Survivable SCADA Systems: An analytical framework using performance modelling

Carlos Queiroz, Abdun Mahmood, and Zahir Tari
Computer Science and Information Technology
RMIT University
Melbourne, Australia
Email: {cqueiroz, abdun.mahmood, zahirt}@cs.rmit.edu.au

Abstract—Supervisory Control and Data Acquisition (SCADA) systems control and monitor industrial and critical infrastructure functions, such as the electricity, gas, water, waste, railway and traffic. Recently, SCADA systems have been targeted by an increasing number of attacks from the Internet due to its growing connectivity to Enterprise networks. Traditional techniques and models of identifying attacks, and quantifying its impact cannot be directly applied to SCADA systems because of their limited resources and real-time operating characteristics. The paper introduces a novel framework for evaluating survivability of SCADA systems from a service-oriented perspective. The framework uses an analytical model to evaluate the status of services performance and the survivability of the overall system using queuing theory and Bayesian networks. We further discuss how to learn from historical or simulated data automatically for building the conditional probability tables and the Bayesian networks.

Index Terms—SCADA systems, Performance metrics, Survivability

I. INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) systems control and monitor industrial and critical infrastructure functions, such as the electricity, gas, water, waste, railway and traffic. SCADA systems have become target of an number of attacks due to its increasing connectivity to corporate networks. Demonstration of some successful attacks highlight the fact that critical infrastructure are no longer immune to attacks. Consequently there is an urgent need to assess the ability of a critical system to survive under attack.

Traditional techniques and models of identifying attacks, and quantifying its impact cannot be directly applied to SCADA systems. There is little widely available knowledge about the true nature of SCADA vulnerabilities since corporations and government industries that run these systems do not disclose the attacks in fear of losing public confidence and revenue. Even SCADA traffic datasets are not publicly available [1], [2], [3]. As a result, techniques that rely on detecting attacks, such as signature based IDS do not work well in the absence of a comprehensive database of attack signatures. Furthermore, IT systems are generally secured by means of either host based or perimeter-based IDSs that can detect executable malicious codes, virus and worms. Similar solutions cannot be applied to SCADA field devices due to their limited resources and real-time operating characteristics.

This paper focuses on the assessment of critical systems’ ability to keep its essential services running even under attack [4], [5], [6], [7]. It measures SCADA services performance to detect anomalies that are likely to be linked to attacks. The paper introduces a novel technique of applying performance metrics to assess system survivability [8], an approach that aims to keep an acceptable level of services running in the presence of failures or attacks. In addition to that, it also discusses how to learn from historical or simulated data automatically for generating the conditional probability tables and the Bayesian networks.

Section II discusses some related work on evaluating survivability through performance measurement. Section III introduces the analytical framework. It describes the performance metrics used to measure SCADA services analytically and show how these metrics can be aggregated into a Bayesian network to quantify survivability. Section IV describes how Bayesian networks can be devised by using machine learning algorithms. Section V presents a case study to evaluate the model.

II. RELATED WORK

The concepts of services essentiality and dependency are core to the framework. A service failure will affect the service itself and all other services depending on it. Therefore, survivability cannot be evaluated without looking at services and all their dependencies.

Due to space constraints the proposed model is compared to Liu and Trivedi [4] framework only, with both using analytical models to quantify survivability based on system performance. Liu and Trivedi [4] proposed a framework that combines performance and availability to quantify performance and survivability. The performance model is a homogeneous CTMC model based on the $M/M/m/m$ queuing model [9]. It assumes memoryless inter-arrival and service times with $m$ servers or processors and $m$ jobs (requests). While the performance model handles performance measurements the availability model uses the notion of failures replacing arrival-time and service rates with failure and repair rates, respectively.

The composite model evaluates the entire system as if the service model worked uniformly across all its services. Sadly, this assumption does not hold for most systems (SCADA and
otherwise). Services may behave differently based on their characteristics and roles in the system. Evaluating the system in terms of performance and availability without considering the system’s heterogeneity gives an inaccurate system picture. Therefore, it is not flexible enough for use on different types of SCADA systems where there are diverse services. The proposed model overcomes this limitation by considering both a service based model and service interrelationships. Furthermore, since the proposed work is focused on the implication of attacks on SCADA systems, it is assumed to be a fault-free system (i.e. does not consider hardware and software faults for performance degradation).

The two models are compared based on the key properties of a SCADA system as follows:

- **System Heterogeneity** - this property defines the capacity of the model for handling different type of services in the system (e.g. having different arrival rates and service times).
- **Service Interdependency** - this property defines whether the model takes into account the relationship and dependencies between services.
- **System Availability** - this property deals with the aspects of availability, failures and repairs of a system.
- **System Behaviour Insights** - the capacity to predict and model system behaviour based on system evaluation.

The main differences between the proposed model and Liu and Trivedi [4] model are summarised in Table I.

<table>
<thead>
<tr>
<th></th>
<th>Liu’s Model</th>
<th>Proposed model</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Heterogeneity</td>
<td>–</td>
<td>√</td>
</tr>
<tr>
<td>Service Interdependency</td>
<td>–</td>
<td>√</td>
</tr>
<tr>
<td>System Availability</td>
<td>√</td>
<td>–</td>
</tr>
<tr>
<td>System Behaviour Insights</td>
<td>√</td>
<td>–</td>
</tr>
</tbody>
</table>

As evident from Table I, Liu’s model does not present the concept of service hierarchy and, more importantly, fails to consider service dependency. Since SCADA services are distributed and interdependent it is imperative that the service model captures this relationship. Their model does not apply for systems that rely on service interaction, where some services are more important than others.

III. THE ANALYTICAL FRAMEWORK

The proposed framework defines an analytical model that quantifies and predicts the survivability of SCADA systems based on attributes such as service processing time and network traffic. It combines queuing theory and Bayesian networks, and is sufficiently flexible to replace the analytical parts by runtime data from live systems. It calculates survivability for each individual service and then aggregates the results into a final score. It is divided into two main parts:

A. Individual service performance analysis

B. Aggregating service metrics

A. Individual service performance analysis

Field services run on field devices with very limited resources (e.g. RTU, PLC). Enterprise services run on more powerful hardware, usually server machines, which are designed to be scalable (e.g. MTU, HMI Server). The metrics measure the probability of services being in one of the three possible status: Normal, Degraded or Unavailable.

Most SCADA services act in a client/server fashion. Field services receive requests from MTUs. HMI servers receive requests from HMI clients. Historians receive requests from HMI servers and MTUs. There is a strong interdependency between services. A service failure may cascade into the failure of other services and sometimes of the entire system. That is why a service characterisation is important in order to define specific performance models based on its service type. Detailed description and characterisation of SCADA services can be found elsewhere [10].

1) Field service metric: Field services are modelled as Continuous Time Markov Chains (CTMC). The number of requests in the service represents the state of the Markov chain. Field services have limited capacity for handling requests. This characteristic can be represented analytically by the $M/M/1/K$ queuing model [11], [12]. It is assumed that request arrival rates are Poisson processes with $\lambda$ rates, service times are exponentially distributed with rate $\mu$ and buffer size $K$ (including one being processed) and runs on a FCFS (First Come First Serve) basis.

The $M/M/1/K$ model has been chosen because it has limited buffer $K$ and it is used to model servers with one CPU only. It is reasonable to assume that field devices have only one CPU and limited capacity to handle requests.

The probability of a request being rejected is proportional to the time the queue is full [12]. The probability defines the unavailable status of the service. Equation 1 calculates the probability. The Equation is derived by solving the balance equations that show the limiting probabilities of each state in a CTMC [11], [12].

$$P_K = \begin{cases} \frac{(\lambda/\mu)^K}{1-(\lambda/\mu)^K}, & \text{if } \lambda \neq \mu, \\ \frac{1}{K+1}, & \text{if } \lambda = \mu. \end{cases} (1)$$

A service is considered as degraded when its service time is over a threshold $t$ value. The threshold is defined by a system expert.

The degraded status probability is defined by the summation of $P_n$, probability for all states where $n * \frac{1}{\mu}$ is greater than $t$ for $1 < n < K$. More formally,

$$P\{\text{Degraded}\} = \sum_{i=2}^{n} P_i, \quad \text{if } i * \frac{1}{\mu} > t, \quad (2)$$

$$P\{\text{Unavailable}\} = P_K \quad (3)$$

Given $\lambda$, $\mu$ and $K$, $P\{\text{Unavailable}\}$ is calculated by Equation 3, $P\{\text{Degraded}\}$ by Equation 2 and $P\{\text{Normal}\}$ that represents the probability of normal status by Equation 4.
\[ P\{\text{Normal}\} = 1 - P\{\text{Unavailable}\} - P\{\text{Degraded}\} \quad (4) \]

2) Enterprise service metric: The metric is represented by the \( M/M/n \) queuing model [13]. Enterprise services are modelled as CTMCs where the state of the Markov chain is defined by the number of requests being processed. It assumes a Poisson arrival rate, an exponential service time and requests are handled in a FCFS (First Come First Serve) basis. The \( M/M/n \) assumes an infinite buffer and can process \( n \) requests at once.

Enterprise services represent common application servers such as HTTP and database servers. It is reasonable to assume that such services can either be deployed in cluster environments with \( n \) servers or in multiprocessor machines with \( n \) CPUs, consequently making them able to process \( n \) requests at once.

Equations 5 and 6 are used to calculate the status probabilities. Equation 5 calculates the probability a request has to wait to be processed, where \( n \) is the state and \( u = \frac{\lambda}{\mu} \) represents the traffic intensity. Equation 6 calculates the probability a request has to wait less than a threshold \( t \), and will be used to calculate the degraded status probability, \( P\{\text{Degraded}\} \).

\[
E_c(n, u) = \frac{u^n}{n!} \left(1 - \frac{\lambda}{\mu}\right) + \sum_{k=0}^{n-1} \frac{u^k}{k!} \quad (5)
\]

\[
P\{0 < W \leq t\} = 1 - E_c(n, u) \cdot e^{-\frac{(n-u)}{\mu} \cdot t^2} \quad (6)
\]

Table II illustrates the algorithm for calculating the status probabilities of enterprise services. We define a service as being normal when there is no delay to process requests, i.e. no waiting time \((W)\). The probability of being normal is defined by Equation 9. The probability of being degraded is defined by Equation 7. Because the model assumes every request will be processed (sooner or later) the unavailable status probability is calculated by Equation 8. The threshold \( t \) is defined by a system expert.

\[ P\{\text{Degraded}\} = P\{0 < W \leq t\} \quad (7) \]

\[ P\{\text{Unavailable}\} = 1 - P\{\text{Degraded}\} - P\{\text{Normal}\} \quad (8) \]

\[ P\{\text{Normal}\} = 1 - E_c \quad (9) \]

### TABLE II

<table>
<thead>
<tr>
<th>Normal</th>
<th>Degraded</th>
<th>Unavailable</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>0.15</td>
<td>0.05</td>
<td>U</td>
</tr>
</tbody>
</table>

**B. Aggregating service metrics**

The proposed framework uses the predictive aspect of Bayesian networks [14] to understand how a SCADA system will perform based on its services’ behaviour. Figure 1 illustrates the Bayesian belief network that represents a simple SCADA system that consists of common SCADA services [15]. It shows the interrelationship between the services, and their effect on one another. Arcs represent the dependencies between them. The Modbus Slave and Historian database services influence the Modbus Master and OPC Server services, which consequently influences the HMI server service. The overall system status is directly influenced by the HMI Server and the OPC Server.

![Bayesian belief network](image_url)

The Bayesian belief network defined on Figure 1 calculates the probabilities of the system being in one of the three possible states. The overall state representing the system survivability is derived from the probabilities of the degraded and normal states.

The Conditional Probability Table (CPT) described in the figure defines the service dependency and essentiality among all the system’s services, and is used to measure the services’ influences on one another. Each row in a CPT represents the conditional probabilities for each possible combination of status of parent services. The values shown in the figure are just an example of how the tables can be built.
IV. LEARNING THE BAYESIAN NETWORK FROM DATA

It is infeasible to get the CPT values from system experts in order to build the Bayesian belief network. A typical SCADA system may contain hundreds, often, thousands of services, which makes the process of manually building the CPT values impractical. Therefore, the network should be built based on training data collected either from simulations or from real systems.

To calculate the CPT values shown in Figure 1 we have to learn it from a dataset of training data. There are numerous algorithms to learn Bayesian networks such as the K2 algorithm proposed by Copper et. al [16]. The main idea behind this technique is that after running the system for a while, performance data from the various services are collected and a training dataset is generated. The training dataset should cover a large number of representative data in order to avoid bias. Figure 2 illustrates an example of possible training dataset for a SCADA system.

Although, the automated learning of Bayesian network for SCADA survivability evaluation is not expanded in this paper, it is important to note that our proposed methodology applies even when the Bayesian network cannot be generated manually due to its size and complexity. To these cases we have built a SCADA testbed [17] that implements a widely used SCADA protocol, Modbus [18], to generate the datasets.

<table>
<thead>
<tr>
<th>RTU</th>
<th>MTU</th>
<th>HMI Server</th>
<th>Historian Database</th>
<th>Survivable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Normal</td>
<td>Normal</td>
<td>Normal</td>
<td>YES</td>
</tr>
<tr>
<td>Degraded</td>
<td>Normal</td>
<td>Unavailable</td>
<td>Degraded</td>
<td>NO</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Fig. 2. Training dataset sample

V. FRAMEWORK EVALUATION

In order to evaluate the proposed framework a case study that shows the operation of a simple SCADA system is demonstrated. Note that the process and data rates are for illustration only, since real SCADA systems are highly complex.

The proposed example addresses part of a SCADA system that controls and monitors an iron smelting process for steel production. A sensor reads the weight of the container on a scale. An actuator is used to open and/or close a valve that allows melted iron pours into a container. A RTU controls this process by receiving temporal updates from the sensor. In cases where a reading is over a threshold the RTU can take the decision to close the valve to stop the flow.

From the enterprise side operators monitor this process through a computer application (HMI) that shows graphically the readings from the RTU. Any changes executed by operators are sent to RTUs via the MTU unit. If the sensor reports a weight equals or over a specific threshold the RTU sends a message to the actuator to close the valve and then sends an alert message to the MTU. All this process happens automatically without any human intervention. However, system operators may intervene in the process any time during the operation. For instance, they can stop the processing by closing the valve.

Under normal conditions the HMI server queries the MTU three times per minute. The sensor updates its status to the RTU every second and the MTU also queries the RTU every second.

As the case study is restricted on showing how the performance of the RTU can be modelled analytically and consequently, how it will affect the overall probability of survivability of the system, the calculations below will take into consideration only the arrival and service rates of the RTU. We assume a RTU average service time \( \frac{1}{\mu} = 0.2 \) and a buffer size \( K = 5 \). The threshold \( t \) is defined as 50% higher than the mean service time.

\[
\begin{align*}
\lambda &= 2 \\
\mu &= 5 \\
\text{threshold}(t) &= 0.3
\end{align*}
\]

According to Section III the probabilities of the RTU’s states are calculated using the field service metric defined in Section III-A. These probabilities are then applied to the Bayesian network illustrated on Figure 1. Then, the Bayesian network shows the final probability of the system being survivable. Figure 3 (A) illustrates the final score. Under the running conditions defined above (arrival and service rates) the system has a high probability of being survivable, nearly 80%.

A. Attack model

Attacks such as DDoS (Distributed Denial of Service) or DoS (Denial of Service) are based on flooding the destination with huge amounts of network packets. In the proposed analytical model such attacks can be recreated by increasing the arrival rates of the services. Massively high arrival rates will degrade the performance of the system’s services and, in some cases, they will make the services completely unavailable. Such performance degradation will be reflected in the survivability of the services and of the overall system. For instance, just by increasing the original arrival rate of the RTU by five times \( \lambda = 2 \times 5 \) makes the probability of the RTU being normal decline to around 5%. As consequence the MTU and the final survivability probabilities have also changed. The new system’s probability of being survivable has dropped to around 55%, a nearly 25% decrease.

Figure 3 (A) shows the probabilities before the increase and Figure 3 (B) shows the probabilities after the increase has been applied to the RTU arrival rate.

VI. CONCLUSION

In this paper we have proposed an analytical framework that provides information about the survivability of a SCADA system. To the best of our knowledge, the framework uses a novel technique to determine service and system status.
based on performance metrics alone. We demonstrated that by using a combination of performance metrics and service interdependence information we can probabilistically evaluate the survivability of SCADA systems. In addition to that, we have shown that the model is flexible enough to replace the analytical parts by runtime data from live systems and to generate the Bayesian network automatically.

Information about survivability is crucial for SCADA system operators and security administrators that sometimes experience degraded performance from attacks such as DDoS (Distributed Denial of Service) or DoS (Denial of Service), which have different arrival rates than regular SCADA communication arrival rates.

Some may argue that this analytical model conditions does not hold in a real world. Therefore as future work we intend to validate this model by comparing it with a simulation to then determine how good the model is.

ACKNOWLEDGMENT

This work has been partially supported by ARC Linkage Grant LP100100404 and ECR Grant of second author.

REFERENCES