Abstract—Cyber threats in Supervisory Control and Data Acquisition (SCADA) systems have the potential to render physical damage and jeopardize power system operation, safety and stability. SCADA systems were originally designed with little consideration of escalating cyber threats and hence the problem of how to develop robust intrusion detection technologies to tailor the requirements of SCADA is an emerging topic and a big challenge. This paper proposes a stateful Intrusion Detection System (IDS) using a Deep Packet Inspection (DPI) method to improve the cyber-security of SCADA systems using the IEC 60870-5-104 protocol which is tailored for basic telecontrol communications. The proposed stateful protocol analysis approach is presented that is designed specifically for the IEC 60870-5-104 protocol. Finally, the novel intrusion detection approach are implemented and validated.

Index Terms--SCADA, Cyber-security, Intrusion detection, IEC 60870-5-104.

I. INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) systems have long played a critical role in power system operation and communications. The increasing standardization and interconnection of SCADA systems in Smart Grids potentially widens the prospect of cyber attacks from malicious sources. Furthermore, cyber-security aspects have not generally been considered during the design phase of most SCADA networks. Therefore, SCADA systems could be compromised by malicious attackers or disgruntled employees via unauthorized access at vulnerable points. Such intrusion has the potential to render simple or elaborate attacks which may jeopardize the system operation and which may ultimately lead to severe physical damage (e.g., the first-ever sophisticated control system malware, Stuxnet [1]). The marriage of new cyber technology to traditional control systems creates an environment where cyber and physical assets interact in ways never envisaged or planned for in most power control systems. Protecting SCADA systems from cyber threats is therefore a pertinent topic and of immediate relevance to modern power systems and smarter grids.

Currently, a number of open international standards exist in SCADA systems of the electrical power, such as Distributed Network Protocol Version 3 (DNP3), IEC 60870-5 series, and IEC 61850. For example, the IEC 60870-5-104 transmission protocol [2] presents network access for IEC 60870-5-101 [3] based on Transmission Control Protocol/Internet Protocol (TCP/IP), which can be utilized for basic telecontrol tasks in SCADA systems. However, the IEC 60870-5-104 protocol transmits messages in clear text without any authentication mechanism [4]. Furthermore, the IEC 60870-5-104 protocol is based on TCP/IP which also has cyber-security issues itself. (IEC/104 is used as the notation, instead of IEC 60870-5-104 in the remainder of the paper.)

In related research, the authors [4] have previously implemented signature-based and model-based detection approaches using Snort to improve the cyber-security of SCADA systems using the IEC/104 protocol. The research presented in this paper enhances the previous work by deriving a stateful protocol analysis approach to create a stateful Intrusion Detection System (IDS) for IEC/104 SCADA systems.

Stateful protocol analysis is a common and effective detection methodology that operates by comparing predetermined profiles of acceptable protocol behaviors for protocol states against observed activities to detect deviations and misbehaviors [5]. “Stateful” means that the IDS is able to identify and track the states of network, transport, or application protocols that have a concept of state. Stateful protocol analysis of SCADA can be performed in order to ensure proper use of protocols, compliance with protocol standards, and can be used to detect anomalous communications, which may include packet injection, replay attacks and data manipulation. The National Institute of Standards and Technology (NIST) guidelines for industrial control systems recommend stateful inspection for standard IT firewalls to evaluate packet contents at the transport layer. This enables tracking of active sessions in order to determine whether session packets are legitimate. NIST further recommends additional rule sets for SCADA applications [6].

Towards this aim, this paper proposes a model using state machines that analyze and track IEC/104 protocol state transitions in the application layer communications. This...
enables validation of communications via the protocol, for the purpose of supporting intrusion detection in the SCADA system.

II. RELATED WORK

Using IDS in SCADA networks is a relatively new concept. Some research has been conducted and applied in intrusion detection approaches for certain SCADA systems, such as signature-based, anomaly-based, model-based intrusion detection methods, as well as other SCADA-specific IDSs [4], [7]-[11]. However, research in this cross-disciplinary cyber-physical context, especially for stateful detection in SCADA systems, still has a long way to go.

Stateful protocol analysis is a typical detection methodology in IT security [5]. Y. Al-Nashif et al. [12] adopt the protocol behavior analysis approach as part of a multi-level IDS, which only describes TCP state behaviors in the transport layer by Finite State Machines (FSM). However, a recent survey has shown that approximately 80% of cyber attacks originate in the application layer [13]. H. Sengar et al. [14] present a protocol state machine based IDS for Voice over IP (VoIP), which identifies any deviation from normal protocol behaviors, and hence, could detect unknown attacks. P. Truong et al. [15] also propose an FSM based intrusion detection model for H.323-based VoIP. In [16], state machines based intrusion detection is proposed for Advanced Metering Infrastructures (AMI) in Smart Grids that includes the device-level state machine for smart meters and the application-level state machine for the American National Standards Institute (ANSI) C12.22 protocol. However, there is little published literature which rigorously considers stateful intrusion detection for SCADA protocols. To this end, by exploiting an in-depth protocol analysis and a DPI method, a stateful intrusion detection approach for IEC/104 driven SCADA systems is proposed in this paper.

III. STATEFUL PROTOCOL ANALYSIS BASED DETECTION

The stateful protocol analysis based detection uses a whitelist methodology, which means it identifies any abnormal packets that violate the predefined protocol state behaviors.

In practical Man-in-the-Middle (MITM) attack scenarios [4], it is difficult to completely block normal communication between clients and servers and replace it solely with malicious data. However, it can be relatively simple to inject a false packet into the normal communication connection (e.g., by a TCP hijacking attack). In this context, a stateful IDS must be able to identify any abnormal packet that violates the predefined state diagrams.

Finite State Machines (FSM) provide an effective methodology to describe dynamic behaviors of reactive systems [17]. A communication protocol is one of the common examples of such systems. For example, the IEC/104 protocol can be designed and implemented by an FSM. This paper proposes a novel stateful detection approach using a Detection State Machine (DSM) based on the concept of the FSM.

A. Detection State Machine

Definition 1: A Detection State Machine (DSM) D is a sixtuple:

\[
D = (S, \Sigma, G, A, T, S_A)
\]

where:

- \( S \) is a finite set of protocol states, including the start state and final state.
- \( \Sigma \) is an event alphabet of the DSM.
- \( G \) is a set of guard conditions on transitions.
- \( A \) is a finite tuple \((a_1, a_2, ..., a_n)\) of transition actions, including entry action and exit action.
- \( T \) is a transition relation: \( S \times \Sigma \times G \rightarrow (S, A) \).
- \( S_A \) is a set of alarm states \((s_{a_1}, s_{a_2}, ..., s_{a_n})\) of the DSM.

Each transition \( t \) in the transition relation set \( T (t \in T) \) is a quintuple \(<s_t, e, g, s_a, a>\), where \( s_t, s_a \in S \) are the source and the target states of the transition respectively, \( e \in \Sigma \) is an event that is an input message to the source state, \( g \in G \) is a guard condition on the transition which is a Boolean expression that must be true for the transition to be taken, and \( a \in A \) defines an action to be performed by the state machine, which is associated with the transition with entering or exiting a specific state. The basic state transition unit is shown in Fig. 1, which is the fundamental element to constitute specific state machines in practical applications.

![Figure 1. The basic state transition unit.](image)

In comparison with conventional FSMs, the presented DSM introduces the concept of a set of alarm states \( S_A \). When any protocol misbehavior occurs that deviates from the state machines based on the protocol specification, the DSM will be triggered and the state will transfer to an alarm state \( s_{a} (s_{a} \in S_A) \).

B. Proposed Stateful IDS for IEC/104

Following an in-depth analysis of the behaviors of the IEC/104 protocol, a stateful IDS is proposed for intrusion detection in IEC/104 SCADA systems. The main idea is to develop and deploy the stateful IDS to monitor the communication traffic between the IEC/104 client and the server. The stateful IDS uses DFM not only to describe the important and normal protocol behaviors in the form of state transitions, but also to detect protocol misbehaviors in the form of alarm states, as shown in Fig. 2. The detection function is performed by state transition analysis of the DFM. In the detection state diagram, the current state together with the captured and parsed IEC/104 packet will determine the normal state transition (update or keep the current state) or the abnormal alarm state. As a whitelist methodology, the DFM based IDS can detect deviations from normal state transitions, and hence, detect unknown attacks.
In Fig. 2, the yellow rectangles represent normal states which are defined according to TCP connection states, APCI information, and the states of the client, as described in Table I. The states named A_1 – A_8 represent the alarm states. The lines with arrows stand for transitions. There are two kinds of events in Fig. 2: packet-capturing events and time events, as shown in Table II. The packet-capturing events are described based on the arrival of packets. For example, an APDU_I event means arrival of an I format APDU packet. A time event will occur when a timer is overtime. For example, an After t_2 event is triggered after timer 2 is overtime.

In terms of actions in Fig. 2, Set_flag means setting a flag for a guard condition; Set_timer1, Set_timer2 and Set_timer3 stand for setting timer 1, timer 2 and timer 3 for corresponding time event, respectively; Update_state is to update a state. The guard conditions are explained in Table III. The constants referred to in the proposed DSM for IEC/104 are described in Table IV.

### TABLE I. THE NORMAL STATE TABLE

<table>
<thead>
<tr>
<th>No.</th>
<th>State name</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Disconnected</td>
<td>No TCP connection between a client and a server</td>
</tr>
<tr>
<td>2</td>
<td>TCP_Connected</td>
<td>A TCP connection has been established between a client and a server</td>
</tr>
<tr>
<td>3</td>
<td>STARTDT</td>
<td>Start data transfer</td>
</tr>
<tr>
<td>4</td>
<td>Client_Tx_APDU_I</td>
<td>A client sends an I format APDU</td>
</tr>
<tr>
<td>5</td>
<td>Client_Rx_APDU_I</td>
<td>A client receives an I format APDU</td>
</tr>
<tr>
<td>6</td>
<td>Client_Tx_APDU_S</td>
<td>A client sends an S format APDU</td>
</tr>
<tr>
<td>7</td>
<td>Client_Rx_APDU_S</td>
<td>A client receives an S format APDU</td>
</tr>
<tr>
<td>8</td>
<td>TESTER</td>
<td>Periodical connection test procedure</td>
</tr>
<tr>
<td>9</td>
<td>STOPDT</td>
<td>Stop data transfer</td>
</tr>
</tbody>
</table>

### Table II. THE EVENT TABLE

<table>
<thead>
<tr>
<th>Event name</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>APDU_I</td>
<td>The arrival of an I format APDU packet</td>
</tr>
<tr>
<td>APDU_S</td>
<td>The arrival of an S format APDU packet</td>
</tr>
<tr>
<td>After t_1_s3</td>
<td>Time event, after t_1_s3 is overtime</td>
</tr>
<tr>
<td>After t_1_s8</td>
<td>Time event, after t_1_s8 is overtime</td>
</tr>
<tr>
<td>After t_1_s9</td>
<td>Time event, after t_1_s9 is overtime</td>
</tr>
<tr>
<td>After t_2_s2</td>
<td>Time event, after t_2_s2 is overtime</td>
</tr>
<tr>
<td>After t_3_s2</td>
<td>Time event, after t_3_s2 is overtime</td>
</tr>
<tr>
<td>After t_3_s3</td>
<td>Time event, after t_3_s3 is overtime</td>
</tr>
<tr>
<td>After t_3_s6</td>
<td>Time event, after t_3_s6 is overtime</td>
</tr>
<tr>
<td>I_RC_RA</td>
<td>The arrival of a remote command or remote adjustment packet</td>
</tr>
<tr>
<td>TCP_connection</td>
<td>TCP three-way handshake</td>
</tr>
<tr>
<td>U_STARTDT_act</td>
<td>The arrival of a START act packet</td>
</tr>
<tr>
<td>U_STOPDT_con</td>
<td>The arrival of a STOPDT con packet</td>
</tr>
<tr>
<td>U_TESTER_act</td>
<td>The arrival of a TESTER act packet</td>
</tr>
<tr>
<td>U_TESTER_con</td>
<td>The arrival of a TESTER con packet</td>
</tr>
</tbody>
</table>

### Table III. THE GUARD CONDITION TABLE

<table>
<thead>
<tr>
<th>Guard condition</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowDir == C2S</td>
<td>The flow direction is from a client to a server</td>
</tr>
<tr>
<td>flowDir == S2C</td>
<td>The flow direction is from a server to a client</td>
</tr>
<tr>
<td>Flag_U_STARTDT_con==0</td>
<td>No receiving a STARTDT con packet</td>
</tr>
<tr>
<td>Flag_U_TESTER_con==0</td>
<td>No receiving a TESTER con packet</td>
</tr>
<tr>
<td>t_3_s2 &gt; T3</td>
<td>Timer t_3 of the state 2 is overtime</td>
</tr>
<tr>
<td>t_3_s3 &gt; T3</td>
<td>Timer t_3 of the state 3 is overtime</td>
</tr>
<tr>
<td>t_3_s6 &gt; T3</td>
<td>Timer t_3 of the state 6 is overtime</td>
</tr>
<tr>
<td>t_3_s7 &gt; T3</td>
<td>Timer t_3 of the state 7 is overtime</td>
</tr>
</tbody>
</table>
The variables used in Fig. 2 are shown as follows: \texttt{Flag\_U\_STARTDT\_con} or \texttt{Flag\_U\_TESTER\_con} is a flag of capturing a \texttt{STARTDT} con or \texttt{TESTER} con packet, respectively; \texttt{flowDir} and \texttt{flowID} mean the direction and the identifier of a flow, respectively; \texttt{tm\_sn} is timer tm of the state \( n \), for example, \( t1\_s3 \) means timer \( t1 \) of the state 3; \( t2 \) stands for timer \( t2 \).

IV. IMPLEMENTATION

Due to the requirements of the DSM, it is not easily feasible to implement using Snort. The proposed stateful IDS is therefore implemented using the Internet Traffic and Content Analysis (ITACA) tool. ITACA [18] is a software platform for traffic sniffing and real-time IP network analysis which has been developed by the Center for Secure Information Technologies (CSIT) at Queen’s University Belfast. The extendable and flexible analysis tool enables the implementation of plug-ins to perform specific tasks, e.g., intrusion detection. The detailed modules and functions of the ITACA architecture are described in [18]. In this paper, the Stateful Protocol Analysis (SPA) module of the stateful IDS plug-in is developed in C/C++ using the ITACA platform, as shown in Fig. 3.

The detailed implementation steps are as follows.

1) The ITACA core captures, extract, parse and analyse the raw IEC/104 packets in order to provide all possible information for the stateful IDS plug-in.

2) The DSM, as presented in Section III and described in Fig. 2, is implemented within a new stateful IDS plug-in.

3) Based on each captured IEC/104 packet and the current state read from the state memory, the DSM will determine whether the state transition is normal or abnormal. If it is abnormal, the stateful detection module will generate an alert notification and record the detection result in the log file.

For example, when the current state is \texttt{TCP\_Connected}, the DSM will execute, as illustrated by part of the DSM program shown in Fig. 4.

```c
switch (CurrentState) {
  case TCP_Connected:
    if((packet->payload == U\_STARTDT\_act)&&(packet-
        >flowDir == C2S))
      StateMemory.state = STARTDT;
      t1_s3 = packet->packet_time;
    else if((packet->payload == I\_RC\_RA)&&(packet-
        ->flowDir == C2S))
      StateMemory.state = Client\_Tx\_APDU\_I;
    else if ((packet->payload == U\_TESTER\_act) &&
              (packet->flowDir == C2S) &&
              (t3_s2 > T3))
      StateMemory.state = TESTER;
      t1_s8 = packet->packet_time;
    else
      alert();

    break;
}
```

Figure 4. A portion of the pseudo codes of the stateful detection module.

First, the current state (\texttt{TCP\_Connected}) is read from the state memory (\texttt{StateMemory.state}).

Second, the DSM executes switch statements to determine the state transition according to the arrived packet and the current state. If the captured packet is a \texttt{START} act packet (\( \text{packet->payload} = \text{U\_STARTDT\_act} \)) and the flow direction is from the client to the server (\( \text{packet->flowDir} = \text{C2S} \)), the state memory is updated to the state \texttt{3} (\texttt{STARTDT}) and meanwhile the timer \( t1 \) of the state \texttt{3} is set. When the arrived packet is a remote command or remote adjustment packet (\( \text{packet->payload} = \text{I\_RC\_RA} \)) in the control direction (\( \text{packet->flowDir} = \text{C2S} \)), the updated state is \texttt{Client\_Tx\_APDU\_I}. If a \texttt{TESTER} act packet (\( \text{U\_TESTER\_act} \)) in the control direction arrives and the timer \( t3 \) of the state \texttt{2} is overtime (\( t3\_s2 > T3 \)), the next state is the state \texttt{8} (\texttt{TESTER}) and the timer \( t1 \) of the state \texttt{8} is set.

Finally, other misbehavior packets will call the alert function that generates and records the alarm messages.

In all, 8 stateful detection rules for IEC/104 are implemented in the proposed stateful IDS.

V. EXPERIMENTAL RESULTS

A. Experiment

An experimental process was developed as follows.
1) The normal IEC/104 traffic was obtained from the commercial source (a mirrored SCADA test-bed in a power company). Abnormal packets were generated by modifying the captured data or by injecting new malicious packets in the pre-captured Packet Capture (PCAP) file.

2) The PCAP file was read by ITACA core that extracts and interprets all available information for the SCADA-IDS plug-in. The IEC/104 packets were monitored and detected by the developed stateful IDS plug-in, as mentioned in Section III.B.

3) The detection results were displayed and recorded into a log file.

B. Results

In the experiment, there were 116497 packets with 28 abnormal packets in the PCAP file, and wherein the number of abnormal packets violating SPA was 28. It is apparent from the experimental results that the proposed stateful IDS effectively identifies all the abnormal data with zero false positive for the given deterministic rules.

The message format in the log file is defined referring to RFC 3164 as follows:

<SEVERITY> TIMESTAMP DEVICE_NAME DEVICE_TYPE ALERT_TYPE EVENT_DESCRIPTION SRC_IP SRC_PORT DST_IP DST_PORT

In this case SEVERITY represents alert severity which is described by numerical code, e.g., 0, 1 and 2 stand for HIGH, MEDIUM and LOW, respectively. The TIMESTAMP field is the local time and is in the format of "YYYY-MM-DD HH:MM:SS". DEVICE_NAME means the name or IP address of specific security device. DEVICE_TYPE is the type of the security device, e.g., IDS. ALERT_TYPE represents alert event type which is violated such as SPA. EVENT_DESCRIPTION describes the detailed information of specific security event. SRC_IP, SRC_PORT, DST_IP and DST_PORT are source IP address, source port, destination IP address and destination port, respectively.

As described in Fig. 5, the SCADA-IDS alert messages show that abnormal packets in the states TCP_Connected, STARTDT and Client_Tx_APDU_I are identified, which violate the detection rules SPA-1, SPA-2 and SPA-3, respectively. Other detection results have also been recorded in the log file using the unified format.

![Figure 5. Alert examples in the log file](http://www.digitalbond.com)

VI. CONCLUSION

Previous research in this area has mainly investigated Modbus or DNP3 protocols [7]-[9]. To the best of the authors’ knowledge, this paper is the first to propose a stateful IDS for IEC 60870-5-104 SCADA networks. A novel detection state machine is proposed and applied to provide the stateful IDS for SCADA networks using IEC/104. The proposed intrusion detection tool, implemented by the ITACA platform, can be applied to monitor and detect IEC/104 traffic in SCADA systems. Stateful protocol analysis is recommended by NIST to detect anomalous behaviors where SCADA protocols are used. This research has directly addressed these challenges through the presented stateful IDS. This work contributes significantly to improving the cyber-security of SCADA systems that use the IEC/104 protocol.

REFERENCES