is one of the main difference from open-source software. The constraints imposed by vendors are listed in the product's end-user license or terms agreement (IBM, 2020). Examples of this software in WAMS are PhasorPoint, PhasorHistorian, SIGUARD, PSGuard etc.

The advantages of proprietary software are that they provide tailored support to their customers. As a paid software, customers are offered extensive, accessible and effective after-sale support to customers. Also, proprietary software has a clear roadmap on the future of their product or software as they have to maintain and keep the customers informed on this progress continuously. In addition, the interface is mostly intuitive because they focus more on meeting the customer's expectation, and feedback is used to improve the usability of the software (Thompson, 2021). However, the cost of the traditional software is very high in the case of the WAMS system, and the individual API have to be license separately, which makes it expensive. Also, the maintenance of the system is not always free, and companies have to add that as an annual cost of using the system to their annual company budget. Users do not easily identify the security loopholes, even when identified and reported, this cannot be sorted straight away and usually takes longer to fix.

4.2 Open-source software interface

Open-source software is the opposite of traditional software, and it is free software that allows the user to modify, utilise and distribute by its end users. Unlike traditional software, open-source allows anyone to make changes to its source, and developers can share their

ideas, thoughts and code to build more inventive software solutions both individually and collectively as a community. Open-source software works with the fundamental standards of peer generation and mass collaboration, making more maintainable software advancement for the users. This software is not dependent on the company that develops them. If the company goes out of business, the source code will exist and improve by the open-source community (IBM, 2020). Example of open-source software in WAMS are OpenPDC, openHistorian, OpenXDA, SIEGate etc.

The benefits of using open-source software are that they are free to modify and easily distributed, as mentioned above. Open-source software is also good for long term projects as they are always available in the public domain. Users can easily improve the code's quality as they can easily spot bugs and either fix them or report the community. Due to the availability of open-source software, there is an increase in the possibilities of discovering vulnerabilities in the system easily (Melwani, 2019). However, open-source software is not intuitive or straightforward; it requires effort and training to be able to master the software. Also, there may be issues in version control as there may be ongoing parallel development of the software. It will present confusion on what functionalities are present in which version. Furthermore, the consumers are solely liable for maintaining compliance with legal obligation because it hardly provides safety, liability or warranty in that context (Wired Gorilla, 2021).

In conclusion, both the traditional and open-source software create software interface to meet the functional purpose of the end-users based on their requirement. This software is all made with the same purpose but differs in terms of affordability and distribution rights. Both systems have vulnerabilities that hackers can exploit; for open-source, these vulnerabilities can be identified and fixed by anyone who knows how to fix it. In traditional software, these vulnerabilities have to be fixed by the creators who made the software. The end-users (individual, company or organisation) is the person who has the right to choose what software best fits the purpose of their project or day to day activities.

5. Security and vulnerability of the grid

5.1 Introduction

WAMS systems today use several communication infrastructures such as analogue, digital, microwave, VPN, or SONET; each of these communication methods has vulnerabilities that can be used to disrupt communication or compromise the WAMS system.

When applied to PDCs, cybersecurity should assess all PDC interfaces while preserving the reliability of the PDC. In PDC the cybersecurity goes beyond securing synchrophasors communications but any communications and access. Because if security practices are poorly applied, they may degrade the performance and/or functionality of the PDC. A PDC may be connected to untrusted networks and PMU; securing these connections is more than just securing synchrophasors and is beyond this research. However, this section of the research focuses on the security issues that have occurred over the years and the security measures that are in place for these attacks.

Cybersecurity recommendations for WAMPAC:

- The security measures adopted should not in any way hamper the primary objective of the synchrophasor system.
- The availability of cybersecurity requirements assures the PMUs and PDC network servers must remain available to perform their primary functions promptly.
- The access to every PMU or PDC of a utility should be through an authentication procedure.
- The system should accept only authenticated and authorised changes in the configuration of the network.
- The transfer of information between different components in the synchrophasor system in WAMS such as PMU, PDU, GPS, control centre and applications must be confidential.
- Accountability must be achieved through the implementation of authorisation, authentication, auditing, and non-repudiation.
- There should be a proper mechanism to validate the integrity of data exchanged.
- The system should continue to perform essential functions in case of loss of synchronised measurements.
- The security mechanism should be able to minimise the impact of abnormalities on the performance of WAMS (Surender Kumar, 2015).

5.2 Types of attack

5.2.1 Replay Attacks

Relay attacks are similar to the Man-In-The-Middle attack where the criminal snoops on secure network communication, intercepting and then fraudulently delays or resends the message to misdirect the receiver into doing what the hacker wants. It creates a back door that allows the attacker to use the stolen network information to access the system. In the case of the gird, the attacker intercepts PMU estimations or the control messages by seizing the packets in transit between the PMU and the Phasor Information Concentrator (PDC) or possibly the control centre. In a few cases, a replay attack is conceivable under encrypted communication as the attack packets are substantial packets with the message's data integrity being intact except for the time-stamp data (Aditya Ashok, 2014).

5.2.2 Malware Attacks

Malicious software is a term used to refer to any software that is designed to cause damage to a computer, server, or network. Terms such as viruses, trojans, worms, phishing, ransomware, backdoors, and denial-of-service (DoS) all fall under malware attacks. In a power grid, malware can also be used for relay attacks where a hacker can attack a transmission in the network and leave a backdoor that can be used to access the grid network later.

5.2.2.1 Phishing

Examples of this attack is the December 2015 attack on the Ukrainian grid using a phishing email in which employees fell trap; this was used to cause black for around 230,000 people during the winter (Chester, 2020). In December 2016 cyberattack on the Ukrainian power grid by Russian Hackers planted a unique malware specimen in the Ukrenergo network that caused a blackout for one hour. It was the first recorded power grid attack that has been documented.

5.2.2.2 Denial-of-Service

DoS is a common attack threat on the Synchrophasor, and this happens when an attacker compromises the availability of an information system. If an attacker manages to gain access to the communication infrastructure, they can launch an attack by overloading the network device, which are important links with fake traffics using packets. Such attacks can cause disruption or reduction in the transfer of real-time measurement data from the field devices to the controller. With the rise in the wireless network use in the WAMS, there can be jamming attacks in the substation, which is primary security (Surender Kumar, 2015). An

example of DoS is the attack in the western United States by exploiting a vulnerability in the firewall firmware. According to the North American Electric Reliability Corporation (NERC) report, "Lesson Learned" in September 2019 on the cyberattack on a low-impact grid control centre and several small power sites states that the attack was a brief outage of internet-facing firewalls that controlled the communication between multiple remote generation sites and the control centre, including the equipment's on these sites. Although each communication failure took less than five minutes, the entire attack lasted for ten hours (Crowell Moring, 2019).

5.2.2.3 Virus

A virus is another form of malware that can be used to target the power grid. In 2012 Shamoon was first discovered in a cyber-attack in Saudi Arabia; the virus has hit both the government ministries and petrochemical firms. Shamoon is a virus that can spread from an infected machine to other computers within the network. When a computer is infected, the virus continuously compiles a list of files from the computer's location, sends them to the attacker, and deletes them. Saudi Aramco, which pumps 10% of the global oil supply, was a victim of this cyber-attack, it aims was to stop the production of oil and gas by the biggest OPEC exporters in the world, and around 30,000 computers were damaged, and data was wiped (Rashad, 2020).

5.2.3 Data Integrity using Injection.

Data integrity attacks are attacks where the information is corrupted within the forward or the reverse within the control stream. This attack is very dangerous in grid systems, especially when it leads to false results in data analysis and management of the grid. The main aim of manipulation is to compromise the integrity of the data, and the damages cannot be undone. Also, data integrity attacks can render important information to be unusable and impact the credibility and reliability of the data from the connected device in the grid. These attacks are not always apparent to the victims unless they notice the odd reading or breach in the system. Using injection methods (Packet injection attack), an attacker can corrupt the data integrity, which specifically debases the sensor data. In the case of WAMS, this can be an actuator or PMU data which is the command given to the protection components or the Volt-Ampere Reactive (VAR) control components. It interprets activities like blocking off the trip signals in scenarios where the controller sent a trip command to the security components or the controller commanded to extend VAR injection (Aditya Ashok, 2014).

5.3 Security measure is in place for these attacks

There are known measures or solutions to cyber-attacks; there are security measures to protect it from external attacks and an information security measure.

5.3.1 Security gateway

A known method of protecting the substation from cyber-attack is using a security gateway to limit external exposure and secure the access points. These security gateway devices provide the network with the same properties as VPN tunnelling and Firewall. Firewalls restrict the incoming and outgoing network traffic based on a set of policies or user-defined rules. It is recommended to use a white-list approach when setting up the rules for the Firewall; this is because, in a white-list practice also known as deny-by-default, all traffic is blocked except it is explicitly allowed by a rule (John Stewart, 2010). Network firewall 1994 publication by Bellovin et al. states that a firewall has to meet these properties "All traffic from inside to outside must pass through the firewall, only authorised traffic as defined by the local security policy will be allowed to pass and the firewall itself is immune to penetration". He stated these properties are design goals, and a good firewall must meet all three requirements (Steven M. Bellovin, 1994).

Using UDP Secure for unidirectional streaming using IEEE C37.118 is suitable for protecting data traffic over an untrusted network. UDP secure allows more straightforward and more restrictive firewall rules because of its unidirectional feature. The IEEE C37.118 does not have built-in authentication mechanisms, and therefore is susceptible to spoofing attacks. To allow integrity and preserve data confidentiality in a UDPS, it is advisable to transport the datagrams through a VPN. Using a VPN and UDPS gives more robust security even in bidirectional data transfer between two locations. Furthermore, it is critical for the network security that all communication routes through this gateway, and there is no other connection or bridging between the two networks (John Stewart, 2010).

5.3.2 Authentication

An excellent solution to prevent unauthorised access to the WAMS or WAMPAC system is using a robust authentication mechanism. Authentication is a security measure devised to protect the communication system from illegal transmissions or simulations by determining the authenticity of the transmission, communication, or initiator. Corporations are advised to implement an implicit deny policy on the network by only granting access through explicit permissions. It is important that WAMS systems use an authentication mechanism because it can be influenced by injected traffics being accepted and enacted by the PDC, state

estimators and other applications. An example of an authentication method is LDAP authentication. LDAP authentication is software that stores and arrange data to make it easily accessible. The main purpose is to serve as a central hub for authentication and authorisation by validating a username and password with a directory server such as Active Directory (e.g., Microsoft AD, OpenLDAP, OpenDJ etc.). It also provides an efficient level of security for organisations to deploy WPA2 (Metzler, 2020). For secure communication, LDAP transactions must be encrypted using an SSL/TLS connection.

5.3.3 Encryption

Another solution to maintain the confidentiality and integrity of data across an untrusted network is encryption of communication on the data link layer by converting the data into ciphertext. Encryption is the method of translating data into a secure format that approved parties can only read. It takes readable data and alters it so that it appears at random. There are two types of encryption scheme: symmetric cryptography key (DES, AES and GCM) that uses the same key to encrypt and decrypt the communication. Asymmetric cryptography key (RSA and DHKE) uses private and public keys to encrypt and decrypt communication, respectively.

5.3.3.1 Symmetric

Data Encryption Standard (DES) is a mode of operation symmetric cypher that IBM developed; the cypher was based on Lucifer under the influence of the NSA. It has an encryption block size of 64bits and uses a key length of 56bit. DES uses 16 rounds which all perform the same operation, and the different subkey in each round is derived from the primary key (Private key). 3DES is a more secured type of DES because it uses 2keys; each key is 56bits giving it a strength of 112bits. The disadvantage of this key is that it is slow. The DES is replaced with the AES in the early 2000s. Advanced Encryption Standard (AES) is an encryption algorithm that generates ciphertext through several iterative recalculations. It has an encryption block size of 128bits and uses key lengths of 128bits, 192bits or 256bits. Galois Counter Mode (GCM) is an authenticated encryption algorithm designed to provide message authentication and message integrity. It uses the chained Galois Field multiplication for the additional properties and has a block size of 128bits. The AES with GCM is widely used due to their performance and are considered to be very secure. Using any of the functions for authenticated encryption means that both integrity and confidentiality is guaranteed.

5.3.3.2 Asymmetric

Rivest-Shamir-Adleman (RSA) uses a pair of keys for encryption and decryption; it has a key size of 2048bits and above. The algorithm assumes that there is no efficient way to factor in vast numbers. Therefore, deducing the RSA key needs an exceptional amount of computer processing time and power. It is mainly used for key exchanges and digital signatures. Diffie Hellman Key Exchange (DHKE) is a key exchange protocol and not used for encryption. It uses discrete logarithms; this allows two users to exchange symmetric keys through a secured channel (wired or wireless). DHKE is used widely by all major VPN gateways (Surender Kumar, 2015). One vulnerability of DHKE is the strength of the key that it generates. Even when using high integers that generate keys with arithmetic seven-figure values, it is still insignificant for an attacker to guess it by brute force.

5.3.3.3 Hash functions

Hash functions are usually mathematical compositions with certain specific properties. First, it is a one-way function. Any data you enter will be converted to results and cannot be converted back to retrieve the original input. Second, the output length is the same regardless of its input length/size or content. There are different types of Hash functions such as the MD1-6, SHA-0/1/2/3, RIPEMD and Whirlpool. The Hash functions protect password storage. It is also used for data integrity check, which helps detect changes made to the original file by generating a checksum on data files.

5.4 Conclusion

WAMS systems use several communication infrastructures such as the SDH, SONET, WDM, analogue microwave, etc. These communication systems all have vulnerabilities that can be explored to interrupt communication or compromise the system. WAMS operates in an environment of complete and implicit trust. WAMS running on the software layer of the power grid requires both securities from the hardware component that interacts with the software and security on software in terms of encryption, authentication and authorisation of the communication and access to data. The companies that use this system have to be aware of the existence of the vulnerabilities and ways to protect their infrastructures from attacks.

6. Vulnerability scan on OpenPDC and OpenHistorian

6.1 Introduction

This section explains the approach to the problems described in the introduction to look into the software used in the WAMS environment by exploring their vulnerabilities. The process used is described in details stating what software is used in each test case and the expected and actual results from the test carried out.

Due to the nature of the research, getting commercial software from any of the companies to carry out the analysis was impossible. As advised by one of the engineers from a Grid solution company, it resolves to look at open-source software as they are beginning to get recognised and used in the industry. Researched and decided to use the Open Historian (OpenHistorian) for the analysis rather than the OpenPDC. The OpenPDC had connectivity issues that were unfeasible to resolves within the time frame set for the analysis. It may not be the same result as a working Historian or PDC in a substation or control centre, but it will give the same insight into the functionalities as those used in those environments.

6.2 The Methodology

6.2.1 Approach

For the research approach, a descriptive method will be used because of the complex nature of the research topic. Access to both primary and secondary data for this program is limited; therefore, sample data available online and within the Grid Protection Alliance community will be used to carry out analysis on the software. Unlike the experimental research method, this method does not require expertise in the subject matter, and judgement can only be based on the result of the data as collected.

Security testing is not a complete science where an exhaustive list of all potential issues that ought to be tried can be characterised. Security testing is a fitting procedure for testing the security of web applications under certain circumstances. The Historian uses a web interface that can be used to view the data streams in real-time and analyse the data that has been stored. For this reason, the OWASP Web Application Security test methods will be used to run the security testing for the software. The test method uses a balance approach which is a combination of manual review, penetration testing, security scanning and vulnerability scan. The test guide will be followed closely to analyse the software to eliminate any assumptions. The black box approach will be used because It is easy to follow, and testers need to have all the information about the application to carry out the test.

Penetration testing

A penetration test (Pentest) is a simulated cyber-attack against a computer system to check for exploitable vulnerabilities. The advantage of using this method is that it is fast, requires a relatively low skill set than source code review, and actually testing the codes exposed to those vulnerabilities. However, this test can only impact the front end of the application. The Pentest will be run in both passive and active mode, depending on the software that is used. ZAP uses both passive and active mode to check the vulnerability of a system.

6.2.2 Software

- Microsoft SQL Server 2008 R2
- Microsoft SQL Server Management Studio 2016
- OpenPDC Manager
- PMU Connection Tester
- OpenHistorian
- OpenVisN
- Nessus
- ZAP
- Wireshark

There are three stages to this approach on how the WAMS data can be acquired. OpenPDC is used to obtain the data from PMU. OpenHistorian issued to save the received data from the PMU. OpenVision is used to analyse and display data held in the Historian.

6.2.2.1 OpenPDC

Open-source Phasor Data Concentrator is a complete phasor data concentrator software that GPA designed to process streaming time-series data in real-time. This is the software version of the WAMS component that was discussed in Section [3.3.3]. It is not just a data concentrator; it is also a flexible platform for processing high-speed time series data, adapting to changing technologies to provide a future-oriented phasor data architecture. OpenPDC can be used to allocate data (real-time and historical data) to the applications used and can be installed anywhere within the synchrophasor infrastructure, even on computers operating in a substation environment. Although the main purpose of openPDC is to centralise and manage real-time flow synchronisation phasors through the functions of the GPA-based time-series library, which inherits the modular design and can be classified as an event flow processor general. The supported protocols include IEEE 1344, IEEE C37.118, IEC 61850-90-5,

BPA PDC stream etc., for this research, I will be using the IEEE 1344 and IEEE C37.118 protocol because these are the two that the sample data can be applied.

6.2.2.2 OpenHistorian

The Open-source Historian is software designed by Grid Protection Alliance for data storage and analysis. It is software with high performance, NoSQL database designed to store synchorphasor effectively, SCADA and other control data to support real-time grid operations and post-operation data analysis. This is the open-source version of the WAMS component, as discussed in Section [3.3.53.3.5]. Some benefit of the software includes lossless compression that reduces storage cost and maintains full data accuracy, and it supports backfilling of data and out of sequence data insertion. The software also supports multiple data types (DNP3, COMTRADE, and MODBUS) and uses the same protocol as the OpenPDC, including Macrodyne (GPA, 2018). The software comes with an inbuilt visualisation system known as OpenVisN. Open-source Vision application is a visualisation and analysis tool for the openHistorian that displays parallel data charts over a provided time window. Multiple values can be quickly trended together to drill down into a particular data window of interest, such as an event.

6.2.2.3 SQL Server

Microsoft SQL Server is a relational database management system (RDBMS) developed in 1989 by Microsoft. As a database server, the main function is to store and retrieve data used by other applications, and the server can run either run on the same device as the other applications or on another device across a network. The version of the SQL server used for the research is 2008 because it is the version that is compatible with the available version of PDC and Historian. For data management of this project, the SQL Server integration services (SSIS) will be used to monitor the database and the SQL Server Management Studio 2016(SSMS).

6.2.2.4 Nessus

Nessus is a vulnerability scanner software created by Tenable security; it Is used during penetration testing, vulnerability assessments, including malicious attacks. For this research, I am using the Nessus Essential, a free version of the Nessus Professional. This version of the Nessus software helps perform high-speed asset discovery, target profile, configuration audit, control systems auditing (SCADA devices), malware detection, sensitive data discovery, etc. Nessus Professional runs on client devices and can be used effectively by the security department within an organisation. Nessus can be used to perform both automated

vulnerability scan and manual vulnerability scan. The downside of an automatic vulnerability scan is that it is not made to find a specific pattern but will generalise the search.

6.2.2.5 ZAP

Zed Attack Proxy (ZAP) is an open-source web vulnerability scanner developed by Open Web Application Security Projects (OWASP). OWASP is an open, online community that creates tools, technologies, guidance, and methodologies to provide secure web applications. The applications can be used to detect issues such as broken access control, exposure of sensitive data, cross-site scripting, security misconfiguration, insecure deserialisation, SQL injection, the vulnerability in components, broken authentication etc. The OWASP tool is the only one selected for this research because it Is the only software that could carry on the assessment on both the web application and the Windows interface. Unlike Burp proxy, the ZAP is free software that can run most of the scan without paying to use specific features that it provides, such as the security alert list.

6.2.2.6 Wireshark

Wireshark is an open-source network protocol analyser created in 1998 by Gerald Combs. The software allows individuals or organisations to analyse data traffic and signals across the communication network. The benefit of this software is that it a free open-source product that can track all activities on the network. It allows IT professionals to identify network issues, analyse and troubleshoot errors, also define communication protocols (Petters, 2020). Furthermore, the software can be used with any operating system such as Window, Linux or Mac. For this project, the software will be used as a packet sniffer and capture network traffic that will be stored for offline analysis. An alternative to this is the SolarWind network performance monitor, but it is expensive.

6.2.3 Set-up

6.2.3.1 Configuring the SQL server

- Launch SQL Server Management Studio Express and connect to your database server.
- In the toolbar, go to "File > Open > File..."
- Navigate to Database Scripts\SQL Server. Otherwise, navigate to sourcedir\openPDC\Database Scripts\SQL Server, select "openPDC.sql", and select "Open".
- In the toolbar, go to "Query > Execute".

- Repeat steps 2-4 with the files "InitialDataSet.sql" and "SampleDataSet.sql" in the same directory.

Modify configuration file for OpenPDC

- Navigate to sourcedir\OpenPDC, select "openPDC.exe.config" or “openPDC.config”, open with any text editor.
- Change the attributes to the connectionString line.
- Replace the *serverName* with the name of the database server.
- Replace username with your *username*.
- Replace password with your *password*.

6.2.3.2 Configuring the OpenPDC

- Make sure the SQL server is installed and running.
- Run OpenPDC.
- Follow the instruction to set up the OpenPDC.
- Choose Windows authentication.
- Finish setting up using the SQL server username and password.

6.2.3.3 Running the OpenPDC

- Go to "Start > Run...".
- Type "services.msc" into the text box and click "OK".
- Find "openPDC" in the list, right-click it, and click "Start".

6.2.3.4 Configuring the PMU Connection Tester

Creating and verifying an IEEE C37.118-2005 data stream

- Run openPDC.
- Go to the PMU Connection Tester window and select the "UDP" tab toward the top of the window.
- In the text box labelled "Local Port", enter "8800".
- In the drop-down list under the "Protocol" tab, select "IEEE C37.118-2005".
- Still, under the "Protocol" tab, click "Configure alternate command channel".
- Clear the check box labelled "Not defined".

- In the text box labelled "Port", enter "8900" and click "Save".
- Still, under the "Protocol" tab, click "Connect".

6.2.3.5 Configuring the OpenHistorian

- Make sure the SQL server is installed and running.
- Run OpenHistorian.
- Follow the instruction to set up the Historian.
- Choose Windows authentication.
- Finish setting up using the SQL server username and password.

6.2.3.6 Running the OpenHistorian

- Go to "Start > Run...".
- Type "services.msc" into the text box and click "OK".
- Find "openHistorian" in the list, right-click it, and click "Start".

6.2.3.7 NTLM configuration for Firefox

- In the Location bar, type about:config and press Enter.
- In the about:config page, search for “network.automatic-ntlm-auth.trusted-uris”, and double-click on it.
- In the prompt that comes up, Localhost:8180.
- Press OK.

The setting should stop any prompt for authentication using Firefox.

6.3 Analysis test cases

Using the OWSAP Web Application security test framework will use 9 out of the 11 categories: information gathering, error handling, cryptography, input validation, authentication, authorisation, session management, identification management, configuration, and deployment managed. These sub-categories will be used because they are important and more feasible with the WAMS software. The focus will be on the analysis of the openHistorian/Vision.

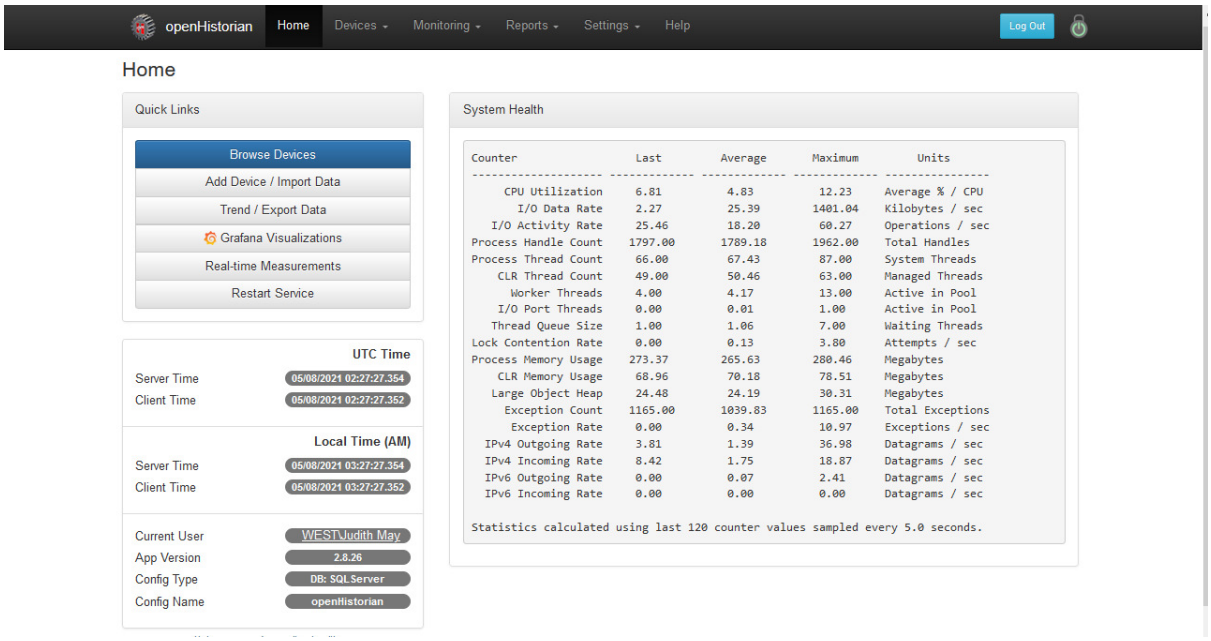


Figure 4: Historian web interface

6.3.1 Information gathering

Information gathering is an important phase in pen-testing; it helps to understand the server configuration, which is as critical as the application security test. After all, an application string is only as consistent as its faintest interface. Application platforms are wide and varied, but a few fundamental setup errors can compromise the application, which means that an unsecured application can compromise the server.

No	Test Name	Description	Expected result	Tools
1	Review Webserver Metafiles for information leakage	This test is to analyse robots.txt and identify <META> Tags from the website.	The response should not have information leakage on the directory or folder path.	Browser
2	Review Webpage Comments and Metadata for Information Leakage	The test is to find sensitive information from webpage comments and Metadata on source code.	There should not be any sensitive information or web comment in the source code.	Browser

3	Identify application entry points	This test is to identify entry from hidden fields, parameters, methods HTTP header analysis	The parameters should not show any hidden entry fields.	ZAP
4	Map execution paths through application	This test is to map the target application and understand the principal workflows.	The test should be able to show the structure of the website.	ZAP

Table 2: Test case for information gathering.

6.3.2 Configuration and deploy management.

This test is to check the configuration and deployment of the system. It involves checking the server connection and ensuring that all links are running properly, and the necessary policies are in place. They are also checking that the information is transported in a secure layer.

No	Test Name	Description	Expected result	Tools
1	Application Platform Configuration	This test is to Identify default installation file/directory, Handle Server errors (40*,50*), Minimal Privilege, Software logging.	There should be less 40* and 50* logs if the configuration is done correctly.	Browser
2	File Extensions Handling for Sensitive Information	This is to find important file, information (.asa , .inc , .sql ,zip, tar, pdf, txt, etc)	There should not be any sensitive file extension or link in the source code.	Browser
3	HTTP Strict Transport Security	This test is to identify the HSTS header on the Web server through the HTTP response header.	All communication should use HTTPS or upgrade to a secure channel on the webserver.	ZAP, Wireshark

4	RIA cross domain policy	The test is to analyse the permissions allowed from the policy files and allow-access-from.	There should be a policy file and allow access policy.	ZAP

Table 3: Test case configuration and deployment management

6.3.3 Identification management

Identity management tests the weaknesses in the identification process, such as user registration. It checks for the role definition used to manage users and their authorisation to system resources. This test helps to check for attack areas such as using default name and password or brute-force attack, opportunities where an attacker can create a false username and password. Or if an attacker can create a dummy account that can be used later to get into the system.

No	Test Name	Description	Expected Result	Tools
1	Role Definitions	This test aims to validate the system roles defined within the application by creating a permission matrix.	The user roles should be clearly defined.	ZAP
2	User Registration Process	This test aims to verify that the identity requirements for user registration are aligned with business and security requirements.	The openHistorian should have a mechanism for new users to register.	ZAP
3	Account Provisioning Process	This test is to determine which roles can provision users and what sort of accounts they can provision.	Only an administrator should provision an account on the OpenHistorian.	ZAP

4	Account Enumeration and Guessable User Account	This test is to check generic login error statement check, return codes/parameter values, enumerate all possible valid user ids (Login system, Forgot password)	The openHistorian should show successful authentication when using a valid username and password. When using an invalid username and password, it should show authenticated fail.	Browser, ZAP
5	Weak or unenforced username policy	This test checks if user account names are often highly structured and valid account names can easily be guessed.	There should be an error message if an invalid account that does not match the username structure is used.	Browser, ZAP

Table 4: Test case for identification management

6.3.4 Authentication testing

Authentication is the process of validating the digital identity of the sender of a transmission. Testing the authentication schema helps to understand how the authentication process works using the information given to evade the authentication mechanism.

No	Test Name	Description	Expected Result	Tools
1	Credentials Transported over an Encrypted Channel	This is to check the referrer, whether it is HTTP or HTTPS. Sending data through HTTP and HTTPS	Credentials should be transferred over HTTPS for secure connection	ZAP
2	Weak lockout mechanism	This is to evaluate the account lockout mechanism ability to	The software should have a strong lockout mechanism	Browser

		mitigate brute force password guessing and resistance to unauthorised account unlocking.	that allows the user a limited trial using the wrong password.	
3	Bypassing authentication schema	This is to test authentication using Force browsing, Parameter Modification, Session ID prediction, SQL Injection	The software should not allow users to bypass authentication by editing the request	ZAP
4	Remember password functionality	This test is to look for passwords that are stored in a cookie. Examine the cookies stored by the application. Verify that the credentials are not stored in clear text but are hashed.	For security reason, the software should not allow a remember password functionality in any browser.	ZAP
5	Browser cache weakness	This test is to check browser history issue by clicking the "Back" button after logging out.	The software does not store sensitive information on the browser by using no-cache	ZAP
6	Weak password policy	This test determines the application's resistance against brute force password guessing using available password dictionaries by evaluating the length, complexity, reuse, and ageing requirements of passwords.	The software should have a weak password policy that allows a password within a certain length and complexity.	ZAP

7	Weak password change or reset functionalities	This is to test password reset (Display old password in plain-text?), test password change (Need the old password?), CSRF vulnerability?	The software should have a secure functionality to allow users to change their password. And hash the password input.	ZAP

Table 5: Test case for authentication

6.3.5 Authorisation testing

An authorisation is the concept of allowing access to resources only to those permitted to use them. The authorisation is the process that comes after successful authentication, verifying that the user has valid credentials associated with a role and privilege. The purpose of this test is to verify if it's possible to bypass the authorisation schema.

No	Test name	Description	Expected Result	Tools
1	Directory traversal/file include	This test the dot-dot-slash attack (../), directory traversal, Local File Inclusion/Remote File Inclusion.	It should not be possible to access remote or local files.	ZAP
2	Bypassing authorisation schema	This test is to check if an attacker can access a resource without authentication? Bypass ACL, Force browsing.	Only the authorised user should access functions such as adding new users or assigning roles.	ZAP
3	Privilege Escalation	Testing for role/privilege manipulates the values of hidden variables. Change some param groupid=2 to groupid=1	Only the administrator should escalate a user privilege.	ZAP
4	Insecure Direct Object References	This test is to check if an attacker can force changing parameter value to access information	Pages or information cannot be accessible through direct object reference	ZAP

Table 6: Test case for authorisation

6.3.6 Session management

Session management is defined as the set of all controls governing full interaction between a user and the web-based application. The session mostly covers everything from when user authentication is performed to the user logging out of the web application. This checks if there are any sensitive information left behind in the cookie and cache during/ after the use of the application.

No	Test Name	Description	Expected Result	Tools
1	Bypassing Session Management Schema	This test for vulnerabilities in SessionID analysis prediction, unencrypted cookie transport, brute-force.	The sessions and cookies should be encrypted.	ZAP
2	Cookies attributes	This test checks for cookie attribute vulnerabilities by checking HTTPOnly and Secure flag, expiration, inspect for sensitive data.	The cookie attribute should have clear paths, domain and HTTP Only attributes.	ZAP
3	Session Fixation	This test that the application doesn't renew the cookie after successful user authentication.	The application should renew the cookie after a successful user authentication	ZAP
4	Exposed Session Variables	Test for Encryption & reuse of session token's vulnerabilities, Send sessionID with GET method.	The cache-control directives should be used to ensure the cache does not expose the data.	ZAP
5	Cross Site Request Forgery	Tests URL analysis, direct access to functions without any token.		ZAP

6	Session Timeout	Checks session timeout; after the timeout has passed, all session tokens should be destroyed or be unusable.	The session must-revalidate each time the session timeout	ZAP

Table 7: Test case for session management

6.3.7 Input validation

This testing aims to validate input coming from the client, which can be a security weakness if not properly validated. This test all the possible input forms to understand if the application sufficiently validates input data before processing it. It is important because weakness from the input can lead to major vulnerabilities in the web application such as SQL injection, system file attacks, cross-site scripting etc.

No	Test name	Description	Expected Result	Tools
1	Reflected cross site scripting	This test checks for input validation; replace the vector used to identify XSS, XSS with HTTP Parameter Pollution.	The OpenHistorian web interface should not allow data entry scripting through the Url to be executed.	ZAP
2	Stored cross-site scripting	This is to check input forms/Upload forms and analyse HTML codes.	The OpenHistorian should not store user input in the cache or any form that an attacker could access.	ZAP
3	HTTP parameter pollution	This is to Identify any form of action that allows user-supplied input to bypass Input validation and filters using HPP	The openHistorian should not allow user input that can bypass validation using HPP.	ZAP
4	SQL injection	This test checks the possibility of an SQL injection using Union,	The OpenHistorian should not be	ZAP

		Boolean, Error based, Out-of-band, and Time delay.	vulnerable to SQL injection attacks.	
5	XML injection	This is to check for possibility of XML injection using XML Meta Characters ', " , <>, <!--/-->, &, <![CDATA[/]]>, XXE, TAG	The OpenHistorian should not be vulnerable to XML injection attacks.	ZAP
6	SSI injection	This test is to check for the presence of .shtml extension, Check for these characters < ! # = / . " - > and [a-zA-Z0-9] including String = <!--#include virtual="/etc/passwd" -->	The OpenHistorian should not be vulnerable to SSI injection attacks.	ZAP

Table 8:Test case for input validation

6.3.8 Error Handling

Error handling section checks for more common errors and brings into focus their relevance during the assessment. The error codes are critical because they reveal a lot of information about the bugs, databases and other technical components that are linked directly to the web application. Attackers can sometimes use search engines to locate errors that disclose sensitive information.

No	Test Name	Description	Expected Result	Tools
1	Analysis of Error codes	This test is to locate error codes generated from applications or web servers. Collect sensitive information from that errors (Web Server, Application Server, Database)	The OpenHistorian should show the list of error codes related to the webs server, application, and database.	ZAP

2	Analysis of Stack traces	This test finds information on invalid/empty inputs, input that contains non-alphanumeric characters or query syntax. Access to internal pages without authentication and bypassing application flow.	The OpenHistorian should stack trace as part of error handling when invalid input or non-alphanumeric characters are entered.	ZAP, Browser

Table 9: Test case for error handling

6.3.9 Cryptography

Sensitive data must be protected when it is transported through the network. For security reasons, data must be protected when it is stored and protected when transferred. This section tests the web application to see if sensitive data are encrypted and how they are encrypted.

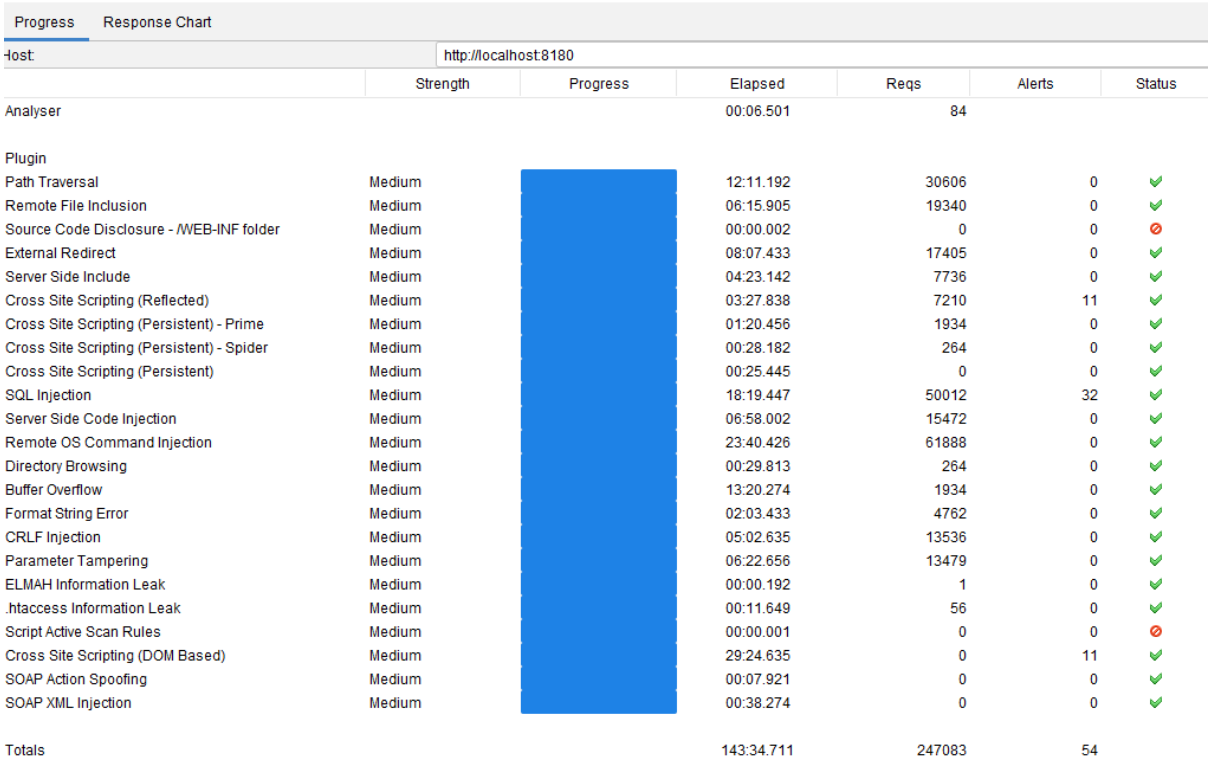
No	Test Name	Description	Expected Result	Tools
1	Testing for Weak SSL/TSL Ciphers, Insufficient Transport Layer Protection	This test is to identify SSL service and weak ciphers/protocols	The OpenHistorian should have strong ciphers/protocols in place.	ZAP
2	Testing for Sensitive information sent via unencrypted channels	This test is to check sensitive data during the transmission, such as information used in the authentication and information protected by laws, regulations, or specific organisational policy	All sensitive information should be encrypted when sending through an unsecured channel	ZAP

Table 10: Test case for cryptography

6.4 The result

The tables below show the actual result of the test. It also includes the explanation of the test that failed and a sample screenshot of evidence of the code. The result is classified into three types: pass, fail and issue.

- Pass – If the actual result is the same as the expected result.
- Fail- If the actual result is different from the expected result and has a high risk.
- Issue – If the test did not fail but had raised issues that are classified as medium or low risk.



Progress | Response Chart

Host: http://localhost:8180

	Strength	Progress	Elapsed	Reqs	Alerts	Status
Analyser			00:06.501	84		
Plugin						
Path Traversal	Medium		12:11.192	30606	0	✓
Remote File Inclusion	Medium		06:15.905	19340	0	✓
Source Code Disclosure - /WEB-INF folder	Medium		00:00.002	0	0	⊘
External Redirect	Medium		08:07.433	17405	0	✓
Server Side Include	Medium		04:23.142	7736	0	✓
Cross Site Scripting (Reflected)	Medium		03:27.838	7210	11	✓
Cross Site Scripting (Persistent) - Prime	Medium		01:20.456	1934	0	✓
Cross Site Scripting (Persistent) - Spider	Medium		00:28.182	264	0	✓
Cross Site Scripting (Persistent)	Medium		00:25.445	0	0	✓
SQL Injection	Medium		18:19.447	50012	32	✓
Server Side Code Injection	Medium		06:58.002	15472	0	✓
Remote OS Command Injection	Medium		23:40.426	61888	0	✓
Directory Browsing	Medium		00:29.813	264	0	✓
Buffer Overflow	Medium		13:20.274	1934	0	✓
Format String Error	Medium		02:03.433	4762	0	✓
CRLF Injection	Medium		05:02.635	13536	0	✓
Parameter Tampering	Medium		06:22.656	13479	0	✓
ELMAH Information Leak	Medium		00:00.192	1	0	✓
.htaccess Information Leak	Medium		00:11.649	56	0	✓
Script Active Scan Rules	Medium		00:00.001	0	0	⊘
Cross Site Scripting (DOM Based)	Medium		29:24.635	0	11	✓
SOAP Action Spoofing	Medium		00:07.921	0	0	✓
SOAP XML Injection	Medium		00:38.274	0	0	✓
Totals			143:34.711	247083	54	

Figure 5: Analysis results

6.4.1 Information gathering

No	Test Name	Pass/Fail/Issue	Actual Result	Comment
1	Review Webserver Metafiles for information leakage	Pass	The response did not have information leakage on the	The metafiles on the webpage are hidden. There was no information related to

			directory or folder path.	the files that were sent for the test.
2	Review Webpage Comments and Metadata for Information Leakage	Issue	There was a visible comment in some of the source code.	The webpage has a visible comment with authors names, license, and version.
3	Identify application entry points	Pass	The response parameters did not show any hidden entry fields, methods	There is no way to identify entry from hidden fields, parameters, methods HTTP header analysis
4	Map execution paths through the application	Pass	There was a clear structure of the application and the principal workflows.	

Table 11: Result of information gathering.

The response appears to contain suspicious comments which may help an attacker. Matches that can be made within script blocks or files can be used are against the entire content.

```
HTTP/1.1 200 OK
Cache-Control: must-revalidate, no-cache
Content-Length: 52650
Content-Type: application/x-javascript
ETag: "1214973845"
Server: Microsoft-HTTPAPI/2.0
Set-Cookie: x-gsf-session=83a2512c-6759-4e60-9ae4-3e60be70581b; path=/
Date: Sat, 08 May 2021 20:21:06 GMT
```

```
//! moment.js
//! version : 2.20.1
//! authors : Tim Wood, Iskren Chernev, Moment.js contributors
//! license : MIT
//! momentjs.com
(function(n,t){typeof exports=="object"&&typeof module!="undefined"?module.exports=t():
typeof define=="function"&&define.amd?define(t):n.moment=t()})(this,function(){
```

Figure 6:Evidence of webpage comment.

6.4.2 Configuration and deploy management.

No	Test Name	Pass/Fail/Issue	Actual Results	Comment
1	Application Platform Configuration	Pass	There are few 40* logs in history.	There are fewer 40* logs and no 50*logs in history, which means the configuration setting was successful
2	File Extensions Handling for Sensitive Information	Pass	There was no indication of sensitive information file (.asa , .inc , .sql ,zip, tar, pdf, txt, etc) on the links and webpages.	
3	HTTP Strict Transport Security	Pass	There is strict transport security on the HTTP to upgrade to a secure channel on the Web server	
4	RIA cross-domain policy	Fail	There are no policy files, and allow-access-from is overly permissive.	The web application has an overly permissive cross-domain policy.

Table 12:Result of configuration and deployment management

Web browser data loading may be possible due to a Cross-Origin Resource Sharing (CORS) misconfiguration on the webserver. The CORS misconfiguration on the webserver permits cross-domain read requests from arbitrary third-party domains, using unauthenticated APIs on this domain. Web browser implementations do not allow random third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. An attacker

could use this misconfiguration to access data that is available in an unauthenticated manner but uses some other security form, such as IP address white-listing.

Figure 7: Evidence of RIA Cross-Domain Policy

6.4.3 Identification management

No	Test Name	Pass/Fail /Issue	Actual Result	Comment
1	Role Definitions	Pass	User roles are clearly defined in the manager	
2	User Registration Process	Pass	The openHistorian have a way for user registration, but this can be done on the historian manager, not the web interface.	The admin gives new users access. New users can be registered using the OpenHistorian manager or on the webpage under settings. There are no options for the forgotten password; this has to be changed by an administrator.
3	Account Provisioning Process	Pass	Only an administrator can provision other accounts.	However, if you have access to the admin account, you can create provision for any account.
4	Account Enumeration and	Pass	When using a valid username and password,	

	Guessable User Account		a successful authentication message was shown. When using an invalid username and password, and an authenticated failed message was shown	
5	Weak or unenforced username policy	Pass	There is a consistent error message if an invalid account irrespective of the username structure.	

Table 13: Result of identification management

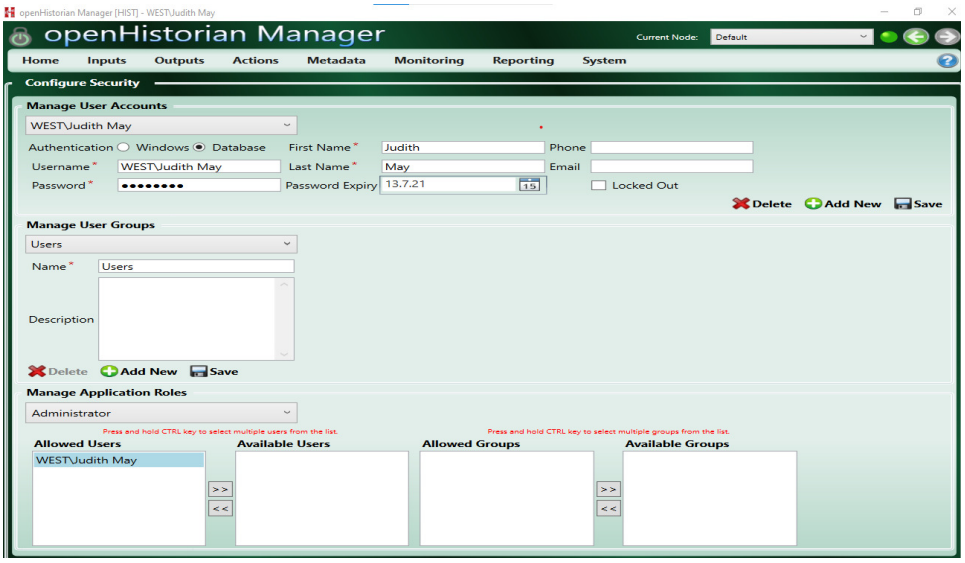


Figure 8:Evidence of Role configurations

6.4.4 Authentication testing

No	Test Name	Pass/Fail/Issue	Actual Result	Comment
1	Credentials Transported over an Encrypted Channel	Fail	The credentials are transferred over HTTP instead of HTTPS.	This result depends on the browser that is used.

2	Weak lockout mechanism	Fail	The software does not have a lockout mechanism in place.	I tried a total of 20times and was still allowed to keep trying the wrong password
3	Bypassing authentication schema	Fail	The software allows users to bypass authentication by editing the request	The authentication test was able to get a response after changing.
4	Remember password functionality	Pass	The software does not permit a remember password function in place.	With the functionality in place, you either have to reinstall the software if you forget the password.
5	Browser cache weakness	Pass	The software does not store sensitive information on the browser by using no-cache	The alert level on this is low because there is not
6	Weak password policy	Pass	The software has a weak password policy as it does not allow user password less than eight letters	It also indicates that it must contain numbers, Capital, and characters. However, using window authentication without a password is acceptable.
7	Weak password change or reset functionalities	Pass	The software does not have the functionality to allow users to change their password.	Only the administrator can change the

				password and issue a new one.

Table 14: Result of authentication

An insecure authentication mechanism is used in the openHistorian. It allows an attacker on the network access to the user-id and password of the authenticated user. For Basic Authentication, the attacker must merely monitor the network traffic until a Basic Authentication request is received and access to the username and the password, if the hash (including a nonce) can be successfully cracked, or if a Man-In-The-Middle attack is mounted. The attacker eavesdrops on the network until authentication has completed.

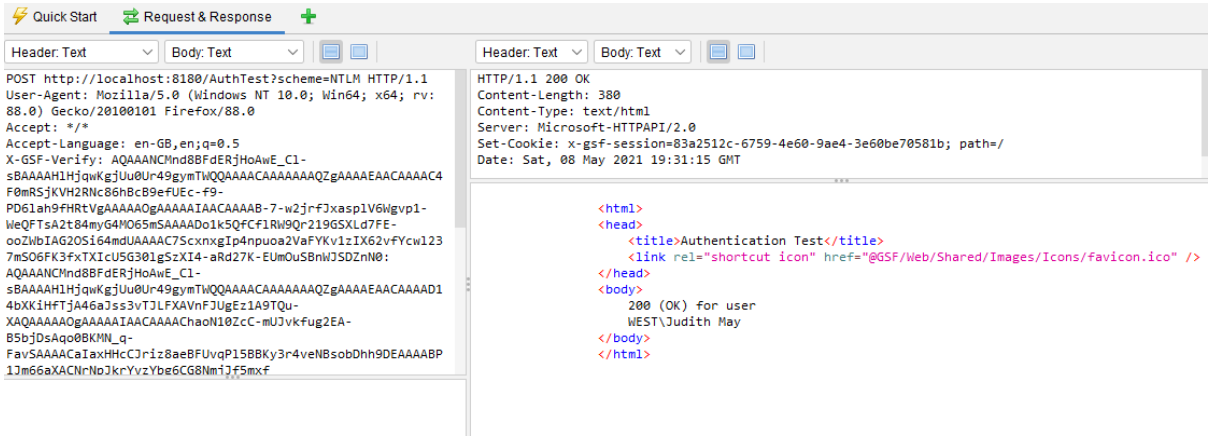


Figure 9:Evidence of bypassing authentication schema

6.4.5 Authorisation testing

No	Test name	Pass/Fail/Issue	Actual Result
1	Directory traversal/file include	Pass	It is not possible to access remote or local files.
2	Bypassing authorisation schema	Pass	Only the authorised users can access some function such as adding new users or assigning roles.
3	Privilege Escalation	Pass	Only the admin can escalate a user privilege
4	Insecure Direct Object References	Pass	Pages or information cannot be access through direct object reference

Table 15: Result of authorisation

6.4.6 Session management

No	Test Name	Pass/Fail/Issue	Actual Result
1	Bypassing Session Management Schema	Pass	The session and cookies are encrypted.
2	Cookies attributes	Issue	The cookie attributes only have insecure paths, and some have an expiring date.
3	Session fixation	Pass	The application renewed the cookie after successful user authentication.
4	Exposed session variables	Pass	The cache-control directives are used to ensure the cache does not expose the data.
5	Cross-site request forgery	Issue	
6	Session timeout	Pass	The software has to revalidate each time session timeout. And it clears all cookies from the browser.

Table 16: Result of session management

A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed using JavaScript. If a malicious script can be run on this page, the cookie will be accessible and transmitted to another site. If this is a session cookie, then session hijacking may be possible.

No Anti-CSRF tokens were found in an HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent to act like the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a website has for a user. By contrast, cross-site scripting (XSS) exploits a user's trust in a website. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. CSRF attacks are effective in several situations, such as when a victim has an active session, been authenticated through HTTP and on the same local network as the target site. CSRF has primarily been used to act as a target site using the victim's privileges, but

recent techniques have been discovered to disclose information by gaining access to the response.

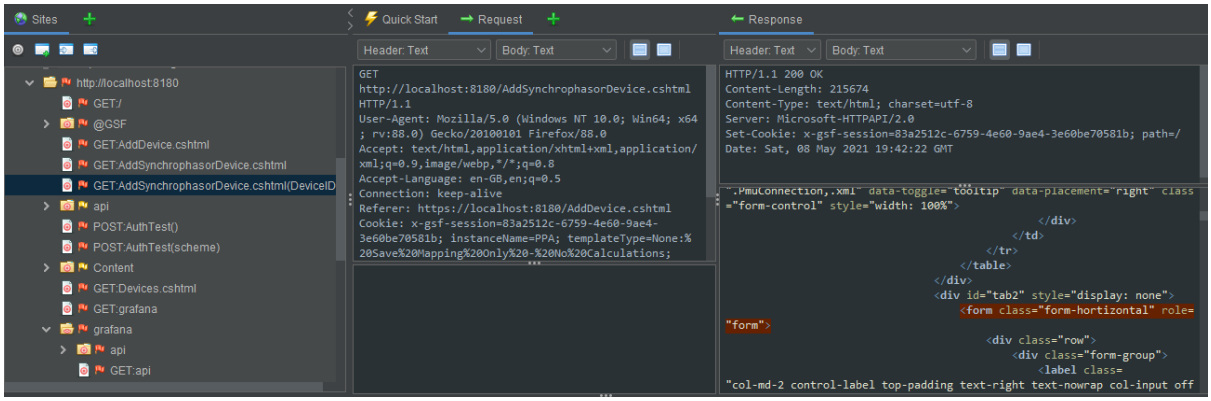


Figure 10: Evidence of cross-site request forgery

6.4.7 Input validation

No	Test name	Pass/Fail/Issue	Actual Result
1	Reflected cross site scripting	Pass	The OpenHistorian web interface should not allow data entry scripting through the Url to be executed.
2	Stored cross-site scripting	Pass	The OpenHistorian should not store user input in the cache or any form that an attacker could access.
3	HTTP parameter pollution	Issue	Impossible to tell from the client side as there is no feedback on the login box.
4	SQL injection	Fail	The vulnerability was explored by manipulating the parameters.
5	XML injection	Fail	The OpenHistorian is vulnerable to XML injection attacks.
6	SSI injection	Fail	The OpenHistorian login and add device interface as .shml

			link makes it vulnerable to SSI injection attacks.

Table 17: Result of input validation

SQL injection is possible. The page results were successfully manipulated using the Boolean conditions. The parameter value being modified was not stripped from the HTML output for the purposes of the comparison. Data was returned for the original parameter. The vulnerability was detected by successfully restricting the data originally returned by manipulating the parameter.

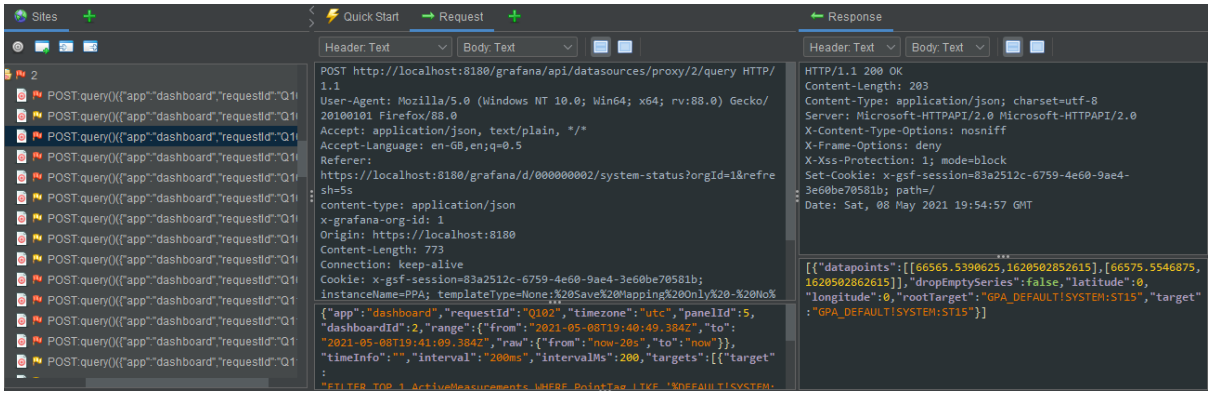


Figure 11:Evidence of SQL injection

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML or JavaScript but may also extend to VBScript, Java, Flash, or any other browser-supported technology. The type of XSS attack that it is most vulnerable to is Document Object Model (DOM)-based. DOM-based attacks require a user to either visit a specially crafted link laced with malicious code or visit a malicious web page containing a web form, which will mount the attack when posted to the vulnerable site. Using a malicious form will frequently take place when the vulnerable resource only accepts HTTP POST requests.

6.4.8 Error Handling

No	Test Name	Pass/Fail/Issue	Actual Result
1	Analysis of Error codes	Pass	The result shows 401(unauthorised) error codes. It means that the page cannot

			be accessed without a valid username and password.
2	Analysis of Stack traces	Pass	The OpenHistorian have stack trace implemented as part of error handling when invalid input or non-alphanumeric characters are entered.

Table 18: Result of error handling

6.4.9 Cryptography

No	Test Name	Pass/Fail/Issue	Actual Result	Comment
1	Testing for Weak SSL/TSL Ciphers, Insufficient Transport Layer Protection	Pass	The OpenHistorian has sufficient transport layer protection as it uses the TSL cipher.	When running the address through the ZAP application, it is forced to use a secure transport layer and run the application through google chrome.
2	Testing for Sensitive information sent via unencrypted channels	Pass	Sensitive information is encoded instead of encrypted by using encode64.	

Table 19: Result of cryptography

7. Evaluation

Looking at the results from the analysis, out of the 40 test cases from the nine sub-categories, 72.5% of them passed the test, 10% had issues and did not entirely fail, 17.5% failed the test. This result shows that the openHistorian is open to possible vulnerabilities within the context of the test cases. The software on its own is not as secure as you would expect of a software used in the energy sector, but it could be secured with extra security measures in place. For example, when using the OWSAP ZAP to analyse the software, it was forced to use a secure transport channel as it uses HTTP instead of HTTPS.

The issues raised in the test indicate areas that may be secure but have some comment or codes that would enable some form of attack on the system. For example, the cookie issues raised because there is no HTTPOnly attribute included in the cookie attribute was a low-risk alert that occurred more than a hundred times, which is why It was classified as an issue.

The failed test cases must be sorted to protect the system from attacks that can compromise the system. An attacker could use a misconfiguration of the server to access data that are saved on the historian; this will be familiar with the virus attack as mentioned in Section 4, sub-section 5.2.2.3.

From the test cases, there is a clear indication that the system's identity management and authorisation management are secure because only people in the company have been provided access with a unique username and password that can use the historian. Since the architecture model of the openHistorian and openPDC are the same, then it is safe to assume that this test could cover both openHistorian and openPDC. The selected sub-categories could include both software because the input data, output data, export, authentication, authorisation, and system identification are the same.

However, security test was done on one of the software used in the control centres by operators in the energy sector. These findings cannot be generalised to the entire WAMS system because it collaborates with different software that makes up the WAMS system and has to be tested together. To generalise it to the whole system, they have to be tested parallel to each other in a coordinated way from a safe location where there is better security.

7.1 Solutions to issues and failed test case:

Information gathering

I will recommend the removal of all comments that return information that may help an attacker and fix any underlying problems they referred. One principle of open-source

software is total transparency, but it will be good practice to obfuscate this information in the source codes for the user of the software.

Configuration and deploy management

Ensuring that sensitive data is not available in an unauthenticated manner (using IP address whitelisting, for instance) is a good way of securing the system. Also, configuring the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains or removing all CORS headers entirely allows the web browser to enforce the Same Origin Policy (SOP) more strictly.

Authentication & Cryptography

I recommend the use of HTTPS and a secure/robust authentication mechanism that does not transmit the user-id or password in an unencrypted fashion. In particular, avoid using the Basic Authentication mechanism since this trivial obfuscation mechanism is easily broken. As mentioned in Section [5.3.2], LDAP authentication will also be a good solution due to the significance of the software.

Session

I recommend the use of a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard. Ensure that your application is free of cross-site scripting issues because most CSRF defences can be bypassed using the attacker-controlled script.

Input validation

In general, I will recommend that all data on the server-side are type check. Do not trust client-side input, even if client-side validation is in place, because they could still be compromised. Escape all data received from the client, apply an 'allow list' of allowed characters, or a 'deny list' of disallowed characters in user input. Grant the minimum database access that is necessary for the application. Furthermore, use the principle of least privilege by using the least privileged database user possible.

Any data that will be output to another web page, especially any data received from external inputs, use the appropriate encoding on all non-alphanumeric characters. If available, use structured mechanisms that automatically enforce the separation between data and code. These mechanisms may provide the relevant quoting, encoding, and validation automatically instead of relying on the developer to provide this capability at every point where output is generated. To help mitigate XSS attacks against the user's session cookie, set the session

cookie to be HttpOnly. In browsers that support the HttpOnly feature, this attribute can prevent the user's session cookie from being accessible to malicious client-side scripts that use document.cookie.

8. Future Work

This project is still ongoing for me as I would like to chase it up in the future while working with a grid analytic company as part of my post-graduate studies. Continuing the project as a post-graduate will give me more time to cover the aspect of the research that was not possible due to the 3months time frame given as a study of this nature usually will take longer.

There are test cases that were not done due to the time frame of the project. In the future, I will complete the test with the complete selection of test cases that fit the requirement of the software. Test cases under sub-categories such as business logic and client-side testing that was not feasible at this point will be done using software that includes Nessus, Curl and Burp proxy.

Also, different vendors have a different product, and security varies in all of them. I can only access Open-Source software because commercially available software is expensive, and companies will not consent to run such analysis due to its impact on the company Image. I will have to conduct the same analysis on the openPDC, openXDA and other software dependent on each other to function.

Another aspect I would like to explore is Human Errors. Human errors can interfere with the generation, transmission and general adequacy of the WAMS data inside and outside the control rooms. Humans are the creators and users of this software, and sometimes there may be errors such as wrong configuration or input that may impact the grid systems.

Futhermore, he project's initial plan was to run the security test on both hardware and software because running the analysis on software alone will not give an accurate result of the entire WAMS system. I understand this is impossible outside of the environment and can only be achieved by working within the environment where these components are made. Companies have secure facilities where this test can be done without affecting the other systems in the grid network.

9. Conclusion

The use of open-source software is a welcoming development in the energy industry, not just applications but also API and GUI. This could also have security issues as the code for these applications is available online, and hackers can study and manipulate these codes to learn the vulnerabilities and how to use them. Aside from the hackers learning about the codes, security issues from open-source software that have not been tackled will also pose a security threat to the new application. Research shows that most commercial software includes some codes from open-source and free source software and these comes with vulnerabilities and issues that have not gone through a security audit.

There may be some issue from moving traditional interface to an open-source interface from this research. Although, traditional and open-source software interface gives the same function as the real-time visualization of the system. A combination of both would be a good solution to some extent. Open-source lowers the costs of some development that may be outsourced from a proprietary software company. However, compatibility becomes an issue because traditional API is mostly compatible with their software and not all security reasons.

The Blackbox approach was used for the assessment because of the unfamiliarity with the actual WAMS system and how it should work. For the industry that makes this software and knows what should be tested, a white box or grey box approach could be used and introduced early in the project during development than after production. Industries still use the waterfall approach, which they are familiarised with, to meet up demands and deal with defects after production. I propose using a combination of agile and waterfall method of testing for security defects because testing should be done at every stage during development till production; this will reduce the number of defects they have to deal with at the end of the release.

The result above in section [7] does not reflect a fully protected solution for software used in a crucial sector such as the energy sector. Even after years of studies and technology improvement in the sector, it still drives on implicit trust. This research shows the vulnerabilities of the system and the serious damage this could cause when these vulnerabilities are exploited by an attacker, causing havoc such as blackout, rendering usable data useless and severe disruptions of service.

This research explores a small fraction of the software application used in the grid system, the open-source phasor data concentrator and Historian. It does not explain the entire

system, so the result cannot be generalised to the existing grid system but can provide insight into the security of the software design architecture.

10. Reflection

What went well?

My research into the different components of the WAMS system went well. I was able to get information from good sources and journal publications, including those published by the IEEE. Researching has improved my knowledge in the subject areas that I am not familiar with and improve my skills by distinguishing between relevant and non-relevant data. I had to downsize the information that I was looking for to only those applicable to the project and the specific component that I will use for the analysis. An example is the data streams per seconds for PMUs, and the information was inconsistent with further research and the vendors were not mentioned. But then I understand that different vendors have different features and required stream time. I have removed this from my report because I did not want to send across false information, not knowing which vendor.

For the analysis, I learned how to use the OpenHistorian software and connect the historian and other software, including their dependencies. The OpenHistorian interface is simple to use and intuitive once you get to understand the software. Although it took me a while to understand the software, exploring the functionalities enabled me to understand how the system works before starting the analysis. Also, studying the OWASP web application assessment and going through the ZAP software tutorials on using the HUD, including types of attack, was a good way for me to know what tools or techniques are needed for the assessment.

Through doing this research, I have improved some skills, such as learning more advanced types of research and analysis. It is the first time I am running an assessment such as penetration testing for a project, and I could do that. It is a good skill to have when working in the security sector. My problem-solving skills have also improved because I was able to get around some of the issues, such as looking for alternative software to use. For example, the analysis was supposed to the done on the OpenPDC, not the Historian. Still, the OpenPDC doesn’t have a web interface and was not connecting with PMU to enable import or export data. I had to resolve using the Historian because it could stream straight from the PMU without the PDC and has a web interface.

What didn't go well and how It was resolved

While doing my research, I came across many challenges that I had to look for other alternatives. Firstly, the vulnerability test was supposed to be done with commercial software. I emailed the companies (SIEMENS and ABB) to request a trial of their software

stating the nature of my research, and I got a late reply from them which was two weeks to the submission of the project. The reply from the SIEMENS was not favourable because they only pointed me to a demo software on their platform. To request from General Electric, I had a meeting with a sales representative who confirmed it was not possible to get the software due to the nature of the research. While searching for alternatives, I found open-source software such as openPDC and openHistorian software as described in Section 6.2.2 above.

Secondly, in my initial plan, It was stated that I would create a simulation on how the software work and do the analysis based on that. However, I found out that it will be more time consuming and would not be feasible within the 3months period I have for this project. Using the alternative software that I found was the best solution for me, similar to what is used in the industry. Based on my research, I found out that most evaluation that has been done so far on the WAMS or PDC subject has been with the open-source software rather than the traditional software.

Thirdly, the data sample to test the device input and import data on the analysis was hard to come by. No one within the university could provide the data, and sample data are not available online on any platform. I researched how to acquire the data set needed for the analysis, then discovered that the sample data test from the PMU could be used to test the software output and test the connection of the PMU with the Historian or PDC.

Fourthly, the initial plan also stated research on the vulnerability of both software and hardware. However, due to the time constraints and schedule, I focused on the basics of the Historian software and the minimum information to meet the project's schedule. Further plans on how to achieve this goal are stated in the future work Section 0 above.

Future improvement

There are areas where improvement could have been done and will take that into account for next time, such as spending less time gathering relevant information for the project and compiling that information into the research and spending more time on the software analysis. I should have compiled the information as I get them to save time, but getting the proper structure, on the other hand, was an issue.

Additionally, proper time management would have given me extra time to cover more test cases than the 40 listed in the analysis and result. With time management, adding a cap time on task will be good practice for the future because it will help against unforeseen circumstances that may arise during the project that may impact it drastically.

Furthermore, I will carry out research of this nature in a controlled environment; it was not an issue because the attack is specific to the software alone. Although the analysis and attacks are made based on the application and its functions, using a virtual machine would have been better.

Overall, I am satisfied with what I can achieve within the timeframe of the research. I achieved the objective of the research, which is looking at the security and vulnerability of the WAMS systems, including the software used in the industry. However, there are still studies that would be interesting to chase in the future as stated in the section above.

11. Abbreviations

3DES – Triple Data Encryption Standard

AD – Active Directory

AES – Advanced Encryption Standard

ARP – Address Resolution Protocol

CSRF – Cross Site Request Forgery

DES – Data Encryption Standard

DHKE – Diffie Hellman Key Exchange

DoS – Denial-Of-Service

EMS – Energy Management System

GCM – Galois Counter Mode

GOOSE – Generic Object-Oriented Substation Event

GPS – Global Positioning System

IEEE – Institute of Electrical and Electronics Engineers

IoT – Internet of Things

IP – Internet Protocol

LDAP – Lightweight Directory Access Protocol

LEDs – Light-Emitting Diodes

MAC – Message Authentication Code

MD1-5 – Message Digest 1-5

OWASP – Open Web Application Security Project

PDC – Phasor Data Concentrators

PMU – Phasor Measurement Units

POC – Proof of concept

PPP – Point-to-Point Protocol

RDBMS - Relational Database Management System

RSA – Rivest-Shamir-Adleman

SCADA – Supervisory Control and Data Acquisition

SDH – Synchronous Digital Hierarchy

SE – State-Estimation

SHA – Secure Hash Function

SONET – Synchronous optical network

SV – Sampled Value

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

UTC – Universal Time Coordinated

VPN – Virtual Private Network

VAR – Volt-Ampere Reactive

WAMS – Wide Area Measurement Systems

WDM – Wavelength Division Multiplexing

WPA2 – Wi-Fi Protected Access 2

12. References

A. Phadke, J. T. M. A., 1983. A New Measurement Technique for Tracking Voltage Phasors, Local System Frequency, and Rate of Change of Frequenc. *IEEE Transactions on Power Apparatus and Systems,* PAS-102(5), pp. 1025-1038.

A. Phadke, T. B., 2018. Phasor measurement units, WAMS, and their applications in protection and control of power systems. *Journal of Modern Power Systems and Clean Energy,* 6(4), pp. 619-629.

ABB, n.d. *SCADA over IP-based LAN-WAN connections application.* [Online]
Available at:
https://library.e.abb.com/public/09e2909e92d2ce8ac1257863004e7da0/SCADA%20application%20flyer_small.pdf
[Accessed 30 03 2021].

Aditya Ashok, A. H. M. G., 2014. Cyber-physical security of Wide-Area Monitoring, Protection and Control in a smart grid environment. *Journal of Advanced Research,* 5(4), pp. 481- 489.

Cheng Xiaorong, W. Y. N. Y., 2012. *The Study on the Communication Network of Wide Area Measurement System in Electricity Grid.* [Online]
Available at: https://core.ac.uk/download/pdf/81994351.pdf
[Accessed 31 03 2021].

Chester, D., 2020. *How and Why Power Grid Cyberattacks are Becoming Terrorists' Go-To.* [Online]
Available at: https://energycentral.com/c/iu/how-and-why-power-grid-cyberattacks-are-becoming-terrorists-go
[Accessed 19 03 2021].

Crowell Moring, 2019. *NERC Issues "Lesson Learned" From a Cyberattack on an Electricity Control Center.* [Online]
Available at: https://www.crowell.com/NewsEvents/AlertsNewsletters/all/NERC-Issues-Lesson-Learned-From-a-Cyberattack-on-an-Electricity-Control-Center
[Accessed 20 03 2021].

GE, 2020. *Statkraft and GE join forces on new stability contract for GB grid, increasing renewables growth and supporting green recovery.* [Online]
Available at: https://www.ge.com/news/press-releases/statkraft-and-ge-join-forces-new-stability-contract-gb-grid-increasing-renewables
[Accessed 18 02 2021].

General Electric , 2019. *Effective Inertia.* [Online]
Available at: https://www.ge.com/digital/sites/default/files/download_assets/effective-inertia-datasheet-ge-grid-analytics.pdf
[Accessed 19 03 2021].

GPA, 2018. *Open Historian.* [Online]
Available at:
https://gridprotectionalliance.org/docs/products/openhistorian/OpenHistorian2018.pdf
[Accessed 02 04 2021].

IBM, 2020. *What is open source software?.* [Online]
Available at: https://www.ibm.com/topics/open-source
[Accessed 12 05 2021].

IEEE, 2019. IEEE Standard for Phasor Data Concentrators for Power Systems. *IEEE Std C37.247-2019,* pp. 1- 44.

Jian Ma, Y. M. Z. D., 2010. Phasor Measurement Unit and Its Application in Modern Power Systems. *Emerging Techniques in Power System Analysis,* pp. 147-184.

John Stewart, T. M. S. C. A. E. E., 2010. *Synchrophasor Security Practices.* [Online]
Available at: https://cms-cdn.selinc.com/assets/Literature/Publications/Technical%20Papers/6449_SynchrophasorSecurity_EE_20100913_Web.pdf?v=20150812-152045
[Accessed 10 04 2021].

John, 2012. *The Advantages and Disadvantages of Optical Fiber.* [Online]
Available at: https://community.fs.com/blog/the-advantages-and-disadvantages-of-optical-fibers.html
[Accessed 31 03 2021].

Luigi Coppolino, S. D. L. R., 2014. Exposing vulnerabilities in electric power grids: An experimental approach. *International Journal of Critical Infrastructure Protection,* 7(1), pp. 51-60.

Luis Fabiano dos Santos, G. A. M. L. S. F., n.d. *The Use of Synchrophasors for Wide Area Monitoring of Electrical Power Grids.* [Online]
Available at:
https://library.e.abb.com/public/9357d370a88948e3adcd1fdc1a216741/The%20Use%20of%20Synchrophasors%20for%20Wide%20Area%20Monitoring%20of%20Electrical%20Power%20Grids.pdf
[Accessed 19 03 2021].

Melwani, U., 2019. *Open-source vs Proprietary Software - Which One Is More Secure?.* [Online]
Available at: https://www.srijan.net/blog/open-source-vs-proprietary-software-which-one-is-more-secure
[Accessed 18 05 2021].

Metzler, S., 2020. *LDAP: Explained.* [Online]
Available at: https://www.securew2.com/blog/ldap-explained/
[Accessed 01 05 2021].

Monteagudo, J., 2020. *Power Grid Cybersecurity – where are we now?.* [Online]
Available at: https://cyberstartupobservatory.com/power-grid-cybersecurity-where-are-we-now/
[Accessed 01 03 2021].

O.Mohammed, T., 2017. Chapter 12 - Design and simulation issues for secure power networks as resilient smart grid infrastructure. *Science Direct,* pp. 245-342.

Paolo Castello, C. M. P. A. P. S. S., 2018. Active Phasor Data Concentrator performing adaptive management of latency. *Sustainable Energy, Grids and Networks,* Volume 16, pp. 270-277.

Peter Wall, K. H. D. W. S. C., 2016. *Deployment and demonstration of wide area monitoring system in power system of Great Britain.* [Online]
Available at: https://link.springer.com/article/10.1007/s40565-016-0218-3#citeas
[Accessed 18 02 2021].

Petters, J., 2020. *How to Use Wireshark: Comprehensive Tutorial + Tips.* [Online]
Available at: https://www.varonis.com/blog/how-to-use-wireshark/
[Accessed 19 05 2021].

Premkumar S., S. V., 2017. *impact of Denial of Service(DoS) attack in Smart Distribution Grid Communication Network.* [Online]
Available at: https://www.ripublication.com/ijaer17/ijaerv12n14_48.pdf
[Accessed 30 04 2021].

Rashad, M., 2020. *Saudi Aramco sees increase in attempted cyber attacks.* [Online]
Available at: https://www.reuters.com/article/saudi-aramco-security-idUSL8N2A6703
[Accessed 21 03 2021].

Steven M. Bellovin, W. R. C., 1994. Network Firewalls. *IEEE Communication Magazine,* 32(9), pp. 50-57.

Surender Kumar, M. K. S. D. K. J., 2015. *Cyber Security Threats in Synchrophasor System in WAMS.* [Online]
Available at:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.695.2420&rep=rep1&type=pdf
[Accessed 02 04 2021].

Thompson, A., 2021. *What is proprietary software and how does it work?.* [Online]
Available at: https://entrepreneurhandbook.co.uk/proprietary-software/
[Accessed 12 05 2021].

Wired Gorilla, 2021. *What is Open Source Software? Explained with Examples.* [Online]
Available at: https://wiredgorilla.com/what-is-open-source-software-explained-with-examples/
[Accessed 18 05 2021].