IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

Secure SCADA Firewall Autmation and Implication for Best Practices

[1]Sai Pradeep Kumar M., [2]Haritha D.

[1]Cyber Security, University College of Engineering (A), Jawaharlal Nehru Technological University, Kakinada, India.
[2]Computer Science and Engineering, University College of Engineering (A), Jawaharlal Nehru Technological University, Kakinada, India.

Corresponding Author: pradeepmycharla@protonmail.com, Tel.: +91-9494669993

Available online at: www.ijcseonline.org

Accepted: 18/July/2018, Published: 31/July/2018

Abstract**—** SUPERVISORY Control and Data Acquisition (SCADA) networks control the distributed assets of many industrial systems. Power generation, water distribution and factory automation are just a few examples that illustrate the critical nature of these networks. SCADA devices are built for reliability, but often lack built-in security features to guard them from cyber-attacks. Consequently, these devices depend on firewalls for protection. Hence, firewalls are integral to SCADA networks control the distributed assets of many industrial systems. Power generation, water distribution and factory automation are just a few examples that illustrate the critical nature of these networks. SCADA devices are built for reliability, but often lack built-in security features to guard them from cyber-attacks. Consequently, these devices depend on firewalls for protection. Hence, firewalls are integral to the safe and reliable operation of SCADA networks. Firewall configuration is an important activity for any modern day business. It is particularly a critical task for the SCADA networks that control power stations, water distribution, factory automation, etc. Lack of automation tools to assist with this critical task has resulted in un-optimized, error prone configurations that expose these networks to cyber-attacks. Automation can make designing firewall configurations more reliable and their deployment increasingly cost-effective. In order to increase the security in firewall we are providing extra automation that would help to detect the packet level conflicts such DoS.

Keywords**—** SCADA network security, Zone-Conduit model,firewall autoconfiguration, security policy, SCADA best practices, IP Fragmentation, Port Fragmentation.

I. INTRODUCTION

The process of categorizing packets into "flows" in an Internet router is called packet classification. All packets belonging to the same flow obey a predefined rule and are processed in a similar manner by the router. Packet classification is an enabling function for a variety of internet applications including Quality of service (QoS), security, monitoring, multimedia Communications. Growing and changing network requirements invokes need of larger filter with more complex rules, which in turn gives rise to different fast packet classification algorithms. Packet classification is needed for non-best-effort services, such as firewalls and intrusion detection, routers, ISPs and usually in the most computation intensive task among others. Services such as bandwidth management, traffic provisioning, and utilization profiling also depend upon packet classification. Packet consists of header and information data and header consists of MAC address, IP address, port number etc.

Traditionally, the Internet provided only a "best-effort" service, treating all packets going to the same destination identically, servicing them in a first come-first-served manner. However, internet users and their demands for different quality services are increasing day by day. So, Internet Service Providers are seeking ways to provide differentiated services (on the same network infrastructure) to different users based on their different requirements and expectations of quality from the Internet. For this, routers need to have the capability to distinguish and isolate traffic belonging to different flows. The ability to classify each incoming packet to determine the flow it belongs to is called packet classification and could be based on an arbitrary number of fields in the packet header. Packet classification is a multi-dimensional form of IP lookup and finding longest prefix matching to provide next-hop in routers.

Different algorithms for packet classification are as follows:

- GoT: Grid of Tries
- EGT: Extended Grid of Tries
- HiCuts: Hierarchical intelligent Cuts
- HSM: Hierarchical Space Mapping
- AFBV: Aggregated and Folded Bit Vector
- CP: Compression Path
- RFC: Recursive Flow Classification

- B-RFC: Bitmap aggregation Recursive Flow Classification
- H-Tries: Hierarchical tries
- SP-Tries: Set Pruning tries
- BV: Bit Vector
- ABV: Aggregated Bit Vector

Network security deals with the protection of data and resources in a communications network, while providing access to authorised users [1]. It is a crucial element of any modern day business in maintaining productivity, minimising disruptions and achieving regulatory compliance. Firewalls are the standard mechanism for enforcing network security. They protect the Confidentiality (C), Integrity (I) and Availability (A) of data and resources inside a network. SCADA networks control the distributed assets of many industrial systems. Power generation, water distribution and factory automation are just a few examples that illustrate the critical nature of these networks. SCADA networks are not like corporate IT networks, they have been designed primarily for reliability and SCADA devices often lack built-in security features for protection from cyber-attacks. Consequently, these devices depend on firewalls to protect them.

II. Related Work

BACKGROUND

SCADA networks are vital to the operation of a nation's critical infrastructure plants. Recently, there has been a significant increase in the number of plant disruptions and shutdowns due to cyber-security issues in these networks [4].

Poor internal network segmentation in SCADA systems is a significant contributor to the quick spread of security threats and attacks between subnets [1,4,12]. The ANSI/ISA standards introduce the concepts of zones and conduits as a way of segmenting and isolating the various sub-systems in a control system [1]. The zone-conduit model is a very useful starting point for a high-level description of security policy, and so we shall describe it in detail here.

A zone is a logical or physical grouping of an organisation's systems with similar security requirements based on criticality and consequence [1]. By grouping systems in this manner, a single security policy can be defined for all members of a zone. For example, 3 security zones can be defined to accommodate low, medium and high-risk systems, with each device assigned to its respective zone based on their security level needed. A low-risk system can be accommodated within a medium or high security zone without compromising security, but not vice versa.

The uniform security policy of a zone is used to guide the construction and maintenance of all systems within the zone [1]. Therefore, selected systems within a zone (e.g., a server) should not have their own separate policies. Allowing separate policies would impart an incorrect sense of security to those systems. These systems are only as secure as the zone itself, in the absence of any firewalls enforcing a real separation.

A conduit provides the communication path between two zones as well as the necessary security functions for them to communicate securely [1]. Since availability is paramount in a SCADA network, a conduit should resist Denial of Service (DoS) attacks and preserve integrity and confidentiality of network traffic. This is achieved using security mitigation mechanisms (e.g., firewalls) implemented within the conduit. In Figure 1, two typical zones in SCADA environments: the SCADA-Zone and the Corporate-Zone are shown connected by a conduit

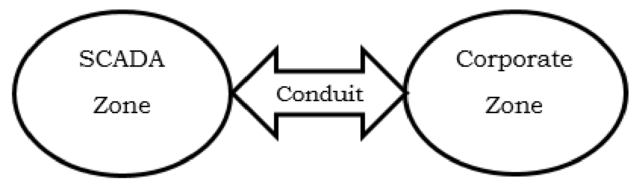


Figure 1: Example Zone-Conduit, adapted form[4]

.

A conduit cannot be a communications link that simply inter-connects zones without restricting traffic-flow [1]

From a security perspective, this does not provide any mitigation capabilities to the connecting zones. Such a conduit fails to enforce a clear separation of zones. It is equivalent to the two zones being a single zone and would prove useless for security policy description purposes. A conduit, in this view, always offers some security mitigation capability, typically using single or multiple firewall(s). A zone-conduit security model of a network is key to the accurate assessment of common threats, vulnerabilities, and required security mitigations to protect SCADA resources [1]. It provides a high-level view of an organisation's security and traffic control strategy. The model helps identify the disjoint security zones in the network, enabling the detection of serious design flaws such as the allocation of low and high security devices into a single zone. Such direct violations of the ANSI/ISA best practices would be a clear indication of exploitable vulnerabilities. The zone-conduit model also reveals unwanted inter-zone communication paths. It enables us to understand whether the security mitigation devices installed on each path are capable of offering the level of mitigation required for secure inter-zone communications. Zone and conduit concepts are intended as a platform for high-level security policy description. Before using these concepts in policy specification for firewalls, it is best to evaluate their usefulness. Particularly how well they cater for security architectures used in practice in real networks

III. METHODOLOGY

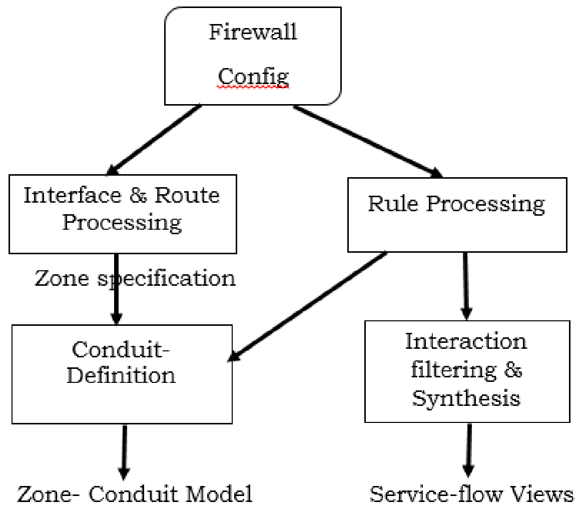


Figure 2: Firewall configuration parsing process

Firewall configurations are long and complex. For example, the SCADA firewall configurations we discuss consist of 1360 lines on average per firewall. We could not use existing tools such as Fang or Lumeta to analyse these as they do not support high-level policies based on the zone-conduit model. Hence we built an automated parser to parse configurations using zone-conduit concepts. We describe the Parser in detail here because it explains the use of zone-conduit concepts in the analysis of practical networks and allows to identify shortfalls in the model and help find solutions. The Parser is depicted in Figure 2. The details are described below:

Firewall Config: The input configuration text-file of a firewall containing interface configurations, static routes and ACLs. Multiple configuration files can be input for simultaneous processing.

Interface and Route Processing: The processing of firewall interface configurations and static routes. This extracts interface names, subnet IP addresses, security levels, additional network and gateway IP addresses. Details of this processing are covered in Subsection 3.1.

Rule Processing: The processing of ACLs assigned to firewall interfaces and any implicit rules. Implicit rules enable services through the firewall over and above ACLs. More details are discussed in Subsection 3.2.

Conduit-Definition: The definition of conduits that interconnect the security zones in the SCADA network. Details are covered in Subsection 3.3.

Zone-Conduit Model: The zone and conduit topology output of the SCADA network.

Interaction Filtering & Synthesis: The filtering of ACL rule interactions and synthesis with implicit rules. Details of this stage are covered in Subsection 3.6.

Service-flow views: The output traffic-flow views for the firewall. A service-flow view describes an enabled protocol through the firewall by zone.

Our current Parser uses one or more Cisco Adaptive Security Appliance (ASA) or Private Internet eXchange (PIX) or IOS firewall configurations as input. It begins by processing the individual firewall interface configurations. It also processes any static route configurations to identify the location of additional networks and gateways. Rule processing partly involves parsing the ACLs assigned to firewall interfaces. These indicate the traffic permitted to traverse each of the firewall interfaces. Additionally, rule processing also involves parsing implicitly enabled services. The Parser then performs conduit definition. This creates the ISA standard zone-conduit model. ACLs and implicit rules are also analysed by the Parser to filter-out any interactions present and synthesised to generate service-flow views as output.

3.1 Zone construction

The Parser analyses the interfaces and subnets defined in the firewall configuration to construct zones. It starts by assuming each interface connects to a disjoint zone, and then looks for indications that these potential zones should merge. A potential zone merge can be identified via indications of traffic leakage between the zones. These leakages occur outside of a firewall but can often be identified through the inspection of ACL contents. Where such a leakage exists, ACLs should control traffic flow equally for those services, on both firewall interfaces connected to the respective zones. By inspecting the ACL contents of a firewall, and applying the policy that inter-zone traffic-flow is allowed via firewall-only paths when redundant paths are available (i.e., not relayed through a 3rd zone), we can deduce that the assumed disjoint zones must form a single zone. The original disjoint zone-firewall model is then updated with identified merged zones. Static routes can help locate additional networks and gateways. Static routes contain IP address details of next-hop gateways and networks reachable via them. By identifying and including these, we can extend our zone-firewall model.

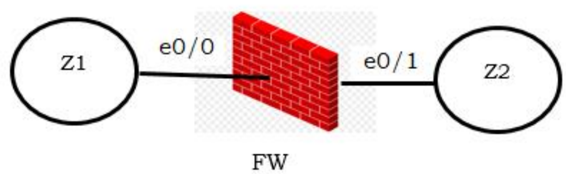
3.2 Implicit rules

In a Cisco firewall, traffic flows can be enabled explicitly through ACLs or implicitly via several alternate methods. One available method in ASA and PIX firewalls is to assign security levels to the firewall interfaces [8]. An interface security level is defined as a level of trust bestowed on the network connected to that firewall interface. In the absence of an ACL assigned to such an interface, certain traffic flows are permitted by default from an interface with a high security level to one with a lower security level [8]. Special configuration commands can also be used to enable services implicitly, for example to enable SSH or HTTP firewall management traffic into the firewall interfaces [8]. Accommodating such management traffic using zones is discussed later in Subsection 3.4.

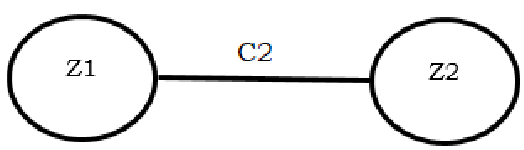
Implicit rules provide quick and easy alternatives to ACLs in enabling services through the firewall. They may not map to clear security policies but are convenient. However, auto configuration relies on clear security policies to permit traffic through a firewall. Implicit rules may aim to provide this, but we will see that they actually confuse the situation

3.3 Zone-Conduit model

As Section 2 discussed, a zone-conduit model describes the logical grouping of systems in a network. It gives a high-level view of an organisation's network segregation strategy. The Parser uses the zone-firewall model to generate a corresponding zone-conduit model. This is done by identifying the security conduits based on ANSI/ISA guidelines. A conduit is defined between zones, based on the available firewall only paths between them. An example conduit (C1) between 2 zones with a single-firewall path is shown in Figure 3. This case is almost trivial, but the question of how to map a network to zones and conduits has more complex cases.

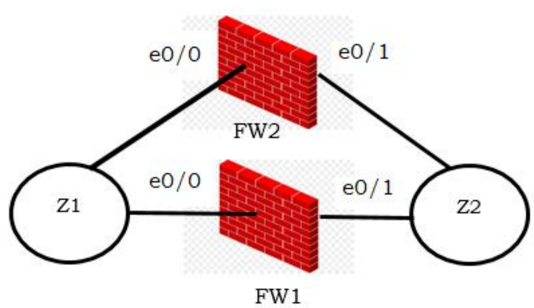


(a)Zone-Firewall model for 2 Zone separated by a Firewall

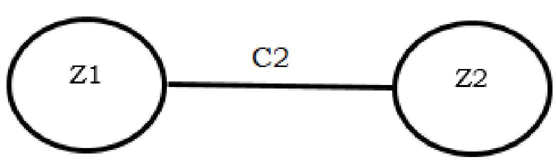


(b)Zone-Conduit model

Figure 3: Single Firewall Conduit Definition



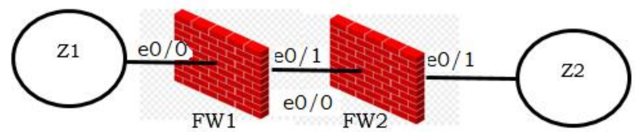
(a)Zone-Firewall model for 2 Zone separated by a Parallel Firewall



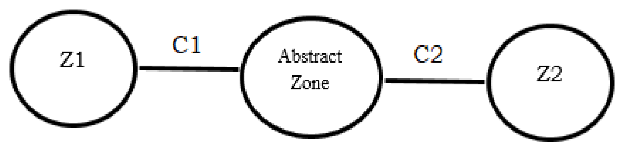
(b)Zone-Conduit model

Figure 4: Parallel Firewall Conduit Definition

When two zones are connected by parallel links, the ANSI/ ISA standard allows them to be modelled as multiple conduits. Doing so however, would imply that multiple security policies could exist between these zones when only one is possible from the strict interpretation of a zone. Hence, we define a single conduit to implement the single policy relationship. An example conduit (C2) is depicted in Figure 4b. Firewall paths can include firewalls in series (Figure 5a). This back-to-back firewall architecture is one of the industry recommended security architectures [5] where defence in depth is achieved by using different vendors' devices.



(a)Zone-Firewall model for 2 Zone separated by a Serial Firewall



(b)Zone-Conduit model

Figure 5: Serial Firewall Conduit Model

This firewall setup can also provide DoS protection by using one firewall to perform simple processing of a large volume of packets while the other firewall conducts complex processing (e.g., deep packet inspection) on a smaller number of packets [5]. Network engineers may also setup logging and alerts to originate from the firewalls differently, in such a circumstance. ANSI/ISA guidelines lack clear specification on how to define zones and conduits to precisely capture the traffic-flow requirements in this context. There is a single conduit containing both firewalls, if we dismiss the link between the firewalls. For automation, a single security conduit hinders precise specification of the distinct firewalls.

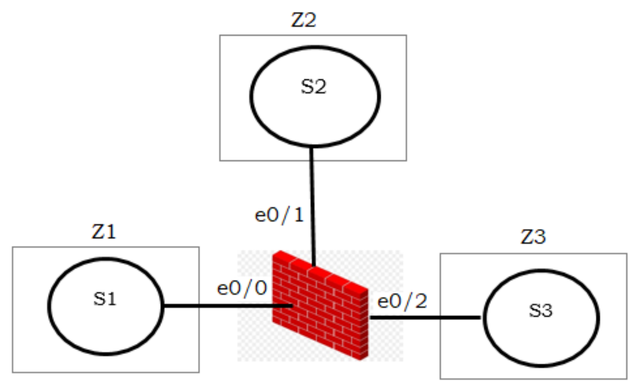
We propose to treat this connecting link as a separate zone, to overcome the specification shortfall. It is referred to as the Abstract-Zone in the absence of any real network devices within it (Figure 5b). The approach creates two separate conduits (C1 and C2), each containing one firewall. Auto-configuration can now leverage the distinct conduits to specify the individual policy requirements. We can also model the security properties of the subnet between the serial firewalls as a Demilitarised Zone (DMZ), in case devices such as a logger are added. A DMZ is used to expose an organisation's external-facing services (e.g., a mail server) to untrusted networks [5]. It adds a layer of security to the company's trusted internal networks by only providing direct external access to the hosts in the DMZ.

A conduit may also inter-connect more than two security zones [1]. ANSI/ISA guidelines lacks clear specification on appropriate conduit definitions in such circumstance. Consequently, the example zone-firewall model depicted in Figure 6a, could be modelled using a hyper-graph (Figure 6b). In this model, the firewall (FW) is located inside the hyper-edge conduit C1 which has one-to-many zonecommunication paths. This complex conduit can implement multiple security policies; between Z1 and Z2, Z2 and Z3 and Z3 and Z1. Catering for this complexity requires the conduit to track the participating zones per policy. There is also no clear mapping of the ACL rules enforcing the policy to the firewall interfaces. Hence, the hyper-graph conduit model is difficult to use for firewall auto-configuration purposes.

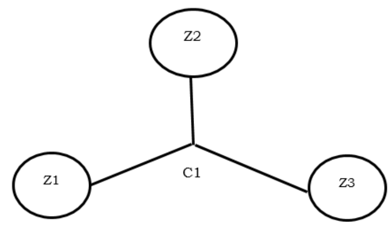
To simplify the complexities of hyper-edge conduits, we propose to generate a zone-conduit model that consists of only one-to-one zone-communication paths (Figure 6c). Each conduit now implements a single security policy between two zones. The simple design also requires each conduit to only contain the firewall interfaces attached to its connecting zones (e.g., C2 contains e0/0 and e0/1). A conduit path now reveals the exact firewall interfaces and their layout with respective to the connecting zones, enabling easy placement of required ACL rules. Consequently, the choice of simple-edge conduits, allows us to enforce a strict 1:1 mapping between conduits and policies. This restriction yields a precise highlevel specification, useful for firewall auto-configuration.

This logical method of conduit-definition leads to multiple conduits sharing the same firewall in their mitigation offering (e.g., C2, C3, C4 share FW in Figure 6c).

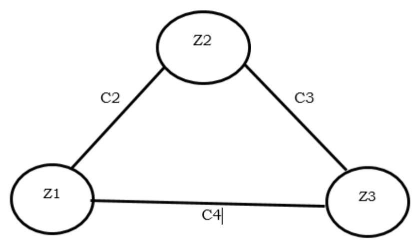
In summary, a single conduit need not always map to a single firewall. In-fact one-to-many and many-to-one mappings between conduits and firewalls are more useful for high-level security specification. However, our recommendation for firewall auto-configuration is that one conduit should always implement a single relationship between only 2 zones.



(a) Zone-Firewall model for 3 zones separated by a firewall



(b) Hyper-graph Zone-Conduit model.

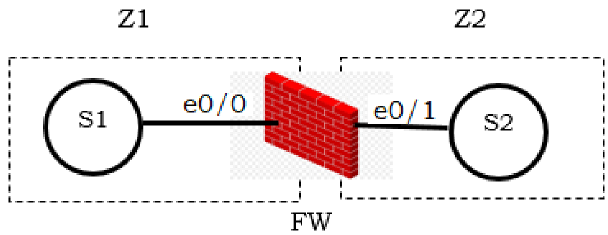


c) Simple-graph Zone-Conduit model.

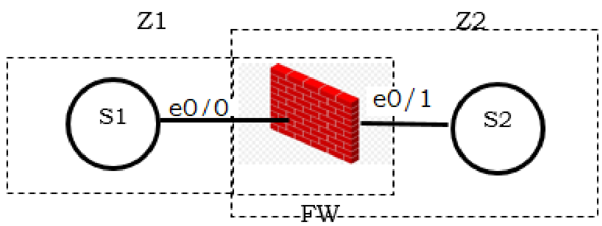
Figure 6: Conduit-Definition Alternatives

3.4 Firewall management access control

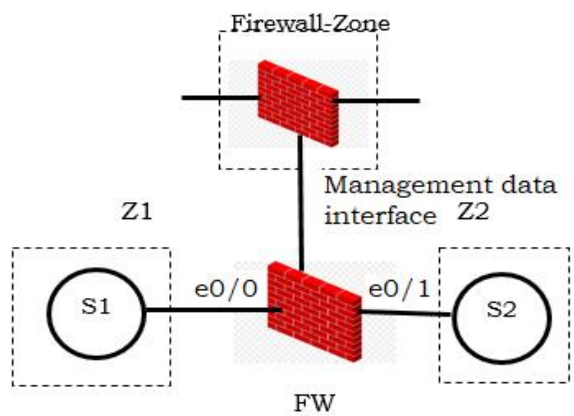
In addition to offering mitigation capabilities to zones, firewalls play a dual role by providing secure, authorised network management access to themselves. ANSI/ISA policy includes the use of a firewall within a conduit as a mitigation device, but does not clearly address how to use zone and conduit concepts to capture firewall management policy requirements. This is a critical shortfall, because if management of the firewall is compromised, the entire system is compromised. There are several possible ways to address the issue, as illustrated in Figure 7



(a) Firewall interfaces in zone



(b) Firewall shared by zone



(c) Dedicated Firewall-Zone

Figure 7 : Firewall-Zone alternatives for 2 subnets S1 & S2 seperated by a firewall

3.4.1 Firewall partially included in zones

With this approach, each firewall interface belongs to the zone directly connected to that interface (Figure 7a). It implies that all IP traffic to the firewall from hosts and subnets of zones Z1 and Z2 is allowed. While simple, this approach has obvious problems. The design prevents restriction of firewall access by traffic type to selected connected-zones. For example, disallowing HTTP access to the firewall by zone Z1 would be impossible with this type of a model.

3.4.2 Firewall shared between zones

This model assigns a firewall interface to all connected zones (Figure 7b), also implying removal of any traffic restriction between hosts and subnets within each zone and the firewall by default. The outcome is similar to that of 3.4.1, preventing placement of a required policy between a zone and the firewall.

3.4.3 Firewall in its own zone

Here, we exclude the firewall from belonging to any existing zone and place it separately in a new security zone on its own. This may seem more complex but actually represents the real situation well. This new Firewall-Zone (FWZ) is connected to the firewall (Figure 7c) via the ManagementData Interface (MDI). The MDI is a logical interface that provides traffic packets to the firewall's control and management plane (Figure 8) from the data path [9]. The control and management plane is responsible for processing the firewall bound management traffic, while the data path handles the traffic forwarded through the firewall.

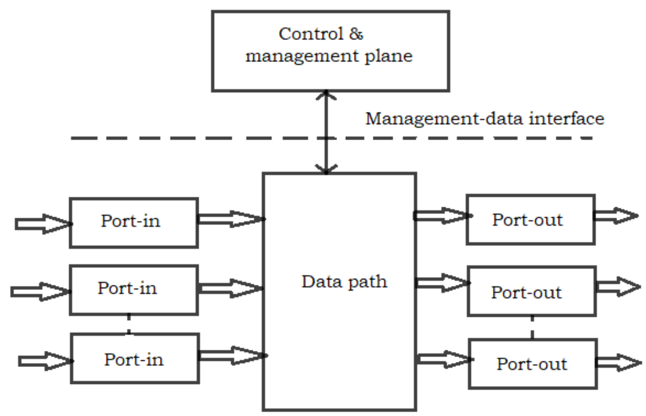


Figure 8: Logical firewall architecture adapted from[9], depicting the firewall-zone Management-data interface

The Firewall-Zone enables each zone to communicate while allowing restrictions to be placed on the firewall to regulate its management traffic. This model now captures the firewall's dual role precisely, and can now impose restrictions such as disallowing HTTP access to the firewall by zone Z1 (e.g., by placing ACL rules on interface e0/0).

Our solution of introducing a dedicated Firewall-Zone has a significant impact on simplifying management policy specification and auto-configuration of a firewall. It allows firewall management and non-management traffic to be considered equally, but to be specified separately in the auto-configuration process. This clean approach facilitates enforcement of further restrictions on the type of management traffic allowed (e.g., disallow Telnet), promoting compliance with industrial policies. Of course additional security mechanisms (e.g., password access) are required, but these are outside the current scope of this analysis.

The zone-firewall model, now precisely captures the distinct zones and their interconnections to the firewall. It includes implicit zones such as the Firewall-Zone required to facilitate firewall management and explicit zones such as the Corporate-Zone. By compiling a model that consists of a rich collection of these zones, their contents (i.e., network devices) and their respective interconnections to the firewall(s), we obtain a high-level view of the security strategy employed in the SCADA network.

3.5 Carrier network abstraction

Real networks commonly utilise a carrier network provided by a telecommunication service provider to interconnect geographically dispersed sites. This is prevalent in SCADA networks which control distributed field-site equipment from a centralised control centre, over for example, a leased line Wide Area Network (WAN). The traffic relayed via the Carrier network is controlled by the security policies between the zones within the two interconnecting sites (Figure 9). Due to the unavailability of every gateway and network-device configuration for analysis, we need a way to model the interconnectivity provided by a Carrier network while abstracting away its underlying implementation details.

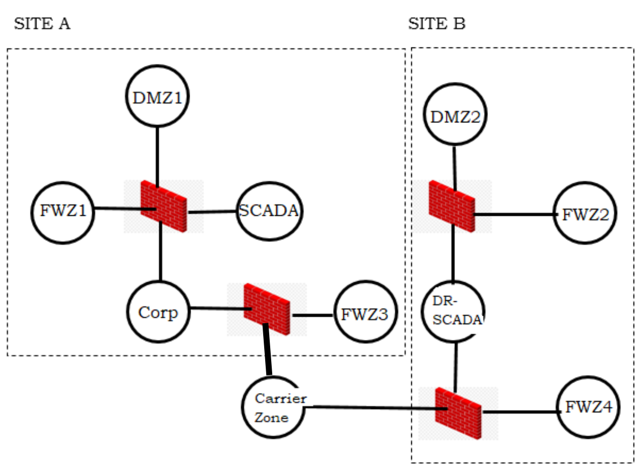


Figure 9: Carrier-Zone interconnecting geographically dispered sites

A simple yet effective strategy is to use a single Carrier Zone in the zone-firewall model as shown in Figure 9, that encompasses the Carrier network. This zone provides connectivity, facilitates security policy specification between the sites and abstracts away unwanted implementation details.

3.6 Service-flow Views

A service-flow view is a directed-graph of hosts (or their respective zones) that are allowed to initiate and/or accept that service protocol. The Parser generates these for the various traffic classes; IP, TCP, UDP and ICMP protocols, broken down by port, host and zone per traffic class. The output views are graphical representations based on GraphML and can be readily viewed using tools that support the format such as yEd [19].

A key goal in processing the firewall ACLs is to identify the types of services enabled explicitly between hosts and/or subnets and between zones. A few obstacles need to be overcome first, to gain this understanding.

Primarily, each rule-set within an ACL can contain potentially interacting individual rules referred to as intra-ACL interactions. These interactions are caused by rule-overlaps, triggered by distinct rules having common packet matching criteria described in Subsection 3.1. An example of such a scenario is provided in Listing 1, where rule1 and rule2 both apply to HTTP packets originating at host 10.0.1.18 destined to host web_svr.

In Cisco firewalls, the outcome of such a pair of rules depends on several factors. These include the order in which they are listed, the level of overlap (i.e., partial, full overlap or subset) and their rule actions. Traffic packets to which both rules equally apply, will be filtered via rule1 in the list (i.e.,by line 2 in Listing 1). rule2 is completely overshadowed. Traffic packets outside the rule-overlapping region (e.g., host 10.0.1.20), that still apply to rule1 or rule2 will continue to be filtered by their intended rule. Based on the extent of overlap, interacting rules can be classified as generalization's , shadowed-rules, partial-overlaps and conflicts. A generalization refers to the case where a subset of the packets matched to a rule has been excluded by one or more preceding rules with an identical action. A shadowed-rule is the opposite, all packets applicable to such a rule have already been matched by a preceding rule with an identical action. A partial-overlap refers to the case where the set of packets matched to a rule partially-intersect with another preceding rule with a similar action. A conflict occurs when the current rule intersects with preceding rules but specifies a different action.

We derive the net-effect of the intra-ACL interactions and generate an interaction free equivalent version (ACL V1) of the ACL. This allows to accurately view the services enabled by each ACL. As a by-product of this processing, the Parser generates a list of all intra-ACL interactions found. These inconsistencies can assist with security audits.

Secondarily, there can also exist inter-ACL interactions that alter a rule's intended behaviour. Figure 10 and Listing 2 present an example, where rule1 in acl-in permits HTTP traffic from host 10.0.1.25 to host web_svr. The same traffic is denied by rule1 in acl-out. Since acl-out inevitably applies to any traffic packet traversing from zone1 to zone2, the net-effect of rule1 is the equivalent of a null rule. Hence, the Parser also needs to analyse potential interACL interactions on ACL V1, to derive a second version (ACL V2) that is interaction free. ACL V2 now reflects the net-effect of all rule interactions possible for a given network.

The Parser also processes implicit rules based on interface security levels and generates an IP service-flow view depicting allowed generic traffic-flows. It processes special

Cisco configuration commands that permit firewall management traffic above ACLs. The Parser then generates corresponding implicit service-flow views for each service enabled.

Finally, the Parser synthesizes the explicit and implicit service-flow views to derive a comprehensive collection of views for each traffic class. These views accurately describe the overall services enabled through the firewall(s)

3.7 Ip addresses fragmentation:

IP address decomposition is done for both source address space and destination address space respectively .The fragmentation is done in the same manner for both source and destination IP addresses. For each address range (including address or subnet) appeared in the policy table, its two boundaries IP addresses are marked down in the corresponding source address or destination address IP space. After completion of construction of policies in the policy table, for each segment that is following at least one policy falls in it, an Equivalence class ID (eq: ID) number is assigned in the ascending order along the direction of increasing IP address, starting from 0.

There are many ways to map a given IP address (i.e. the source or destination IP address of a received packet) to a segment. In RFC (Recursive Flow Classification), this is achieved by taking any number of chunks that are convenient. We have kept the number of segments same at phase0 for all fields as that in HSM (Hierarchical Space Mapping) so that pre-processing time is not taken into consideration during analysis.

One of the main reasons why the Internet Protocol (IP) is enormously successful is that it can be used over virtually any physical media. In complex SCADA architectures, there is a variety of both wired and wireless media and protocols involved in getting data back to the central monitoring site. This allows implementation of strong IP-based SCADA networks over mixed cellular, satellite, and landline systems. SCADA communications can employ various ranges of both wired (telephone lines, optical fibers, ADSL, cables) and wireless media (radio, spread spectrum, cellular, WLAN, or satellite). The choice depends on a number of factors that characterize the existing communication infrastructure. Factors such as existing equipment, connections, available communications at isolated sites, data rates and polling frequency, remoteness of site, installation budget, and ability to accommodate future needs all impact the final decision for SCADA architecture.

A major enhancement in new SCADA systems comes from the use of WAN protocols such as the Internet Protocol for communication between the central station and communications equipment. RTUs can communicate with the master station using an Ethernet connection. A networked SCADA system. Another advantage brought about by the distribution of SCADA functionality over a WAN is that of disaster survivability. By distributing the processing across physically separated locations, it becomes feasible to build a SCADA system that can survive a total loss at any one location. Many of the traditional utility devices such as RTUs or even relays are today equipped with Ethernet interfaces. This, however, does not imply that all services can be migrated immediately in a plug-and-play manner to an IP-based communication infrastructure. Differential protection services are known as one of the most delicate applications.

Legacy SCADA system components may still work as initially designed. However, new operational and business processes often require new, higher-level functionality not included in the original components. Such extensions, including new physical and logical communication network connections, bear additional risks in terms of cyber security. SCADA systems were traditionally walled off from business systems and operated independently via the operational network only. Prior to the awareness of the risk of possible attacks, this seemed to provide all the protection the SCADA system needed. Their often proprietary character (operating systems, protocols, etc.) were often seen as additional safety assurance.

To run SCADA information over an IP network, various issues have to be considered such as operating equipment types, bandwidth used for SCADA center communication, network redundancy criteria and protection schemes, restoration times in case of failures, and other IP services within the network. There are several relevant advantages brought by IP technology. These advantages include the efficient use of bandwidth to avoid the allocation of capacity where it is not necessary, widely accepted standards based on proven technologies and a high degree of interoperability. Also, reliability is enhanced because in IP networks, packets are instantly rerouted if a node or link fails. Other related advantages are scalability to cope with growth, high degree of freedom to evolve network performance according to the strategic needs of use, optimization of the total cost of ownership, and taking into account initial investments and later costs of operation. Lastly, upgrades, maintenance, and related personnel cost and protection of the investment are secured by the integration of Ethernet/IP over existing transport networks.

Along with advancements in IP technology, IP-based SCADA systems have incorporated various beneficial features as well. These features include unlimited locations for servers and clients where users can install and move their SCADA servers, RTUs, and terminal servers to any site, which gives high flexibility in terms of redundancy and security and in the case of failure in SCADA servers where servers connected to the IP network provide mutual backup for optimized availability. Also, other benefits we can consider are service takeover and remote support as the control centers are not manned during the night. During this period, other regions can either take over the control or supervise log-ins via VPN in case of emergencies. Lastly, savings are obtained through IP-enabled RTUs; many front-end devices are no longer required since a lot of hardware, spares, and cabling can be saved and maintenance costs reduced.

3.8 Port number fragmentation:

The principle of port number fragmentation to get Port Sequence Number (PSN) is similar to IP address fragmentation. For the port number mapping, a direct look-up table is more efficient when there is enough memory to be allocated.

IV. RESULTS AND DISCUSSION

Security of IP addresses SCADA system:

SCADA systems were originally designed to control and monitor industrial processes using proprietary serial protocols. They were normally located away and secluded from other computer systems. However, in recent years, SCADA systems have been connected to corporate networks and the internet. This can enable businesses to monitor line processes and to support and enhance the process of making correct and beneficial decisions. However, the downside of this is that SCADA systems were never designed with security. With IP-based communications, unexpected threats that did not exist with legacy serial communications can occur at any point and anywhere. It is imperative that we understand how to securely design and to manage SCADA systems in internet-based settings and environments.

To protect SCADA systems from cyber threats, we have to perform the following tasks [4]. (1) The SCADA IP network should be located physically separate from corporate networks and other untrustworthy networks. When physical separation is not possible, logical separation must be applied. Logical separation is more complicated to implement effectively and runs the risk of ineffective configuration. One should avoid the use of the virtual LAN technology for keeping SCADA IP communications logically separated from corporate IP communications, as VLAN technology is not designed as a security measure but as a bandwidth-shaping tool. (2) IP communications that originate from untrustworthy networks from outside the SCADA system networks should terminate in a buffer network. They should not be allowed direct connections with components in the SCADA system networks; devices inside the SCADA system networks should not be able to communicate directly with the internet. Occasionally, existing corporate IT network infrastructure such as switches, routers, and WAN links must be used as a transport method for portions of the SCADA communications. If that is the case, then the SCADA communications should be encrypted and routed through a VPN tunnel that runs through corporate IT or other noncritical networks. Avoid SCADA devices that are dual-homed to two or more networks at different security zones or trust zones. (3) Additionally, when building a complete end-to-end IP network, avoid using devices that use layer 3 separations between SCADA and other noncritical networks. For proper network isolation, operate equipment that can provide a layer 2 separation. Lastly, a solid cyber defense must offer active blocking devices such as firewalls, IPS, and in-line network antivirus appliances. (4) Designs and procedures are another crucial component. Develop quality insurance techniques to ensure that all security requirements are recognized during the design phase and then executed and tested within the final product. In addition, consider using the ISA S99 security levels as a model when constructing SCADA systems based on IP protocols. If remote access to the SCADA system is permitted over an IP-based network, do not allow users to undergo a similar authentication process used to log into the corporate network. Instead, a different authentication procedure should be applied.

Once a unique SCADA IP-based network is designed and constructed; here are eight recommendations to follow to manage security. First, disable unnecessary services which apply to IP-enabled telecommunication devices, network equipment, PLCs, RTUs, protocol gateway converters, and any other embedded device. Second, limit the utilization of clear text protocols such as telnet, ftp, and http. Instead, force the use of encrypted protocols where technically possible. Third, ensure that the latest version of the Simple Network Management Protocol (SNMP) is up to date, since most IP-enabled telecommunication devices are supported for monitoring the health and performance of the devices. Fourth, keep an event log resident on the device and have a copy sent down to the centralized Syslog server. Fifth, consider deploying in-line network appliances at the choke points that perform network intrusion prevention and antivirus functions. In this way one can filter and drop packets and traffic known to be malevolent based on heuristics and signature matches. Sixth, firmware for IP-enabled telecommunications equipment and control devices should be kept up to date with the latest version. Seventh, control devices such as PLCs, RTUs, smart meters, Ethernet I/O, and IP-enabled instrumentation should be employed with an encryption of PIN code. Eighth, any network devices in front of control devices should be given rate-limiting commands to restrict and limit data from flooding the device.

4. Performance Metrics

The merit of a routing protocol is qualitatively and quantitatively judged by the performance metrics. The following performance metrics are considered:

4.1 Throughput

The number of bytes of data successfully delivered per unit time is termed as throughput, which is controlled by available bandwidth, signal-to-noise ratio and hardware limitations. The throughput is usually measured in bits per second (bit/s or bps) but it indicates how data stored in database of network, and sometimes in data packets per second or data packets per time slot.

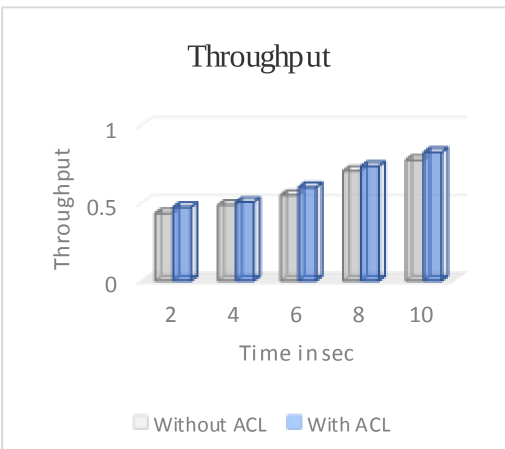
The aggregate throughput is measured as the sum of the data rates which are delivered to all the terminals in a network.

Throughput = ∑PR/∑(tst)-∑(tsp)

Where, PR – Received Packet Size, tst – Start Time, tsp – Stop Time. Unit-bps (bits per second)

awk file will calculate the throughput with running time (throughput versus time).
The following code will count all the received application packets in a network such that we can calculate the network throughput. If a throughput of a specific node has to be calculated, then we can simply add the node_id in the if condition.
The code simply prints the observed throughput during the time interval throughout the simulation time. We can change the time_interval variable according to our requirements.
In the following code:

- packet_size * recv * 8.0 gives the total number of bits received. Packet size is the size of packed used in Application layer.
- Dividing the value by 1000 gives us the throughput in kbps.



4.2Energy consumption

The nodes are participating in network and its working based on individual energy levels. Here we can calculate energy levels of nodes and maintain the routing. The bytes of data collected and depend on energy consumption. The network process working on lifelong must know energy consumption.
The energy model represents the energy level of nodes in the network and it shown in table 2. The energy model defined in a node has an initial value that is the level of energy the node has at the beginning of the simulation. This energy is termed as initialEnergy_. In simulation, the variable "energy" represents the energy level in a node at any specified time. The value of initial Energy_ is passed as an input argument. A node loses a particular amount of energy for every packet transmitted and every packet received. As a result, the value of initialEnergy_ in a node gets decreased. The energy consumption level of a node at any time of the simulation can be determined by finding the difference between the current energy value and initialEnergy_ value. If an energy level of a node reaches zero, it cannot receive or transmit anymore packets. The amount of energy consumption in a node can be printed in the trace file. The energy level of a network can be determined by summing the entire node's energy level in the network.

Table: Energy model's attributes:

Attribute	Meaning	Value	Default value
Energy model	Type of energy model	Energy model	None
rxPower	Power for receiving one packet	Power in watts (i.e.0.4)	281.8mw
txPower	Power for transmitting one packet	Power in watts (i.e. 1.0)	281.8mW
Initial energy	Energy of node in the Beginning	Energy in joules	0
Sleep power	Power consumed during sleep state	Power in watts	
Transition power	Power consumed during state transition from sleep to idle	Power in watts	
Transition time	Time in seconds taken during transition	Seconds	

Energy analysis:

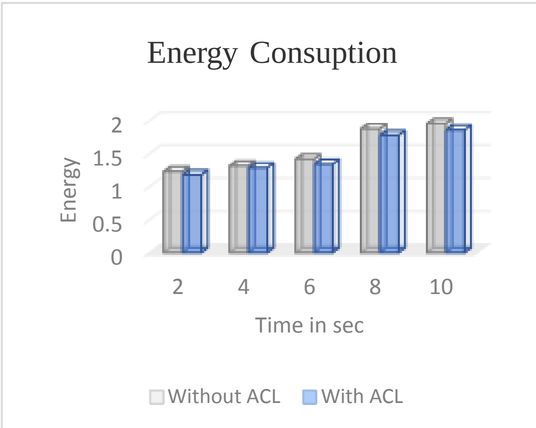
After simulation energy stored in following format in trace file:

[energy 998.999217 ei 1.000 es 0.000 et 0.000 er 0.001]

In above formate first name of attribute is given then it's value.

- energy: total remaining energy
- ei: energy consumption in IDLE state
- es: energy consumption in SLEEP state
- et: energy consumed in transmitting packets

- er: energy consumed in receiving packets



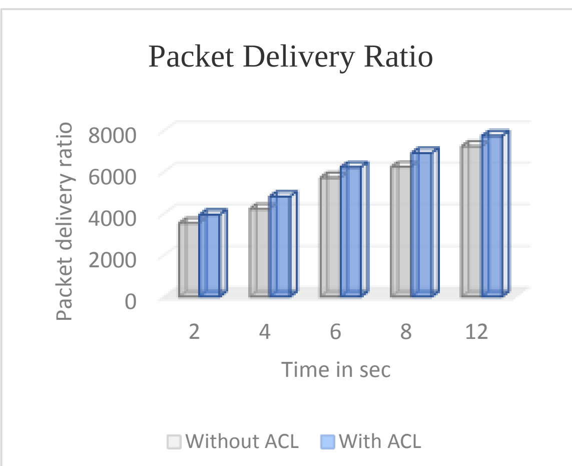
4.3Number of Dropped Packets

The number of dropped packets is the measure which indicates the packets of information which are dropped at the nodes due to repetitions or due to congestions. The number of dropped packets is measured at each node throughout simulation time.

Packet loss in a communication is the difference between the generated and received packets. Packet Loss is calculated using awk script which processes the trace file and produces the result.

Packet drop ratio is defined as

$PLR = \text{Generated packets} - \text{Received packet}$



V. DISCUSSION

Our case studies allowed us to identify the requirements for auto-configuration of firewalls. Most prominent is a good set of high-level abstractions.

Implicit rules are nascent attempts to provide high-level abstractions, but are too restrictive that you cannot write flexible rules. For example, Cisco security levels allow quick and easy access between internal and external firewall interfaces, but lack the flexibility to specify detailed traffic restrictions. Hence the large ACLs supplementing these levels Likewise, ANSI/ISA zone-conduit abstraction was too flexible, allowing alternate ways of defining zones and conduits to cater for business models. The abstraction is good when used by humans, but for automation we need precision. A good abstraction is a tussle between the above two approaches. It should provide clear mapping between policies and networks, with some restrictions but also the required amount of flexibility.

For instance, the standards allow 1:n or n:1 mapping between conduits, firewalls and policy. We argue that maintaining a 1:1 mapping between policies and conduits leads to a simple, understandable and useful abstraction for high level policy specification. Otherwise the ambiguity might lead to specification of policies that breach the restrictions implied by a zone, i.e., a single policy within a zone. For another instance, when firewalls are placed in series, the best practice is vague about how zones and conduits should be defined. We argue that there needs to be an Abstract-Zone to capture the distinct policies that could be reasonably applied to the two firewalls.

ANSI/ISA best practices also lacked specification on how to precisely capture firewall management traffic. Adding a Firewall-Zone addressed the problem. The service-flow views generated, also play an important role in auto-configuration. They help verify that the nettraffic flows enabled through firewalls match those specified via high-level policy.

Any discrepancy would indicate flaws in the auto-configuration process. The average firewall configuration length in our case studies, was 684 lines. It is trivial to accidentally leave-in lapsed ACL rules inside a lengthy configuration, when the composition of network devices changes with time.

These rules can lead to potentially dangerous outcomes and keep firewall configurations from being concise and up-to-date. An auto-configuration process should therefore, allow detection and removal of obsolete rules. ACLs and implicit rules can have complexed interactions.

For example, a rule within an ACL can overlap and conflict with other preceding rules in the same ACL, potentially altering or even reversing its intended effect. With lengthy ACLs, managing interaction free rule-sets manually is a near impossible task but is addressable through automation. Implicit rules can override ACLs, rendering the effort tendered to the careful design and deployment of ACLs obsolete. For example, consider using security levels through firewall interfaces. It continues providing network access implicitly, in the absence of ACLs and can easily be overlooked. We assert that the use of implicit rules should be avoided where possible, and replaced with explicit ACL based access control instead. This will be the difference in being able to automatically generate clear, simple and effective firewall configurations from confusing, complex and ineffective ones.

Our case studies did not comprise large, complex networks. This simplicity implies that the task of configuring the network firewalls should be relatively easy. Additionally, due to the critical nature of the industrial control equipment protected by these firewalls, one expects them to be correctly configured. As we found, this is far from reality. Even in the simplest of cases, SCADA firewalls are still badly configured!. Needless to say, what chances do we have of correctly configuring firewalls in a large, complex network? We have taken a significant step towards making firewall auto-configuration a reality. By refining the ANSI/ISA zone-conduit abstraction we make it precise and complete. Firewall configuration is complex and difficult as re-asserted by our case studies. The refined zone-conduit model, provides a precise, simple yet rich high-level abstraction for firewall policy description that is suitable for automation.

VI. Conclusion and Future Scope

CONCLUSIONS

The existing system provide a zone-conduit model for firewall policy specification, but the model lacks key aspects for IP level configuration and automation of firewall configuration. We propose new type configurations such as port number fragmentation and IP address fragmentation. The fragmentation is done in the same manner for both source and destination IP addresses. After completion of construction of table for individual data in every node setup, given the equivalent unique ID number is assigned in the ascending order along the direction of increasing IP address starting from zero. We conclude that our simulation process in network done by using NS2 simulator with level performance.

FUTURE SCOPE

We can extend the security policies used in single port to multiple ports with multiple verification conditions. This will improve the verification intensity. Also we extend our work with the support of fuzzy logics. Also we could implement our proposed algorithm with active control lists (ACL), and analyze security intensity.

References

[1] Dinesha Ranathunga, Mathew Roughan: Case Studies of SCADA Firewall Configurations and the Implications for Best Practices. IEEE Transactions on Network and service manaagement, Dec 2016

[2] ANSI/ISA-62443-1-1. Security for industrial automation and control systems part 1-1: Terminology, concepts, and models, 2007.

[3] Y. Bartal, A. Mayer, K. Nissim, and A. Wool. Firmato: A novel firewall management toolkit. ACM Transactions on Computer Systems (TOCS), 22(4):381–420, 2004.

[4] S. Bellovin and R. Bush. Configuration management and security. IEEE Journal on Selected Areas in Communications, 27(3):268–274, 2009.

[5] J.D. Guttman: Filtering postures: local enforcement for global policies. In IEEE Symposium on Security and Privacy, pages 120-129, 1997

[6] Ondrej Rysavy, Jaroslav Rab, Microslav Sveda: Improving security in SCADA systems through firewall policy analysis Federated Conference on Computer Science and Information Systems March 2013

[7] Khaled Salah, Khalid Elbadawi, Raouf Boutaba: Performance Modelling and Analysis of Network Firewalls, IEEE Transactions on Network and Service Management (Volume: 9, Issue: 1, March 2012)

[8] Avishal Wool: Trends in Firewall Configuration Errors: Measuring the Holes in Swiss Cheese, IEEE Internet Computing 14(4):58 - 65 · September 2010

[9] S. Bellovin and R. Bush. Configuration management and security. IEEE Journal on Selected Areas in Communications, 27(3):268–274, 2009.

[10] E. Byres. Using ANSI/ISA-99 standards to improve control system security. White paper, Tofino Security, May 2012.

[11] E. Byres, J. Karsch, and J. Carter. NISCC good practice guide on firewall deployment for SCADA and process control networks. National Infrastructure Security Co-Ordination Centre, 2005.

[12] M. Casado, T. Garfinkel, A. Akella, M. J. Freedman, D. Boneh, N. McKeown, and S. Shenker. SANE: A protection architecture for enterprise networks. In Usenix Security, 2006.

[13] W. R. Cheswick, S. M. Bellovin, and A. D. Rubin. Firewalls and Internet security: repelling the wily hacker. Addison-Wesley Longman Publishing Co., Inc., 2003.

[14] Cisco Systems. Cisco ASA 5500 Series Configuration Guide using the CLI. Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706, USA, 2010.

[15] Cisco Systems. Cisco ASA 5585-X adaptive security appliance architecture. White paper, Cisco Systems, May 2014.

[16] R. Jamieson, L. Land, S. Smith, G. Stephens, and D. Winchester. Critical infrastructure information security: Impacts of identity and related crimes. In PACIS, page 78, 2009.

[17] K. Stouffer, J. Falco, and K. Scarfone. Guide to Industrial Control Systems (ICS) security. NIST Special Publication, 800(82):16–16, 2008.

[18] T. Tuglular, F. Cetin, O. Yarimtepe, and G. Gercek. Firewall configuration management using XACML policies. In 13th International Telecommunications Network Strategy and Planning Symposium, Sep, 2008.

Authors Profile

Mr. M sai pradeep kumar pursed Bachelor of Technology from Raghu Engg College, Visakhapatnam in 2016 and Master of Technology from University College of Engineering Kakinada (A), Kakinada in year 2018. His main research work focuses on Network Security and Digital Forensics.

Dr D Haritha currently working as Professor in Department of Computer Science and Engineering, University college of Engineering Kakinada (A), Kakinada since 2003. She is a member of IEEE & IEEE computer society. She has published more than 20 research papers in reputed international journals and conferences including IEEE and it's also available online. Her main research work focuses on Image Processing. She has 19 years of teaching experience and 3 years of Research Experience.