Cloud-based Autonomic Computing Framework for Securing SCADA Systems

Sajid Nazir
Glasgow Caledonian University, UK

Shushma Patel
London South Bank University, UK

Dilip Patel
London South Bank University, UK

ABSTRACT

This chapter proposes an autonomic computing security framework for protecting cloud-based SCADA systems against cyber threats. Autonomic computing paradigm is based on intelligent computing that can autonomously take actions under given conditions. These technologies have been successfully applied to many problem domains requiring autonomous operations. One such area of national interest is SCADA systems that monitor critical infrastructures such as transportation networks, large manufacturing, business and health facilities, power generation, and distribution networks. The SCADA systems have evolved from isolated systems into a complex, highly connected systems requiring constant availability. The migration of such systems from in-house to cloud infrastructures has gradually gained prominence. The deployments over cloud infrastructures have brought new cyber security threats, challenges and mitigation opportunities. SCADA deployment to cloud makes it imperative to adopt newer architectures and measures that can proactively and autonomously react to an impending threat.

KEYWORDS

Autonomic computing framework, cloud SCADA, cyber security, communications, critical infrastructures, hybrid cloud

INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) systems are used to monitor and control complex infrastructures of national importance such as transportation networks, power generation and manufacturing plants. SCADA systems can be visualised as a layered architecture, as shown in Figure 1. The field devices (sensors, etc.) at the lowest layer interact with the physical processes. At layer 2, the Programmable Logic Controllers (PLC), and Remote Terminal Units (RTUs) aggregate data values from the lower layer and communicate the commands and their responses through the communications network to the SCADA server and Human Machine Interface (HMI). The generation of commands at the top layer and collection of responses from the lowest layer results in the monitoring and control of the process. The applicability of SCADA systems has become widespread due to industrial automation, cost reduction and growth in global economies (Nazir *et al.*, 2017a).

Traditionally, SCADA systems were developed as closed systems with security being the overriding factor, and no Internet connectivity. Isolation and obscurity as a mechanism for protection is no longer an option for critical infrastructures (Mahoney and Gandhi, 2011) because in order to leverage efficiency and gain a competitive advantage, the systems are increasingly becoming connected to the Internet and cloud technologies. SCADA system security vulnerabilities were first highlighted by the Stuxnet attack (Karnouskos, 2011). Subsequently, there has been an increase in the frequency and sophistication, of the attacks as evidenced by Constantin (2014).

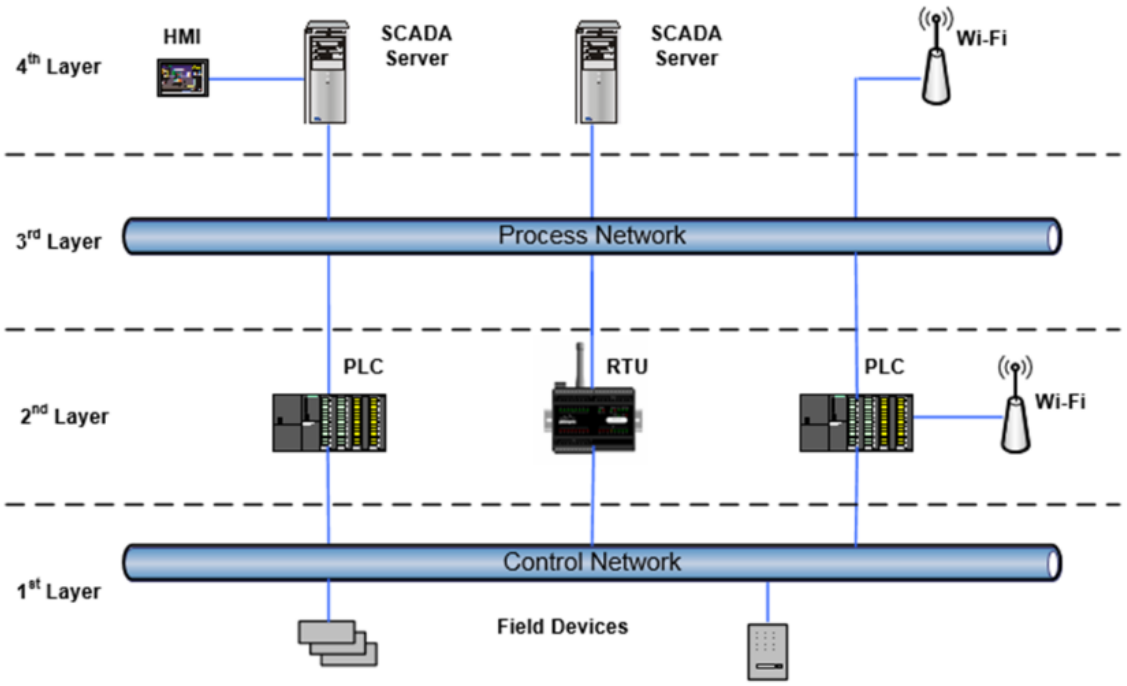

Fig. 1. Layered Architecture of a SCADA system.

The SCADA systems deployment to cloud can be configured in many ways, to suit the application. The SCADA application could be split over a hybrid cloud especially where the nature of the application dictates physical control over critical records such as in medical or finance applications. Also, it may be possible to deploy the complete application to the public cloud but a more likely cloud based deployment is where the sensors and control devices send the data over to the cloud, which can then be interpreted in real time (Larry, 2011). The HMI (Human Machine Interface) could be made available over the cloud for both the operator control and visualisation of the data and reporting. SCADA systems deployed on a cloud infrastructure could use the cloud providers' software and integrated tools for data analytics, reporting, dashboards and user interface. There are endless possibilities of integrating SCADA systems with cloud infrastructure and software (Sam IT Solutions) that can provide many benefits compared to a system hosted on a private cloud. The SCADA system can thus be accessible from anywhere in the world. However, such cloud deployments add many levels of complexities.

SCADA systems are getting more complex and it is difficult to develop effective defence strategies, as there is a lack of understanding of the complex interactions between the many system entities (Khadraoui and Feltus, 2015). The systems complexity and interactions go beyond the capability of system developers and integrators as a result of interconnectivity (Kephart and Chess, 2003). Thus, increasingly there is a lack of understanding of the whole system, which makes it very difficult to tune a system and to make decisions in case of changing requirements. This has led to a realization that conventional and inflexible security techniques will not help. What is needed is a new way of looking at the problem of cyber security that is robust, manageable and self-realising with a minimum requirement for a human operator to monitor systems to make intelligent decisions.

The complexity of developing and managing computing systems has become an important challenge facing the IT industry. The term 'Autonomic Computing' was first used by IBM in 2001 to combat the looming complexity crisis (Ganek and Corbi, 2003). The concept has been inspired by the human biological autonomic nervous system. It relates to intelligent computing platforms that are based on the disciplines of artificial intelligence, machine learning, and other innovative technologies. These technologies can be used to design systems that mimic the human brain to learn about their environment and can autonomously predict an impending anomalous situation. An autonomic system is self-healing, self-regulating, self-optimising and self-protecting (Ganek and Corbi, 2003). Therefore, the system should be able to protect itself against both malicious attacks and unintended mistakes by the operator. What is

proposed is an entirely new way of thinking about the problem where the system itself is intelligent and helps to maintain and extend its behaviour, with the use of autonomic computing (Kephart and Chess, 2003). Mallouhi (2011) describes a testbed for SCADA system security through autonomic software protection system using different attack scenarios.

The basic principles of autonomic computing are highly relevant for the protection of the increasingly complex cloud based SCADA systems because: (i) the boundaries between physical and virtual systems have been blurred through virtualisation. It is possible to host a cluster of machines in a virtual environment; (ii) even with hardware there are sufficient advances in other domains with self-healing materials; (iii) advances in machine learning, artificial intelligence and the knowledge base need to be capitalised for protection; (iv) the systems are highly interconnected and the distributed nature of the systems pose an exponential complexity.

Autonomic computing applications have been developed for use within complex SCADA systems. Greer and Rodriguez-Martinez (2012) and Amgai *et al.*, (2014) have discussed the application of autonomic computing for smart grids, as a solution to manage system complexities. Key components of a self-protecting SCADA system have been proposed and a survey of techniques provided for the realisation of such systems (Chen and Abdelwahed, 2014). Also, there are a number of dedicated research groups focusing research on the applicability of autonomic computing to cyber security (Autonomic Computing Lab; Cloud and Autonomic Computing Centre; Fortes *et al*., 2014). JADE (JADE) provides a framework for building autonomic management systems. A test bed was developed for modelling critical infrastructures for testing autonomic technologies (Autonomic Computing Lab; Cox, 2011).

We propose to apply the autonomic computing paradigm features to SCADA system security, in particular focussing on self-protecting cloud based SCADA systems. This chapter incorporates autonomic computing paradigm elements to cloud based SCADA systems to safeguard against the emerging cyber security challenges and threats facing cloud based SCADA applications.

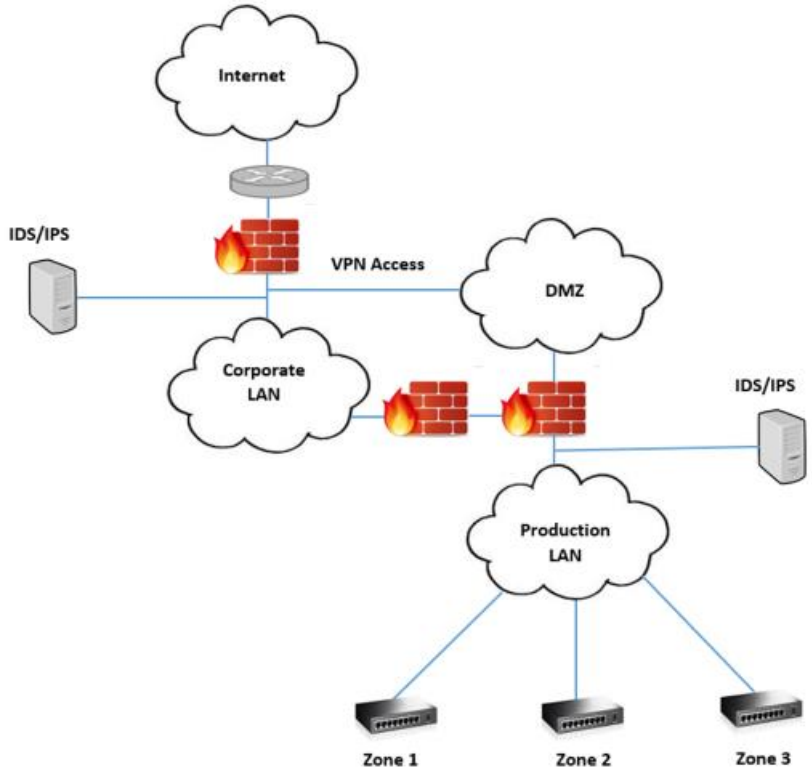

Fig. 2. Multiple pathways and Internet Connectivity to a Production System.

2. CLOUD BASED SCADA SYSTEMS

2.1 Vulnerabilities and Threat Landscape of SCADA Systems

SCADA systems were developed to be used as stand-alone systems, which by their very nature made it difficult for an outside attacker to exploit the system. However, the many benefits associated with interconnecting the system to the Internet have transformed the SCADA systems into a highly interconnected system accessible over the Internet (Fig 2) (Taveras, 2013; Nazir *et al*., 2017a), resulting in an increased exposure to threats. The system interactions are complex, opening new threat entry points as there are many third party libraries and hardware assembled with components from around the world, with exploitable threats such as backdoors, often unknown to the SCADA system vendor.

The systems developers design customized solutions to address a particular problem. The systems are fairly long term deployments because the controlled processes have large financial and industrial outlays. The criticality of maintaining the process means that the systems remain in continuous operation and have a range of redundancies incorporated to protect stalling the system for foreseeable problems.

SCADA communications protocols such as Modbus, Distributed Network Protocol (DNP), IEC 870-5 and T103 are described by the GE Communications Protocol. Most SCADA communications protocols have no encryption as they were designed when the SCADA systems existed only as stand-alone systems, rendering protocol authentication unnecessary. The Modbus protocol is one of the most common protocols for SCADA systems, which operate on simple request-response messaging (Al Baalbaki *et al*., 2013). The diversity of the protocols and their inoperability also creates obstacles to design secure communications (Sheldon *et al*., 2004). There are many publicly available tools that can capture network traffic wirelessly. Also the wireless devices that feed data to the SCADA system provide easy entry points for the intruder into the system because the end devices do not have adequate protection, due to very low power requirements.

SCADA application vendors design their software to be hosted on generic operating systems such as Windows and Linux variants for widespread deployments. However, this exposes SCADA applications to the same vulnerabilities as that of the operating system. The long operational lifetime of SCADA software means that the host operating system may be beyond technical support. Additional features being added to SCADA systems add further complexity and the systems become difficult to develop and maintain. Thus it becomes difficult to understand and restore systems to their operational state from a compromised state resulting from a cyber attack.

Cyber attack paradigms have progressed much beyond the traditional attack methodologies such as man-in-the-middle (MITM) and Denial of Service (DoS) attacks (Chen and Abdelwahed, 2014), and have become sophisticated to avoid detection. The traditional defence approaches are unable to cope with the latest attack methodologies, where for example, the system parameters are altered, and are individually legitimate, but on the whole result in system collapse. Correct operation of the system needs not only the correct commands but commands that are consistent with the prevailing state of the system. It is possible for an attacker to inject a valid sequence of commands that gradually take the system to an unstable condition. The systems also operate under very tight timing constraints and can have undesired consequences in case of timing violation. Even the smallest intrusions on the critical infrastructure controls, can result in malfunctions which have devastating ripple effects on the system as a whole. The system is susceptible to attacks with minor effects, which can alter the system behaviour in a negative manner, leading to a ripple effect that compromises the whole system. The SCADA system entities are generally spread over a large geographical area, thus necessitating synchronisation of information at each location.

The threat landscape is rapidly evolving (Khadraoui and Feltus, 2015) and has gained momentum because SCADA systems are now accessible over the Internet, and are no longer protected by obscurity as the communications protocols and characteristics are available to interested parties. Currently, both the state and non-state agents are trying to exploit the system's vulnerabilities. Cox (2011) discusses threat ontologies in detail.

In contrast to the attacks launched from outside, threats can also emanate from an innocent or deliberate mistake from an insider. Such attacks could cause more harm as they could be launched with some understanding about the systems operations.

Cloud infrastructure for data storage, management and processing is becoming popular and being adopted by an increasing number of organizations, however cyber security is still a predominant concern (Villalobos, 2014). One of the main issues for hosting SCADA in the cloud are the security concerns that arise due to resource sharing, communications over public networks, and maintaining data outside the organisation's control (Shahzad et al., 2013). The benefits of utilizing cloud resources for telecommunications applications and role of communications between the distributed data centres are described in (Vasilenko, 2016). This is important because data security is a major concern for cloud applications and is addressed by autonomic computing data monitoring security systems that monitor the data changes for potential threats (Zhang, 2018).

2.2 Threats and Vulnerabilities of Cloud based Systems

2.2.1 Closed network open to public

The major threat is that the data and applications deployed to the cloud are accessed over the Internet. Thus the closed network protection as in the case of an on-site fully controlled network is no longer available. Therefore the advantage of access from anywhere in the world also makes it susceptible to attacks from malicious users, such as distributed denial of service (DDoS) attack.

2.2.2 Shared Infrastructure

The cloud vendors do not provide any guarantees that hardware infrastructure would not be shared with other businesses. Thus whether the sharing companies or users can be trusted becomes an issue to be considered, as there might be competitors sharing the same physical server. The sharing of resources can result in many controversies for critical and real-time applications (Ahn and Cheng, 2013). This problem can be addressed by having a virtual private cloud. A virtual cloud is housed on a public cloud but it is fully segregated and, the company placing its data and applications on the virtual cloud is the only entity that has access to all the resources of the virtual cloud to itself.

2.2.3 Communications links

The communication links connecting a business to the cloud provider network are over the public cloud. This exposes the link to cyber attacks. Such issues can be resolved by having a virtual private network (VPN) to connect to the cloud infrastructure and use of encryption.

2.2.4 Virtual Machines

The virtual machines provided by the cloud vendor need to be fully protected and this could be the responsibility of the developer depending on the deployment model, if it is the infrastructure-as-a-service (IaaS).

2.2.5 Cloud Infrastructure unavailability

Another threat could be the cloud infrastructure going down and remaining unavailable for an extended period. Such an eventuality could have disastrous consequences for a business utilising any service and especially for SCADA applications. This would be more relevant where the regulations require holding the data in a particular region or data centre.

2.2.6 Insider Attack

The employees of the cloud provider have full access to the data and espionage or subversion for gains with a malicious intent cannot be ruled out. This also could be a result of an innocent mistake resulting in loss of access to the SCADA resources.

2.3 Key Benefits of Cloud Based Systems

Cloud infrastructures provide many benefits and savings to the businesses including:

- Cost reduction
- Availability
- Internet or browser access
- Elasticity and Scalability
- Virtualisation
- Security
- Resource Provisioning for storage, compute and networks
- Global Access
- Cost savings –Pay-per-use
- Reliability and Fault resolution.

2.4 SCADA System Deployment to Cloud

The migration of many business systems to the cloud, has taken place due to cost savings, however such large scale migration of data is not observed for real-time applications such as SCADA (Ahn and Cheng, 2013). The business critical nature of SCADA applications makes it a difficult decision for system developers, companies and the application users to consider deployment of SCADA to the cloud. However, confidence in cloud infrastructure is growing amongst businesses due to confidence in cloud vendors. Despite this, SCADA applications must mitigate against time critical communications constraints for adequate functioning of the system. The four possible ways of taking advantage of a cloud deployment for a SCADA application could be, SCADA HMI and Data Storage in Cloud; Complete SCADA system in Cloud; Hybrid cloud SCADA; and Public versus Virtual Private Cloud.

2.4.1 SCADA HMI and Data Storage in Cloud

The SCADA application runs on-site, whereas the information can be stored and displayed in the cloud with the control network providing data to the SCADA server. This is a common configuration and provides remote access and visualisation of data (Larry, 2011). A system based on Message Queue Telemetry Transport (MQTT) which is a publish/subscribe model protocol (Nazir and Kaleem, 2019) is proposed by Siemens SCADA WinCC OA (Siemens) for Industry 4.0 where IoT data can be populated through MQTT broker to the cloud. The data resides in the cloud database and has the resulting advantage:

- Data is available globally
- Can be easily replicated on to different server
- The interface can be connected to and from anywhere in the world
- Data analytics from the cloud vendor is available for visualization.

2.4.2 Complete SCADA system in Cloud

The SCADA application runs in the cloud but has remote connections to sensors and control devices. This configuration may suit a distributed SCADA system with PLC, RTU feeding data to SCADA servers in the cloud (Larry, 2011). A performance comparison of an entire SCADA system in the cloud environment for cost and security is described by Shahzad *et al.* (2013). The security threats considered were attacks on VM, inter VM attacks, and denial of service.

2.4.3 Hybrid cloud SCADA

A hybrid cloud is a composite cloud where some of the application resides on a private cloud and the remaining can be migrated to the public cloud as required depending on different contingencies, for example, in case of an overload or organisational requirements. A scenario could be that the more critical and confidential information is maintained in the private cloud, whereas the public cloud could be optionally used to absorb additional processing and storage. The migration of application and data to the public cloud due to an overload is known as cloud burst.

2.4.4 Public versus Virtual Private Cloud

A virtual private cloud is an infrastructure housed on public cloud but from the access perspective it is like a private cloud, that is, an organization getting the cloud services has dedicated and exclusive access to the cloud resources.

3. COGNITIVE INFORMATICS AND AUTONOMIC COMPUTING

3.1 Cognitive Informatics

Cognitive Informatics is a broad and multidisciplinary field of cognition and information sciences that investigates the human information processing and its applicability for computing applications. A comprehensive review of the cognitive informatics framework is provided by Wang (2007a) and it also describes the applications from the fields of computing and software engineering. It can have diverse goals based on the application field but the overriding aim is to improve the human-machine interaction through better decision-making. For example, object recognition and classification problems in computer vision are hard for computers but come naturally to humans, where a lot of progress has been made by mimicking the cognitive processes of the brain through Artificial Neural Networks (ANN). Similarly, the application of machine learning and agent based processing can help overcome the cyber threats facing SCADA systems.

Cognitive computing comprises of intelligent computing methodologies to build autonomous systems that mimic the inference mechanisms of the human brain (Wang, 2009). Thus a system can detect anomalies, events and entities in a system through pattern recognition and data mining. These pro-active and self-learning systems can provide an effective defence against cyber threats, as signature based approaches can only work against known threats.

The advances in the field of cognitive informatics have led to the development of cognitive computing. Computing can be classified at four levels of computation intelligence: data, information, knowledge, and intelligence (Wang *et al*., 2011c; 2015). Data and information processing have been well studied but the same is not been the case for the higher levels of computational intelligence. The trends in "Cognitive processes of the brain, particularly the perceptive cognitive processes, are the fundamental means for describing autonomic computing systems, such as robots, software agent systems, and distributed intelligent networks." (Wang, 2007b).

3.2 Autonomic Computing Paradigm

Autonomic Computing is one of the trans-disciplinary applications of Cognitive Informatics and an autonomic computing system using its intelligence can autonomously carry out its actions based on the set of events and goals (Wang, 2007a; 2007b). This contrasts with an imperative system whose behaviour is controlled by a stored program and is thus deterministic. The motivation for autonomic systems is to deal with the system complexity, which has reached an overwhelming proportion and is inspired by the human nervous system (Poslad, 2011).

The increase in system complexity and applications heterogeneity has made it difficult to process the information. This has necessitated the use of paradigms inspired by biological systems such as autonomic computing (Parashar and Hariri, 2005) that have a goal to realise systems and applications, which operate autonomously based on high level rules to meet the system mission. It differs from Artificial Intelligence (AI) in that unlike those systems the humans may take the ultimate decision.

As the size and complexity of an application grows so should the software to control it, to become more flexible and dynamic, so as to be self-managed (Kramer and Magee, 2007). Designing such systems is the real challenge. The basic idea of the Autonomic Computing paradigm is that the system should be sufficiently intelligent to enable it to develop and maintain itself in an optimised state. The research challenges of autonomic computing are described by Kephart (2005). The human body's feedback and control mechanisms (Kephart and Chess, 2003; Parashar and Hariri, 2005) have formed the basis of general systems theory and holism for the development and management of computer based systems. The autonomic computing paradigm mimics the human autonomic nervous system. The ability to self-manage SCADA system security threats by developing learning systems that recognise vulnerabilities will be

hugely advantageous. The agents and software services will form a part of the systems, gathering data and monitoring systems continuously (Yang *et al*., 2012).

3.3 Autonomic Computing Features

Autonomic computing can be developed using different technologies; however an autonomic system must demonstrate the following four main features: self-configuring; self-healing; self-optimising; and self-protecting (Ganek and Corbi, 2003):

1) Self-configuring: The system must be able to reconfigure its behaviour based on the changing system requirements. For example, to acquire more system resources, such as memory, in case the system is overburdened.

2) *Self-healing:* In response to detecting a compromised element in its configuration, or lack of resources, an autonomic system can respond by repairing itself to a good state. Based on this assessment the system should be able to, for example, isolate the system components that have been compromised and continue operation with the remaining elements and at the same time attempting to restore the compromised system elements.

3) Self-optimising: The system must be able to assess the current state of the system variables and be able to tune them resulting in an optimised tuned behaviour. This is crucial, especially in the case of complex systems where there are thousands of system parameters that can affect the system performance. For best results knowing or applying them all in a reasonable amount of time, is beyond the grasp of the human mind.

4) Self-protecting: The system should be aware of the normal system operation and be able to continuously monitor the current system state to determine when deviations occur. It can then take measures to contain the threat and to handle it.

Autonomic computing facilitates identifying factors that relate to a specific state – homeostasis. The development of a knowledge network helps to identify what 'homeostasis' is and when there is an imbalance, to understand the structure of the network, the defences, the threats and the attacks. The threats can be classified into two categories: 1) process-related: when valid credentials are used to make legitimate changes that can impact on industrial processes. These can also be due to an error in the input of incorrect values or an actual attack (Crawford, 2006) for example, by disgruntled employees; and 2) system-related: which are exploited via software or configuration vulnerabilities. For example, flaws in communication protocols, which are low level (layers 1 and 2) attacks on the SCADA architecture (Pidikiti *et al*., 2013). Developing a mechanism to mine logged data on process-related incidents is a potential solution to developing an autonomic computing approach for SCADA security. Identifying user activities and classifying the actions into signed-on or known user actions allows the analysis of threats as legitimate system commands by legitimate users, or by illegitimate users, to distinguish the threats into attacks or errors by developing a knowledge base (Hadžiosmanović *et al*., 2012). The open issues and challenges of autonomic computing covering the industrial and academic systems are provided by Salehie and Tahvildari (2005). All autonomic computing features are important but self-healing is important for cloud deployments, whereas self-configuration and self-optimization can be controlled through policies by the system users (Karakostas, 2014).

3.4 Autonomic Computing to Secure SCADA

Some recent technology adoptions and improvements in SCADA systems are promising to aid developing systems that can result in an autonomic SCADA system. System protection can be ensured through many techniques. The majority depend on the judgement of a human to provide safeguards for the system.

The latest trends and innovations, such as virtualisation, analytics and databases, and wireless communications, which must work together in close collaboration to achieve the system mission, have been applied to SCADA systems. The integrated framework can rightly be called systems of systems as

the complexity has increased beyond simple control and monitoring tasks, the fundamental basis of SCADA. This complexity implies that developing and maintaining such systems are reaching the limits of human cognition (Kephart and Chess, 2003; Huebscher and McCann, 2008).

System vendors have been cognisant of the prevailing cyber security environment and have added a number of features to the product offerings. These features include, for example multiplexing proxy, encryption and role based access to make the intruder's task difficult. Most SCADA vendors allow integration with relational databases in addition to the built-in historical databases that have some advantages (SQL: The Next Big Thing in SCADA). Relational databases such as Oracle have their own integrated analytics and data mining services that can make it easier to uncover any anomalous activity.

A review of machine learning techniques for outlier detection of different temporal data sets is provided in Gupta et al., (2014). Machine learning and data analytics techniques have revolutionised many application domains and have recently been introduced in SCADA applications software. Such native integration makes it easier for the SCADA developers to analyse the systems operations and identify impending attacks (Kirsch et al., 2014; Carcano et al., 2011). Machine learning and other such techniques can effectively analyse a system to detect anomalous activities. Such unsupervised anomaly detection schemes are more appropriate and efficient compared to human analysts (Jiang and Yasakethu, 2013) and other signature based approaches (Chen and Abdelwahed, 2014). The system can thus learn new approaches and provide defence against as yet unseen scenarios, as in the case of supervised learning approaches. The other techniques of interest could be based on agent based, artificial intelligence, and adaptive systems (Greer and Rodriguez-Martinez, 2012). The future of cyber security lies with exploiting such techniques that can not only autonomously assess the threats to the system security, but also contain and mitigate the threat from spreading. The operator alert can notify the human operator to initiate disaster recovery operations.

Virtualisation techniques provide many benefits that can advantageously be applied to support the autonomic computing paradigm. Virtualisation enables easy containment of an attack, restoring and disaster recovery, change and optimisation of system resources, etc., in a truly elastic manner.

A recent breakthrough in this direction is that of the Autonomic Computing paradigm. With Autonomic Computing, the ultimate control still rests with a human but the drudgery of data manipulation and threat assessment can be taken out of the loop. An autonomic system can automatically detect and fix anomalies, which help reduce human intervention (Ahad et al., 2015). An autonomic cloud manager autoJuJu is proposed by Karakostas (2014) that makes autonomous decisions for scaling up or down the number of VMs, showing how the proposed manager meets self-configuration and self-optimization by considering broad policies and rules for cloud deployment, running in a sense-plan-act loop.

4. AUTONOMIC ARCHITECTURE FOR SECURING CLOUD BASED SCADA SYSTEMS

In this section we provide a brief overview of the architectures proposed in the research literature and propose a framework that can be used to design SCADA systems that have built-in layered protection against both known and unknown threats. We also provide details of the autonomic computing SCADA architecture proposed by the authors (Nazir *et al.,* 2017b).

4.1 Autonomic SCADA Architecture

Some autonomic architectures have been proposed in the research literature. The IBM autonomic computing system comprises, monitoring, analysing, planning, executing and a knowledge base component (Ebbers *et al*., 2006) and was proposed for large-scale commercial systems. The architecture utilises Touchpoint Autonomic Managers that are self-configuring, self-healing, self-optimizing and self-protecting.

An introduction to autonomic computing together with the challenges and opportunities are presented in Parashar and Hariri (2005). They propose architecture for an autonomic element as a smallest functional unit and propose a manager for each autonomic element. Chen and Abdelwahed (2014) highlight the need

for better security for the SCADA system and present an autonomic security model comprising of risk assessment, early warning and prevention, intrusion detection, and intrusion response.

An autonomic computational intelligence based system suitable for big data was proposed for identifying cyber attacks on smart grids by Demertzis and Iliadis (2018). Real-time control and monitoring of smart grid through cloud framework was proposed in (Kulkarni, 2019) that helped early detection of grid failures, user verification, and prevention of grid failure from anywhere in the world. A cloud based autonomic framework for smart grid that analyses the user social media and sensor data for energy demand for household is proposed considering it as a big data problem (Qin, 2014). A multilevel user access control layer was proposed for cloud platform security hosting SCADA system accessible through services via service oriented architecture (Baker, 2015). A detailed survey of autonomic computing models and applications is provided by Huebscher and McCann (2008). An Autonomic Critical Infrastructure Protection (ACIP) system using anomaly detection and autonomic computing is proposed by Al-Baalbaaki and Al-Nashif (2013). The modular system has online monitoring, feature selection and correlation, multi-level behaviour analysis, visualisation, and adaptive learning. The evaluation of ACIP is described using Modbus traffic generator for the Modbus traces between a server and five different PLCs. The proposed system could detect and stop a variety of attacks on the Modbus protocol (Al-Baalbaaki and Al-Nashif, 2013). An autonomic computing architecture for a virtual private cloud to house real-time medical applications is proposed in (Ahn and Cheng 2013) by considering a distributed VM monitoring system for a virtual private cloud.

Autonomic computing for intrusion detection in cloud using big data through an intrusion response autonomic system using Hadoop for data organization and Map-Reduce for data extraction can provide self-awareness, self-configuration and self-healing in the cloud (Vieira, 2014). An intrusion detection system was proposed for self-healing in cloud infrastructure through trade-offs for performance and energy response where VMs can be replaced in case of attacks (Villalobos, 2014). It was shown that by incorporating knowledge of a physical model of the system it was possible to identify the attacks through changes in system behaviour (Cardenas *et al*., 2011). The detection of attacks was formulated as an anomaly-based intrusion detection. The results show that the response algorithm keeps the system in a safe state during an attack. Automatic response mechanisms were proposed on system state estimation. However, they caution that an automatic detection and response methodology might not be applicable for all processes in control systems.

A methodology for designing a smart critical architecture that protects communications, controls and computations using moving target defence and autonomic computing is proposed by Hariri *et al*. (2017) who also developed a Resilient Smart Critical Infrastructure Testbed (RSCIT). A general autonomic computing environment (Autonomia) was developed for control and management of smart critical infrastructures.

A survivable cyber-secure infrastructure (SCI) architecture is proposed by Sheldon *et al*. (2004) for a power grid and proposes a cognitive agent architecture combining agent-based and autonomic computing. Cognitive components are described as comprising of processes that are reactive, deliberate, or reflective.

4.2 Proposed Architecture

An autonomic system enables a SCADA system to optimise, configure and protect itself in case of changing the system state to a compromised one. The work to date for securing SCADA security focuses on discrete approaches.

The authors had proposed an integrated approach that combines, the discrete knowledge based approaches with cognitive approaches. The memory layer of the Layered Reference Model of the Brain (LRMB) (layer 2) reflects the knowledge base that captures the short term, long term and transient memories. This can be utilised to capture process- and systems-related threats. Memory can be defined as a set of subconscious cognitive processes that retain the external or internal information about various SCADA security events. The subconscious knowledge base is inherited from the range of events and threats identified, and the conscious subsystem, however, is acquired and flexible, based on the autonomic computing paradigm (Wang *et al.*, 2006a; Wang and Wang, 2006b).

In contrast to the architectures above, our proposed architecture combines three features to provide a threat-resilient SCADA framework: (i) virtualisation of computing and networking resources (ii) hierarchy of autonomic managers (AMs) to identify threats at different scales (iii) protection against false alarms. In the following sections we describe the autonomic manager element and the corresponding autonomic SCADA architecture.

4.2.1 Autonomic Manager

Virtualisation refers to creating a virtual rather than physical version of computer hardware, storage and networks. The advantages are that the computing resources can be elastically assigned as required and it is much easier to monitor the virtual machines. In case of a cyber attack, a clean instance can be easily launched and the compromised machine can be isolated for forensics. Also, disaster recovery and rollback can be performed easily. We propose hosting the SCADA system on a virtual platform. The advantages are that it can provide high availability through protection against hardware and software failures. Thus creating a broad generalised structure based on virtualisation wherein appropriate technologies can be selected to best suit an application within the given framework.

We propose the concept of hierarchical autonomic managers that can scale protection from a small to a wide area. A domain autonomic manager, AM_d performs real-time analysis of their limited domain (database, communications, etc.,) at a small scale. These domain-based analyses are then aggregated at the local system level, AM_l for identification of anomalies to counter the threats locally. This relieves the central autonomic manager, AM_c to take more holistic actions. Thus, a central autonomic manager can perform an analysis of system wide aggregated analysis to counter system wide variations to identify possible threats.

Thus, the inference of AM is based on the intelligent aggregation of the inferences of its lower level AM.

$$Inferences\ AM_c = \sum_{i=1}^{N} InferencesAM_l$$

We argue that despite the current state-of-the-art in autonomic computing applications, such as, machine learning and neural networks applied to SCADA systems, the ultimate decision should lie with the human operator. This is due to the criticality of the SCADA applications that might jeopardise the safety and health of people, or compromise national security and infrastructures in case of false alarms. This of course, will vary from one application to another and a human decision-maker could be in the loop at some or all layers of AMs. The hierarchy of autonomic managers abstracts the information as it proceeds from low to high levels (domain to global) and can recommend actions to make it easier for a human operator to make a decision.

The structure and execution cycle of an AM is shown in Fig 3. It is planned and executed based on the given goals and rules. The execution starts at plan stage, followed by evaluate stage which could be monitoring or comparison, to determine a condition to be an anomaly or a progression towards one, inferring the threat, and reporting the inference to its higher AM. The knowledge base is analogous to the human nervous system storing structured and unstructured information used by the autonomic manager during its operation.

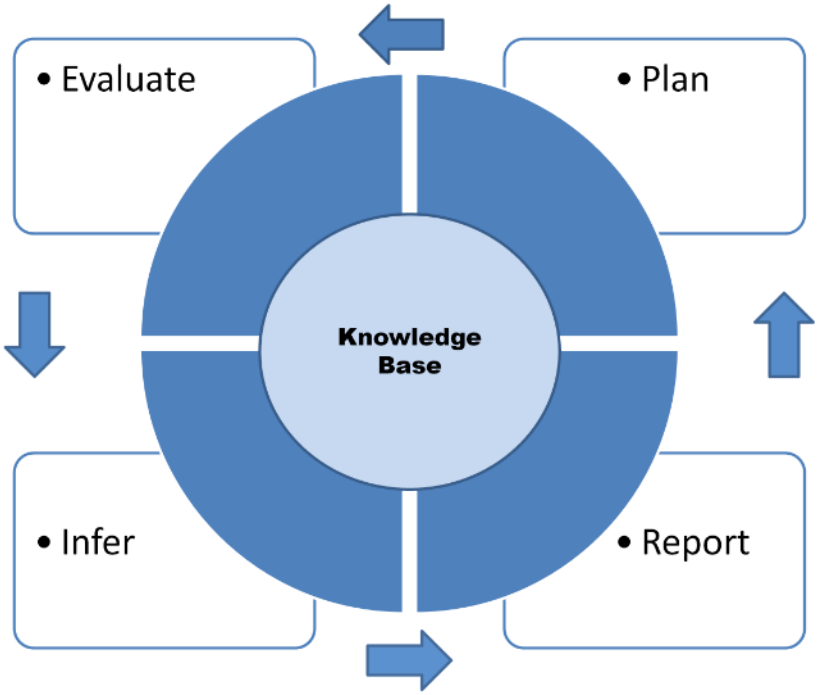

Fig 3. Structure and execution cycle of an autonomic manager.

The autonomic manager, as shown in Fig 3, can be used at various security layers of the system. The hierarchy helps to place the inferences at appropriate levels and the intelligence can travel up to the highest layer, that is, the central AM.

4.2.2 Autonomic SCADA Architecture for a Non-cloud Deployment

A SCADA system can have a large geographical spread, exposing it to exploitation at many locations, therefore necessitating an autonomic manager at each location that can monitor the security in the local areas and coordinate the efforts through the central manager. A simplified SCADA system architecture is shown in Figure 4. At the heart of the system is a central autonomic manager that can enforce the broad threat mitigation and containment policies in the managed system, as defined by the system administrator. The knowledge base provides the various historical system models that are continuously modified to the current state and are analysed to check conformance. The local autonomic managers continually observe the system state and act promptly in case of identified security threats to the local system.

Our proposed architecture provides a broad generalised structure based on virtualisation wherein appropriate technologies can be selected to best suit an application within the given framework. The identification of anomalies at an area level helps to counter the threats locally, relieving the central autonomic manager to take more holistic actions to counter system wide threats.

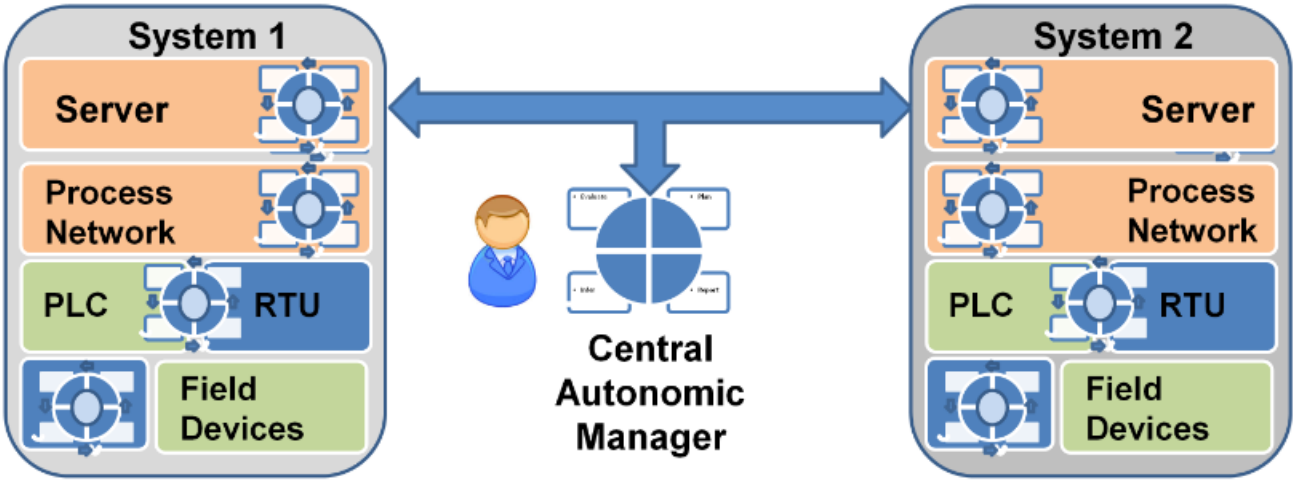

Fig 4. Proposed Architecture for an autonomic SCADA system.

The autonomic computing system incorporated to monitor a SCADA system may generate false alarms and therefore it may be necessary, based on the application domain, for a human operator to make a final decision based on the evidence.

It is also pertinent to point out here that the autonomic manager itself can be the target of a cyber attack. Such exploitation can be avoided through redundant deployments of managers and an integrated approach as proposed.

4.2.3 Cloud based Autonomic Framework for SCADA Security

The aim is to propose a generalised framework that can be helpful of deployment of autonomic SCADA applications to the cloud. The objective is to protect the SCADA deployment against cyber attacks and in doing so continuously evaluate any imminent breaches. In case of breaches the steps are taken to contain the breach and to restore the system to its pre-compromise state. The framework makes use of machine learning and data analytic techniques to quickly identify an attack vector.

The virtual machines provide many benefits to house a SCADA system, in case of any compromise it can quickly be restored to safe state. Such redundant applications can be run so that in case of a compromise of one, the other can continue to operate. Virtualization also makes it possible for a running application to migrate to another server, a phenomenon known as 'Live Migration'. This could be very helpful if the server itself is compromised and there are dangers of process disruption.

This work proposes the application of autonomic computing paradigm to cloud based deployments of SCADA applications. The timing constraints and real time nature of information dictates a hybrid cloud where the more critical elements could be deployed off-site in a private cloud and less critical elements could be housed in the public cloud. The framework proposed is generic enough so that it can be made use of for different types of SCADA applications and deployment domains.

A high-level cloud-based autonomic computing framework to secure SCADA systems is proposed as shown in Figure 5. We propose the hosting of SCADA systems on a hybrid cloud where the SCADA server is placed on an on-premises private cloud and the Human Machine Interface (HMI) on the public cloud. The control devices such as PLC provide data to the SCADA server running on the on-premises private cloud.

A strong protection against cyber threats is possible by integrating the security features of the selected cloud and SCADA system with the hosting organisation's customised security features. Public clouds provide many features for security and disaster recovery such as two-factor authentication, access keys, data backup, and security groups (acts as firewall). For example, a Denial of Service (DoS) attack can be detected and contained through the use of cloud load balancer. The SCADA application provides a heartbeat mechanism used for continuous monitoring of different software and hardware entities in the system, for failure through a timeout mechanism. An absence of heartbeat signal could result in an

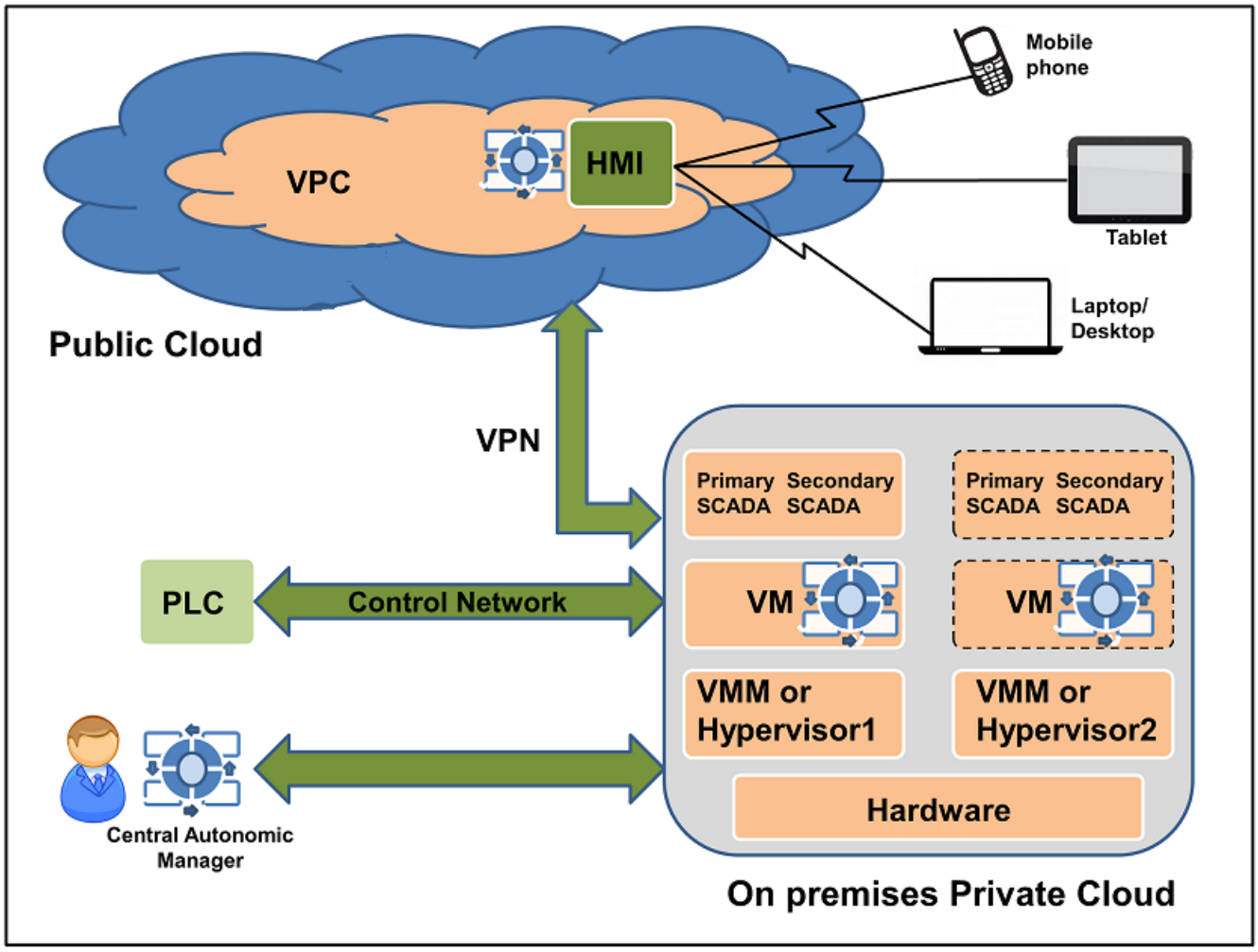

Fig 5. Proposed cloud-based autonomic computing architecture for SCADA system.

automatic invocation of redundant SCADA servers in cases of detection of problems with the primary server or could also be flagged for the attention of the human operator.

As shown in Fig. 5, Virtual Machine Monitors (VMM) or hypervisors on multiple host servers (hardware) can counteract a cyber threat by restoring a VM from a snapshot or backup. The active or primary SCADA application is run together with a secondary or standby server that has same state as primary server and can take over in case of problems with the primary server. The public cloud environment inherently provides redundancy and failover but with the proposed hybrid deployment some failover mechanisms would have to be provisioned in the on-premises cloud. For example, a redundant Internet connection could safeguard against the only connection going down (Larry, 2011).

A VPN is used to connect the SCADA database in the on-premises cloud with the HMI on the public cloud. For security and to avoid sharing of cloud resources with other users, we propose the use of a virtual private cloud (VPC) within the public cloud. HMI could be accessed over the public Internet through Internet connected devices. Access control rules based on Internet Protocol (IP) addresses, location or protocols can be used for added security.

The autonomous operation of the SCADA server in the cloud infrastructure is based on a set of rules and policies that are continuously controlled and monitored through the autonomic manager for cyber security. The autonomic manager through machine learning techniques can detect once a value is beyond a specified threshold or can detect outliers by considering many attribute values together. Based on the

specified rules and policies a failure detected by an autonomic manager could automatically contain that attack by activating a redundant system, or automatically prevent an attack by updating the encryption algorithms.

The autonomic manager is proposed for use with all devices and software that are processing or communicating the data, and reporting detected threats to the central AM for decision making. The autonomic cloud based SCADA infrastructure protects itself autonomously against security threats. The threat detection is basically a binary classification process, but where decision is not robust (probability of threat) it can be referred to the human operator for review or decision-making. Thus, the proposed architecture also results in an added benefit of reducing direct operators' involvement to continuously analyse the cyber threats.

5. CONCLUSION

This chapter has proposed a cloud based autonomic framework for protecting SCADA systems deployed to the clouds. We propose the concept of hierarchical autonomic managers that help to extract, aggregate and refine intelligent inferences for ultimate decision making by a human operator. The proposed framework is generic and can be suitably applied across a range of real-world SCADA applications. The importance of the need to have such frameworks has arisen due to a gradual recognition that the cloud deployments provide enormous benefits to SCADA systems. The evolving cyber threat landscape dictates changes to cyber defence approaches for the protection of SCADA systems in the cloud. Unlike the traditional defence approaches where the response is governed by tailoring and monitoring according to threats, the concept of autonomic computing provides an advantage, as the systems are self-protecting.

Thus, the autonomic computing paradigm is very promising to develop SCADA system cyber security architectures that facilitate proactive threat mitigation methodologies, without an active intervention by a human operator. The autonomous nature enables flexible and scalable solutions across a wide range of SCADA system architectures and applications. In future, the end deployments of cloud such as fog computing and cloudlets can be explored that make it possible to migrate the complete SCADA application to the cloud without compromising the strict timing constraints of the SCADA applications. Such deployments would also be aided by the widespread deployments of 5G networks that provide better reliability and high data rates for the wireless connection.

REFERENCES

Autonomic Computing Lab. http://acl.ece.arizona.edu/research.html [Accessed 20 October 2019].

Ahad, R., Chan, E., and Santos, A., 2015. Toward Autonomic Cloud: Automatic Anomaly Detection and Resolution. *2015 International Conference on Cloud and Autonomic Computing*, Boston, MA, 2015, pp. 200-203.

Ahn Y. W., and Cheng A. M. K., 2013. Autonomic Computing Architecture for Real-Time Medical Application Running on Virtual Private Cloud Infrastructures, *Newsletter ACM SIGBED Review - Special Issue on the Work-in-Progress (WiP) session of the 33rd IEEE Real-Time Systems Symposium (RTSS'12)* Homepage archive, vol.10, issue 2, July 2013, Pages 15-15.

Al Baalbaki, B., Al-Nashif, Y., Hariri, S. and Kelly, D., 2013. Autonomic critical infrastructure protection (acip) system. In *Computer Systems and Applications (AICCSA), 2013 ACS International Conference on* (pp. 1-4). IEEE.

Amgai, R., Shi, J., and Abdelwahed, S., 2014. An integrated lookahead control-based adaptive supervisory framework for autonomic power system applications, *Electrical Power and Energy Systems* 63 (2014) 824–835.

Baker T., Mackay M., Shaheed, A., and Aldawsari B., 2015. Security-Oriented Cloud Platform for SOA-Based SCADA. *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*.

Carcano, A., Coletta, A., Guglielmi, M., Masera, M., Fovino, I.N. and Trombetta, A., 2011. A multidimensional critical state analysis for detecting intrusions in SCADA systems. *IEEE Transactions on Industrial Informatics*, *7*(2), pp.179-186.

Cárdenas, A.A., Amin, S., Lin, Z.S., Huang, Y.L., Huang, C.Y. and Sastry, S., 2011. Attacks against process control systems: risk assessment, detection, and response. In *Proceedings of the 6th ACM symposium on information, computer and communications security* (pp. 355-366). ACM.

Chen, Q. and Abdelwahed, S., 2014. Towards realizing self-protecting SCADA systems. In *Proceedings of the 9th Annual Cyber and Information Security Research Conference* (pp. 105-108). ACM.

Cloud and Autonomic Computing Centre: https://sites.google.com/nsfcac.org/home [Accessed 20 October 2019].

Constantin L., 2014. "New Havex malware variants target industrial control system and SCADA users," *PC World*, Jun 2014.

Cox, D.P., 2011. The application of autonomic computing for the protection of industrial control systems. The University of Arizona.

Crawford, M., 2006. Utility hack led to security overhaul. Computerworld, 2006, pp.1-2.

Demertzis, K., and Iliadis, L., 2018. A Computational Intelligence System Identifying Cyber-Attacks on Smart Energy Grids, In book: Modern Discrete Mathematics and Analysis.

Ebbers M., Byrne F., Adrados P. G., Martin R., and Veilleux J., 2006. Autonomic Computing (Chapter 8) IBM Introduction to the New Mainframe: Large-Scale Commercial Computing. ftp://public.dhe.ibm.com/systems/z/z_coursematerials/lscc/Large_Scale_Commercial_Computing_Student.pdf [Accessed 20 October 2019].

Fortes J., Parashar M., Hariri S., Banicescu I., 2014. Center for Cloud and Autonomic Computing (CAC). Compendium of Industry-Nominated NSF I/UCRC Technological Breakthroughs.

Ganek, A.G. and Corbi, T.A., 2003. The dawning of the autonomic computing era. IBM systems Journal, 42(1), pp.5-18.

GE, Communications Protocol, https://www.gegridsolutions.com/app/DownloadFile.aspx?prod=gesapm&type=8&file=7 [Accessed 20 October 2019].

Greer, M. and Rodriguez-Martinez, M., 2012. Autonomic computing drives innovation of energy smart grids. *Procedia Computer Science*, *12*, pp.314-319.

Gupta, M., Aggarwal, C. C., and Han, J., 2014. Outlier Detection for Temporal Data: A Survey. *IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 9, Sep 2014.*

Hadžiosmanović, D., Bolzoni, D. and Hartel, P.H., 2012. A log mining approach for process monitoring in SCADA. *International Journal of Information Security*, pp.1-21.

Hariri S., Pacheco J., Tunc C., and Al-Nashif Y., 2017. A Methodolgy for Designing Resilient and Smart Critical Infrastructures. https://pdfs.semanticscholar.org/6c64/f2300b2cef6731e87d006a8db494cb1bb6be.pdf [Accessed 20 October 2019].

Huebscher, M.C. and McCann, J.A., 2008. A survey of autonomic computing—degrees, models, and applications. *ACM Computing Surveys (CSUR)*, *40*(3), p.7.

JADE - A framework for developing autonomic administration software. http://raweb.inria.fr/rapportsactivite/RA2009/sardes/uid40.html [Accessed 20 October 2019].

Jiang, J. and Yasakethu, L., 2013. Anomaly detection via one class svm for protection of scada systems. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013 International Conference on* (pp. 82-88). IEEE.

Karakostas, B., 2014. Towards Autonomic Cloud Configuration and Deployment Environments. *2014. International Conference on Cloud and Autonomic Computing*, London, 2014, pp. 93-96.

Karnouskos, S., 2011, November. Stuxnet worm impact on industrial cyber-physical system security. In *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society* (pp. 4490-4494). IEEE.

Kephart, J.O. and Chess, D.M., 2003. The vision of autonomic computing. *Computer*, *36*(1), pp.41-50.

Kephart, J. O., 2005. Research Challenges of Autonomic Computing. *ICSE '05 Proceedings of the 27th international conference on Software engineering*, Pages 15-22.

Kirsch, J., Goose, S., Amir, Y., Wei, D. and Skare, P., 2014. Survivable SCADA via intrusion-tolerant replication. *IEEE Transactions on Smart Grid*, *5*(1), pp.60-70.

Khadraoui, D. and Feltus, C., 2015. Designing Security Policies for Complex SCADA Systems Protection. *INFOCOMP 2015*, p.66.

Kramer, J., and Magee, J., 2007. Self-Managed Systems: an Architectural Challenge. *Future of Software Engineering (FOSE '07)*, Minneapolis, MN, 2007, pp. 259-268.

Kulkarni, N., Lalitha, S. V. N. L., Deokar, S. A., 2019. Real time control and monitoring of grid power systems using cloud computing. *International Journal of Electrical and Computer Engineering*, Apr 2019, vol. 9(2), pp.941-949.

Larry, C., Cloud Computing for SCADA. 2011. *Control Engineering*, December 2011. Available online: https://search.proquest.com/docview/1023312549?accountid=15977

Mahoney, W. and Gandhi, R.A., 2011. An integrated framework for control system simulation and regulatory compliance monitoring. *International Journal of Critical Infrastructure Protection*, *4*(1), pp.41-53.

Mallouhi, M, Al-Nashif, Y., Cox, D., Chadaga, T., and Hariri, S., 2011. A Testbed for Analyzing Security of SCADA Control Systems (TASSCS), 2011. *ISGT 2011*, Anaheim, CA, 2011, pp. 1-7.

Nazir, S., Patel, S. and Patel, D., 2017a. Assessing and augmenting SCADA cyber security: A survey of techniques. *Computers & Security*, *70*, pp.436-454.

Nazir, S., Patel, S., and Patel, D., 2017b. Autonomic Computing Architecture for SCADA Cyber Security. *IGI International Journal of Cognitive Informatics and Natural Intelligence, Nov 2017.*

Nazir, S., Kaleem, M., 2019. Reliable Image Notifications for Smart Home Security with MQTT *International Conference on Information Science and Communication Technology (ICISCT)*

Parashar M. and Hariri S., 2005. Autonomic Computing: An Overview," in Unconventional Programming Paradigms. *Lecture Notes in Computer Science*, vol 3566. Springer, Berlin, Heidelberg.

Poslad, S., 2011. Ubiquitous computing: smart devices, environments and interactions. John Wiley & Sons.

Qin, Y. B., Housell, J., Rodero, I., 2014. Cloud-based Data Analytics Framework for

Autonomic SmartGrid Management. *International Conference on Cloud and Autonomic Computing*, September 2014, pp.97-100

Salehie, M., and Tahvildari, L., 2005. Autonomic Computing: Emerging Trends and Open Problems, *ACM SIGSOFT Software Engineering Notes*, Jan 2005.

Sam IT Solutions: https://www.samitsolutions.com/leveraging-the-cloud-for-wincc-oa/ [Accessed 20 October 2019].

Shahzad, A., Musa, S., Abourujilah, A., and Irfan, M., 2013. *International Conference on Advanced Computer Science Applications and Technologies.*

Sheldon, F., Potok, T., Langston, M., Krings, A. and Oman, P., 2004. Autonomic approach to survivable cyber-secure infrastructures. In *IEEE Int. Conf. on Web Services (ICWS 2004), California, USA*.

Siemens - https://cache.industry.siemens.com/dl/files/955/109760955/att_967236/v1/109760955_WinCCCloudConnector_en.pdf

SQL: The Next Big Thing in SCADA, White Paper, Inductive Automation, 2012. https://www.automation.com/pdf_articles/inductive_automation/WhitePaper_SQL_The_Next_Big_Thing_in_SCADA.pdf [Accessed 20 October 2019].

Taveras, P., 2013. SCADA live forensics: real time data acquisition process to detect, prevent or evaluate critical situations. *European Scientific Journal, ESJ, 9*(21).

Vasilenko, O., 2016. Dynamic Allocation of Cloud Resources for Telecommunication Applications. *2016 International Conference on Cloud and Autonomic Computing (ICCAC)*, Augsburg, 2016, pp. 119-122.

Vieira, K. M. M., Schubert, F., Geronimo, G. A., Mendes, R, Westphall, C. B., 2014. Autonomic Intrusion Detection System in Cloud Computing with Big Data, *International Conference on Security and Management (SAM 2014)*.

Villalobos, J. J., Rodero, I., Parashar, M., 2014. Energy-Aware Autonomic Framework for Cloud Protection and Self-Healing. *2014 International Conference on Cloud and Autonomic Computing*, London, 2014, pp. 3-4.

Wang Y., Wang Y., Patel S.,and Patel D., 2006a. A layered reference model of the brain (LRMB),". *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 36(2), 124-133.

Wang Y. and Wang Y., 2006b. Cognitive informatics models of the brain" *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 36(2)*, 203-207.

Wang Y., 2007a. The Theoretical Framework of Cognitive Informatics. *Int'l J. of Cognitive Informatics and Natural Intelligence*, 1(1), 1-27.

Wang Y., 2007b. Towards Theoretical Foundations of Autonomic Computing. *Int'l J. of Cognitive Informatics and Natural Intelligence*, 1(3), 1-16.

Wang Y., 2009. Cognitive Computing and machinable thought. In *2009 8th IEEE International Conference on Cognitive Informatics*, Kowloon, Hong Kong, pp. 6-8.

Wang Y., Widrow B., Zhang B., Kinser W., Sugawara K., Sun, F., Lu J., Lu J., Weise T., and Zhang D., 2011a. Perspectives on the Field of Cognitive Informatics and its Future Development. *Int'l J. of Cognitive Informatics and Natural Intelligence*, 5(1), 1-17.

Wang Y., 2011b. The Cognitive Processes of Formal Inferences. *Int'l J. of Cognitive Informatics and Natural Intelligence*, 1(4), 75-86.

Wang Y., Berwick R. C., Haykin S., Pedrycz W., Kinser W., Baciu G., Zhang D., Bhavsar V. C., and Gavrilova M., 2011c. Cognitive Informatics and Cognitive Computing in Year 10 and Beyond. *Int'l J. of Cognitive Informatics and Natural Intelligence*, 5(4), 1-21.

Wang Y., Rolls E. T., Howard N., Raskin V., Kinsner W., Murtagh F., Bhavsar V. C., Patel S., Patel D., and Shell D. F., 2015. Cognitive Informatics and Computational Intelligence: From Information Revolution to Intelligence Revolution, *International Journal of Software Science and Computational Intelligence (IJSSCI)*, 7(2), 50-69.

Yang L., Cao X., Gen X., and Zhang J., 2012. A Knowledge expression method of SCADA network attack and defence based on factor state space. *Journal of Theoretical and Applied Information Technology*, 46(2).

Zhang, J., Wu Q., Zheng R., Zhu J., Zhang M., and Liu R., 2018. A Security Monitoring Method Based on Autonomic Computing for the Cloud Platform. *Hindawi Journal of Electrical and Computer Engineering, vol. 2018, Article ID 8309450, 9 pages.*