|  |
| --- |
| **Project Title:**  **Control logic based cyber attacks in Industrial control systems**  We have a data set collected from industrial water plant for 11 days. By taking the data as input, learning the control logic of the entire water system is the first step. Once we have the control logic, the next step is to design successful attacks automatically using machine learning techniques to on water system using learned control logic. |
| **Notes 11/10/23:**  Most studies assume insider access…. More useful to model unknown system and generate attacks against it? Use know system here to make methodology/ program to learn system then probe behaviour to work out attacks.  As a separate task, use know attack data to create CGAN that finds attack state closest to current state- take in current values and type of attack wanted ( i.e. tank you want to overfill) and produce attack plan to reach goal in way which closest matches normal behaviour.  Bayesian Network / correlation coefficients can map relation of system components to each other. Analysis of behaviour can show duty cycle for components and max/ min values etc.  Automated probing of system can manipulate these values to find error states ( successful attacks).  These error states can be used to train CGAN.  Initially do this with extracted values but ideally modify to decode MODBUS communications and generate attack packets.  Focus on Level 1 between PLC and ADC/DAC’s.  This works as MODBUS includes slave address…. All other slaves get signal but ignore non-relevant. Potentially knowing the response time of system components would allow the attack slave to respond before real component…. Race conditions… man on the side attack.  Sridhars report says their attacks were at level 1 ( between PLC and SCADA) and involved faking sensor value packets. This suggests the PLC do not do much standalone processing- decisions are made by scada and passed to PLC to implement.  <https://ieeexplore.ieee.org/document/7423145> has 3 axis diagram for type of attack…. System performance, component integrity/ life and property of system such as pH of output water. |
| **Abstract**  This paper investigates the use of machine learning techniques to model the operation of, then create attacks against Cyber Physical Systems (CPS). CPS have a physical aspect, which measure and influence the real world through sensors and actuators, and a cyber aspect, which provide control and monitoring functions.  CPS are traditionally used in industrial process but with the advent of the Internet of Things (IoT) they now proliferate modern life- from home automation systems and autonomous vehicles through to the Critical National Infrastructure such as power grids and public transport networks. As such, under- standing the threats posed to these system by emerging technologies is a key area of research for both the public and private sector.  This paper focuses on Secure Water Treatment testbed (SWaT) as variants of these are present in the majority of industrial process and include a range of components which are common to other processes.  Initially a cyber attack methodology was applied to identify aspects of the SWaT which are vulnerable when an ‘insider’ access level is assumed- that is, full knowledge of, and access to all aspects of the system ( including any anomaly detection systems).  A cyber-attack can have multiple facets, investigated here are:  • False anomalies to provoke unnecessary maintenance- DROP AS TOO EASY??  • Spoofing control signals to increase wear/ impact system operation  • Hiding/ spoofing error messages to allow components to be damaged |
|  |
| **Introduction**  The water treatment Cyber Physical System is a collection of sensors, actuators and control logic which autonomously control the industrial process. In the Secure Water Treatment testbed (SWaT) there are 6 distinct stages which perform the process of converting the incoming dirty water into the output of drinkable water. Each of the stages is controlled by a dedicated Programmable Logic Controller (PLC) that communicates with the PLCs in the other stages in order to ensure the whole system works harmoniously.  The inputs to the PLC are sensors which produce continuous signals (analogous to physical properties of the system) such as water level, fluid flow rates and temperature and discrete signals which show the state of system components ( if a valve is open or closed for example).  The PLC monitors these incoming signals and applies pre-programmed logic to generate output signals which operate various switches and actuators. The effect of this is that the PLC is a standalone micro-controller which is able to control a stage of the industrial process autonomously.  For example, a PLC receiving a low water level signal as an input will send a control signal to a pump or valve (an actuator) which will operate until the input signal (water level signal) is again within range.  The internal logic of the PLC is designed to account for all possible system states and includes as maximum & minimum values for parameters such as temperature, pressure and flow rate. These parameters form the basis of the rule checker that acts as anomaly detector- flagging behaviour which indicsates a system fault or attempted attack. Should a rule be broken the PLC will take appropriate action and raise an alert.  These PLC’s communicate with each other in order to pass or request data pertinent to their own stage of the process e.g. request a faster flow of water or indicate a batch is ready to move to the next stage.  The signals between the PLC and the input and output components are said to be at Level 0 of the OSI model as they either analogue or digital signals in a range suitable for operating switches etc. These signals are typically continuous voltage signals which are converted into a current range between 4 20 mA to avoid transmission losses. This current signal is processed by an external Analogue to Digital Converter (ADC) or Digital to Analogue Converter (DAC) as appropriate as the PLC uses binary encoded values at Level 1 of the OSI model.  There are multiple communications protocol’s used for this PLC – PLC communication, these are often proprietary to a particular manufacturer but are usually based on the Ethernet/IP industrial Ethernet standard. This configuration is known as Industrial Internet of Things iIOT or Industry 4.0.  The updating of the logic within the PLC’s is accomplished via a Human Machine Interface which communicates on the same Ethernet/IP ring network as the PLC’s. The monitoring of process parameters is performed by Supervisory Control and Data Acquisition (SCADA) system. An Historian will record all system values for performance analysis and fault finding.  Machine Learning is already applied to many CPS in the form of anomaly detectors. These are usually Support Vector Machine or Deep Neural Networks which have been trained to classify normal and anomalous behaviors (having been trained on data from normal operation and simulated attacks). These anomaly detectors may reside on the SCADA system or possible on a stand-alone processor at PLC level- an example of edge computing ( reference Arduino site).  A cyber vulnerability methodology was used to identify vulnerabilities suitable for machine learning techniques to be applied to. The study assumes an ‘insider’ level of access where the attack has full access  to the system- this allows manipulation of the raw signals at level 0 through to the high-level TCP/IP packets which carry the control and monitoring messages.  A SVM Classifier was trained on the sample data to act as a baseline then a Generative Adversarial Network (GAN) was created to generate data for the attack. A GAN consists of a Classifier and a Generator where the error from the classifier is used to improve the generator until it can successfully produce data which can fool the classifier.  This method was applied to the individual Stages of the SWaT system and to system as a whole. |
|  |
| NEW CONTENT 2/10/23 |
|  |
| What is the end goal- focussing on system wear……. Stress component to maximum degree but don’t set off any alarms. i.e. turn pump on and off very frequently by having system operate just below max values. This would mean pump and inlet valve are constantly operating but level sensor alarms is not triggered.  Manually workout what are critical system values and therefore what triggers alarms.  Model needs to have recurrent type inputs so that effect of preceding actions are taken into account. This means small modifications to the system are possible that give big effects.  This suggests only tampering with signals at level 0 or those packets passed between PLC’s i.e. not the ‘alarm’ signals which show abnormal operation??? ( Check Data to see if these are present).  Attcks need to be coordinated and persistent- e.g. low level signal to trigger pump operation then ok level signal to trigger outflow valve. Identifying system states which create most activity in system is best for aging attacks- find the balance point where the system is constantly trying to reach a state and maintain this overshooting/ instability— This could be whole focus of project….. try genetic algorithms and GAN’s to produce data to meet this condition.  A methodology for this could then be used for other parts of the system.  Maybe workout which parts of the system are highly correlated first….. check it applies to insider understanding. See how these perform against the anomaly detectors I make using SVM and Neural Networks.  This can then be modelled in Minicps and potentially in the testbed.  NOTES:  Training a GAN to produce attack data means system parameters need to be in appropriate state before attack signal can be used to cause anomaly.  As such the GAN output is effectively a system description- in order to apply the attack the system has to be moved into the appropriate prior state. Therefore each stage has to be modelled and understood, including having functions to set values. i.e. to get a particular tank level the input & output has to be manipulated until the value is reached.  Each of these pre-cycles have to be implemented below the threshold of the anomaly detectors.  The most basic is to calculate the rules present in the PLC which should be achievable by extracting min and max values in the normal data. Potentially there are more complex interactions that are not obvious- each stage needs to be isolated and a model created which includes it’s inputs & outputs to other plc’s.  This is likely to consist of actuator/ sensor relationships such as a valve being opened causes a linear increase in tank level. If a pump to take water from this tank is also opened then (assuming constant pump speed) the fill rate will slow/ stop/ reverse.  Isolating activites to ascertain true relationships could be done after basic corelation measurements. This would have the effect of producing an explainable model of the system which may be less evident in neural network.  Iterating through all data with the dependent variable being swapped each time would allow the system influences to be mapped accurately. This could be used to produce a visualisation of the system interactions to aid attack planning.  A probability model of each system’s state in regrards to eaches effect on the dependent variable would help model the system. Possibly this would need to include raising, falling and static states for the variable.  This could be implemented using a regression model to predict each dependent variable then the weights for each parameter should indicate which are relevant. The data could also have a delay added to indicate hysteriesis in the system….. does adding ‘x’ delay increase model accuracy? Likely as data is instantaneous system values so will show acutuator values even if they are yet to have an effect on the system. Could this be used to avoid anomaly detector? Turn off actuator so that it show’s low when alarm stae is reached?  Each step towards the system state could be checked against the Anomaly Detector to sure it’s not trying to make too large a step etc.  What info is passed between PLC’s? Presumably actual level control etc. is decided by local PLC and implemented at Level 0, as such setting of pre-levels will need to be done by spoofing Level 1/0 which would surely involve manipulating the logic within the PLC.  Also, presumably the GAN outputs random system states to meet the required attack. This wouldn’t necessarily account for the current system state so could require large changes when there was an attack state thar required less change.  The gan could just produce an attack state table from which a solution is picked but if current system state could be incorporated this would be more efficient. Would a genetic model be more appropriate? Could the system heuristics mean the model can find a more efficient attack?  Can the latent variable in CGAN not be totally random? What happens if system state is also added as a second condition?  Dress report has factors such a colour, shape concatenated to input vector. Wasserstein thing is about most efficient way to move from one distribution to another ( Earth Mover something). This seems to be the way as it should fine minimal cost to move from current system state to one that meets the attack state. |
|  |
| SWaT Data |
| Characteristics of dataset (SWaT.A1\_Dec 2015)  11 days of continuous operation: 7 under normal operation and 4 days with attack scenarios  Collected network traffic & all the values obtained from all the 51 sensors and actuators  Data labelled according to normal and abnormal behaviors.  Attack Scenarios: Derived through the attack models developed by our research team. The attack model considers the intent space of a CPS as an attack model. 41 attacks were launched during the 4 days and are described in the PDF. |
| 2015 seems best as it has both normal and attack scenarios. Data is available for network traffic and physical values. PDF’s included for list of attacks.  Version 0 includes 30 mins where water tank is drained for maintenance ( outside of normal operation) Version 1 has this 30 minutes removed so is to be used.  V1 Normal and V0 Attack are merged, new column to indicate which dataset is added. |
|  |
|  |
|  |