Cyber-Physical System Discovery – Reverse Engineering Physical Processes

Alexander Winnicki
Hamburg University of Technology
Hamburg, Germany

Marina Krotofil
Honeywell Industrial Cyber Security Lab
Duluth, GA 30097, USA

Dieter Gollmann
Hamburg University of Technology
Hamburg, Germany

ABSTRACT

Successful cyber attacks against cyber-physical systems require expert knowledge about the dynamic behaviour of the underlying physical process. Therefore, obtaining the relevant information is a crucial part during attack preparation. Previous work has shown manual acquisition of knowledge about process dynamics to be prohibitively laborious. This paper presents first insights into semi-automated process-aware system discovery that goes beyond IT-related trivia, and focuses on the physical core of a system.

Keywords

Cyber-physical systems, process discovery, signal correlation

1. INTRODUCTION

The stages of cyber attacks against Cyber-Physical Systems (CPS) are divided into access, discovery, control, damage, and cleanup, where adversaries have to perform different tasks at each stage to accomplish specific goals [13]. In order to be successful at the control stage, attackers need to sufficiently understand the underlying physics of the targeted system such that they are able to control the physical core process themselves, and bring it into a desired state. Assuming the presence of compromisable IT and embedded devices inside a CPS, which allow adversaries to gain remote access to an operating control system, we examine the capabilities of attackers at the control stage.

Targeted system discovery against CPS has already been observed in the wild in 2013, when the Havex malware infected numerous industrial sites via trojanized installers of software packages, and started gathering information about industrial control and field devices [8]. Although easily obtainable, this type of information does not suffice to properly control a physical process. Moreover, static information related to process physics, e.g. process flow-sheets, is also simple to obtain but does not account for the dynamic behaviour of the underlying physics either. Previous work has shown that acquiring adequate knowledge about dynamic process behaviour without any a-priori information is a difficult task due to the large scale and complex interdependence of CPS subcomponents [13]. Certain aspects of process behaviour might be undocumented or previously unseen, and therefore unknown even to process operators.

In order to stay undetected attackers need to avoid triggering operational and safety constraints when controlling the process. However, simply knowing set thresholds is not sufficient, as the physics of a core process can behave in highly non-linear manners, easily causing alarms when not expected. Additionally, process behaviour outside of the operational envelope of the deployed control structure may be undefined and unpredictable. From the attacker's perspective understanding the desired state of a physical process that achieves certain attack goals, and knowing how to reach that state are two distinct problems. Both demand extended knowledge about the underlying process dynamics. However, the complexity of a CPS may make manual acquisition of information about dynamic process behaviour prohibitively laborious. Automated procedures for identifying behavioral plant models by means of control and automation engineering methods (see e.g. [7, 2]) may not be suitable for an attacker who has penetrated the target but has to work with limited computational resources and may not be able to exfiltrate large amounts of data.

To address this issue, we propose a generic and semi-automated approach for exhaustively cataloguing physical process dynamics by capturing sensor readings on-the-fly. First, we propose a generic attacking approach inspired by controller tuning to allow process probing without detection, and to account for a lack of a-priori knowledge about process constraints. Next, we present a lightweight algorithm for correlation-based clustering of sensor readings based on captured process responses induced by probing. Finally, we verify our results with a derived metric to ensure adequate precision, and propose a visual representation of the results to enable a convenient and expressive summary.

Our paper is structured as follows: Section 2 presents the simulation framework that we use as a testbed for our approach and outlines why the adopted framework suits our use case. Section 3 provides a detailed explanation of our proposed approach. Section 4 discusses a selected set of representative results, Section 5 summarizes our work.

2. SIMULATION FRAMEWORK

We test our approach with the Tennessee Eastman (TE) test problem, originally offered for study by the Eastman Chemical Company [12], and further extended with a Matlab Simulink model by Ricker [9]. The test problem is de-

ASIA CCS '17, April 2–6, 2017, Abu Dhabi, United Arab Emirates.

DOI: http://dx.doi.org/10.1145/3055186.3055195

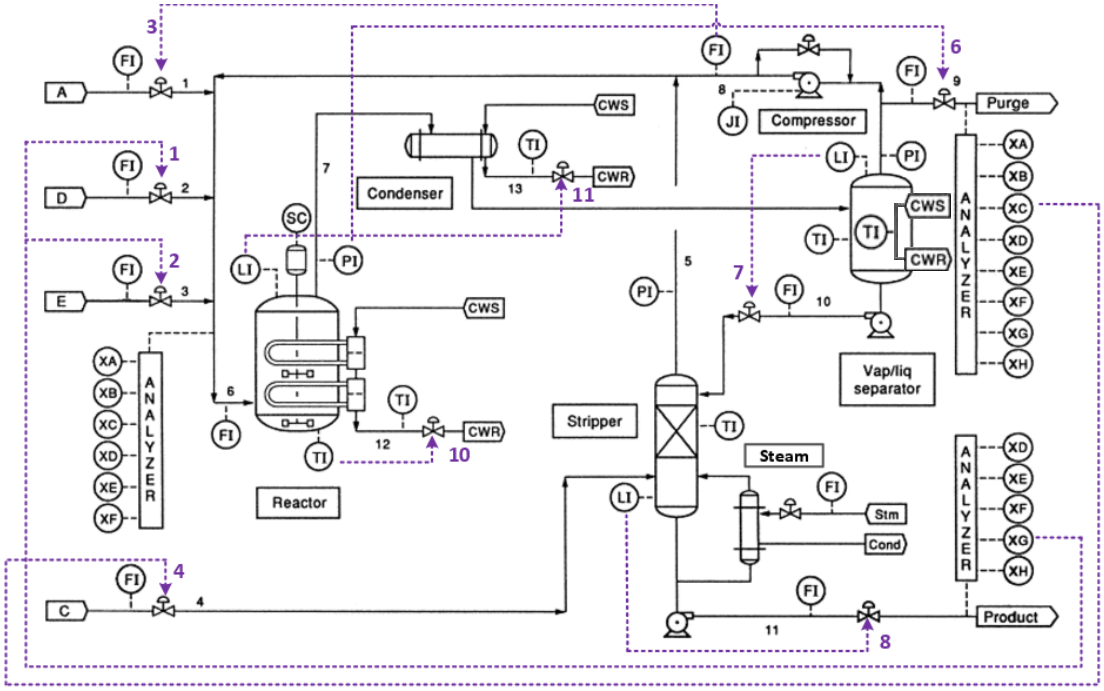

Figure 1: Tennessee Eastman plantwide test problem

rived from an actual industrial process where components, system dynamics and conditions were adjusted to protect the intellectual property of the Eastman Chemical Company. Nevertheless, the problem provides a generic and realistic simulation environment of a chemical industrial CPS.

The TE test problem is suited particularly well for testing our approach because it is not bound to specific process particularities, e.g. the chemical components involved are unknown. We can thus verify our approach in terms of its generic nature without needing knowledge about chemical aspects of the process. Hence, our results can be detached from this particular problem, and be interpreted in the context of a variety of distinct CPS.

Our goal is to reverse engineer the behaviour of a physical process based on the dynamics of process data, without relying on the semantic meaning of that data. For instance, we identify sensor signals that are correlated due to similar impulse responses, and not because the underlying chemistry would imply their correlation. Therefore, our approach accounts for the situation of adversaries facing a process without a priori knowledge regarding its behavior. We investigate whether it is possible to derive dynamic behavioral information by observing process responses to crafted impulses, and whether this knowledge is sufficient to facilitate success at the control stage of cyber attacks.

The Matlab model of the TE test problem is implemented as a C-based Mex S-function with a Simulink model. The default simulation time is seventy-two hours, with a sampling frequency of one hundred measurements per hour. Start-up and shut-down conditions of the system are not simulated; the execution starts with pre-defined base values instead. There are twelve controllers (called XMV), and forty-one sensors (called XMEAS) providing measurements from the simulated process. All of the measurements contain Gaussian noise with a standard deviation typical of the particular measurement type. Furthermore, there are twenty different disturbance modes (called IDV) which can be turned on and off selectively. The output of each simulation is a data matrix where each column contains measurements of a distinct sensor, and each row is a step in time, i.e. 36 simulated seconds. Consequently a default simulation provides 7201 measurements per sensor, yielding a 7201x41-sized matrix. Simulation times can be extended or shortened to adjust the number of generated sensor measurements.

A diagram of the process flow and control structure is shown in Figure 1. The process has five major operational units: a reactor, a product condenser, a vapour-liquid separator, a recycle compressor, and a product stripper. The reactor feed rates are partially controlled by outcomes of the analysis of chemical products. In reality, such an analysis is a non-automated offline procedure, which can take up to half an hour. Therefore a real plant could not apply such a control strategy, because reactor feed rates usually have to meet stringent real-time requirements.

To simulate cyber attacks against the TE process we use an extension to the original Simulink model provided in [4], which supports different attack modes against the process. The extension provides different modes of data integrity and Denial of Service (DoS) attacks against most of the given controllers, sensors and actuators. Time, duration and periodicity of attacks can be specified for each simulation run.

3. APPROACH

We now present our approach for reverse engineering a physical process from captured sensor measurements. The procedures we propose are semi-automated: While the attacker has to actively perform process probing for which we give a systematic framework, the collection and processing of sensor data is performed by our algorithms on-the-fly.

3.1 Process Probing

An attacker facing an unknown process is challenged by a lack of knowledge about the tuning and responses of control loops. Hence, trying to catalogue the dynamic behaviour of the process by sending crafted impulses becomes a non-trivial task. Ultimately, the attacker has to apply trial-and-error process-probing, where errors must not trigger alarms, detection and scuppering of the attacker's goals.

There is no general solution to the problem of how to specifically attack a given control loop with unknown responses, such that no alarms are raised. However, control loop tuning procedures face similar problems when orchestrating control loops across an entire plant to optimize and balance a process [10, 11]. Therefore we derive a unified attacking approach inspired by control loop tuning procedures such that we obtain a generic probing approach that we apply to all available control loops. The purpose of our unified approach is to provide a starting point for probing the process in a safe manner, i.e. without causing alarms or safety shut-downs.

We start by analysing and quantifying the noise level of a sensor signal. To account for the lightweight nature of our approach, we estimate the noise level with the following assumptions. Without exterior influence, a signal fluctuates around a specific mean value, and the fluctuations have approximately the same magnitude both in positive and negative direction. We further assume that the noise level of a signal is constant across different signal means, i.e. enforced set-points. This is consistent with our observations of the behavior of the TE process. Equation 1 depicts our estimation of a signal's noise level based on these assumptions. The values $signal_{min}$ and $signal_{max}$ constitute the minimum and maximum observed values for the given signal during normal operation, and the resulting noise level represents the maximum expected magnitude of noise in positive or negative direction.

$$noise_{level} = \frac{signal_{max} - signal_{min}}{2} \tag{1}$$

$$noise_{factor} = \frac{|normal_{mean} - deviated_{mean}|}{noise_{level}} \tag{2}$$

Next, we induce a step change of a few percent in the corresponding controller. Once the sensor signal settles at the new set-point, we compute the magnitude of change regarding its previous state, and quantify it as a factor of the noise level. We call the resulting value noise factor, defined in Equation 2. Mean values of the previously normal state and the induced deviated state are denoted by $normal_{mean}$ and $deviated_{mean}$. As suggested in [10], we check whether the resulting signal deviation exceeds the significance threshold of five times the signal's noise level. If not, we adjust the controller's set-point by another few percent, and let the signal settle again. We do so until the deviation leads to a noise factor greater than five. The controller set-point that creates such a deviation is the starting attack value that we use to further analyse process responses.

Figure 2 shows the sensor signal of a reactor feed stream with a step change induced in the corresponding controller at $t = 20$ hours, resulting in a deviation with a magnitude of approximately five times the signal's noise level. The attack parameter that creates this response is recorded for further process probing. Note that valve dynamics are not modeled in TE simulation framework. Therefore the new set-point is reached almost instantaneously. A real world physical process would require a certain time to reach its new state. The change in sensor signal of interest can be detected in an automated way, e.g. by a light-weight CUSUM algorithm as shown in our previous work [3].

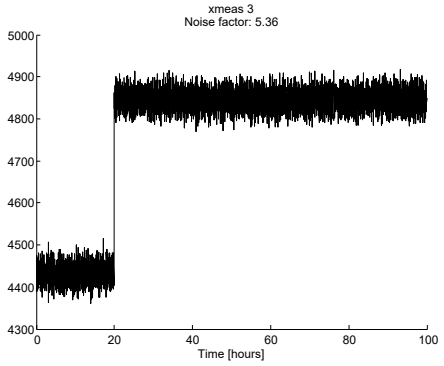

Figure 2: Deviation with desired noise factor

Importantly, our attacking approach provides the adversary with a generic way of silently probing individual control loops. It is by no means a complete solution to the problem of manipulating a process without a priori knowledge. Thus, while for most of the controllers of the TE process it suffices to induce a step change of around 10% to achieve the desired noise factor, one controller requires a change of more than 80%. Furthermore, control loops related to temperature and cooling water flows are so sensitive that it is impossible to achieve a noise factor of less than a hundred. Since the magnitude of process response is not known to the attacker beforehand, it represents a significant challenge and an uncertainty for the attacker.

As a result out of the twelve available controllers in the TE model, three cannot be attacked as they have constant settings, and four are related to temperature parameters and therefore too sensitive. We are thus left with five attackable controllers to which our unified approach of silently probing the process can be applied. Four of these controllers regulate feed streams, and one is responsible for the reactor purge rate. Table 1 depicts the attack parameters we derive for these controllers using our approach.

Another challenge of probing a process is uncertainty in terms of time. To establish our unified attacking approach we first used step attacks, i.e. persistent changes of a controller's set-point. However, for modeling a complete range of process behaviors the attacker cannot solely rely on step attacks. Instead, interval attacks with defined timing parameters should be applied, such that after a successful probing action the process is left to recover before a new attack can be started. The process recovery phase is important for discovering hidden responses, e.g. post-reactions, that occur after the attack is over.

Process response and recovery times cannot be guessed. Therefore, for every probing action the attacker needs a certain amount of time to observe a deviation, and to let the process recover into its original state. Moreover, if the attacker cannot observe (detect) any response for a certain amount of time, it is uncertain whether the attack was in-

Controller	Sensor	Mean	Attack Value	Unit	Step Change	Noise Factor
XMV(1) (D Feed Flow)	XMEAS(2)	63.00	68.75	kg/h	9.1%	5.41
XMV(2) (E Feed Flow)	XMEAS(3)	53.13	58.00	kg/h	9.2%	5.36
XMV(3) (A Feed Flow)	XMEAS(1)	26.11	29.00	kscmh	11.1%	5.07
XMV(4) (A and C Feed Flow)	XMEAS(4)	60.57	65.30	kscmh	7.8%	4.26
XMV(6) (Purge Valve)	XMEAS(10)	25.74	47.00	%	82.6%	5.14

Table 1: Attack parameters for controllers

significant and there will be no response, whether there was a small but imperceptible response, or whether the response is yet to occur. Finding timing parameters of the attacks and size of process observation windows requires empirical experimentation on the live process.

When working with a simulation framework, process probing can be manually stopped and restarted at any point in time, thus practically allowing to ignore recovery times of the process. However, a real process under attack cannot be simply turned off and restarted. In our work we would like to be as realistic as possible while simulating the attacker's situation. Therefore we extend the attack parameters found by our approach with timing parameters such that we can perform alarm-safe interval attacks with fixed start and end times. Effective timing parameters can be found using the same strategy, i.e. by starting with small intervals and extending them until a noticeable deviation is observed. This procedure is in line with the fact that shorter attack intervals are advantageous for the attacker regarding effort, risk, and recovery time.

In summary, process probing by means of attacks on controller set points is a slow process associated with a large number of uncertainties. Such uncertainties are highly disadvantageous to the attacker as they increase probability of accidental errors and detection.

3.2 Sensor Clustering

We now present our algorithmic approach to the problem of reverse engineering a physical process, and a time-dependent measure of correlation as a means of verification. Our approach distinguishes two phases of signal processing: approximation and clustering (of correlated signals).

In our initial approach we used an adaption and modification of the Swinging Door Algorithm (SDA), which iteratively reduces a data series to a set of linear regression lines [1, 6]. First presented in 1990, the Swinging Door trending algorithm is a widely used lossy algorithm in the industrial process control. It is a linear fitting algorithm which compresses the data by reducing the amount of data points that need to be stored. SDA is widely used by many real-time databases such as PI and others because of its high efficiency and high compression rate.

After approximating the dynamic behavior of sensor signal with a set of lines, we subsequently analyse the resulting lines with regard to their slopes. Further, we group signals with similar counts of rising and falling slopes into clusters. However, the SDA is very sensitive to data outliers because even after down-sampling every sample has a strong impact on the resulting regression lines. Therefore we could not achieve a satisfying trade-off between the amount of remaining significant signal behavior and random noise. Moreover, the resulting correlation clusters did not distinguish different types of correlation, e.g. both linear and inverse linear correlations could occur in the same cluster. Uncorrelated signals which have accidentally similar slope counts could be also be grouped in the same cluster, which occurred quite often in our simulations. Lastly, the SDA approach does not provide any additional information about the signal analysed.

Unable to achieve satisfactory results with the SDA algorithm, we developed our own Sliding Mean Algorithm (SMA). In essence, SMA iteratively derives clusters of correlated sensors without computing the mathematical notion of correlation. Determining mathematical correlation for the vast number of occurring sensor measurements would be computationally intensive. Instead, we approximate correlations between sensors by comparing mean values across different measurement series with the use of sliding mean intervals. SMA is immune to data outliers due to the inherent mean computations. Signals that are clustered together have the same type of correlation because they are identified as having a defined response pattern, and not merely similar slope counts. Additional signal information, e.g. deviation magnitudes, recovery times, post-reactions, is returned as collateral information obtained by the mean computations. As we show in Section 4, our approach is lightweight and produces adequately accurate results.

A detailed assessment and comparison of both algorithms would be out of the page count of this paper. Instead we focus on the Sliding Mean Algorithm, which gives the most accurate and useful results.

3.2.1 Core Algorithm

Our Sliding Mean Algorithm starts by assessing the baseline behaviour of a signal. This is done by computing a mean value for the time window before the attack. Attack time and duration are input parameters to our algorithm.We assume that attacker knows when and for how long to attack victim system, and is able to correctly place their actions on the timeline of system events.

Assuming that an attack starts at $t = t_a$, our algorithm computes an average value based on the time interval $t = t_a - s$ until $t = t_a$, where s is the number of samples we average. We use a value of $s = 100$ as it proves to be a good trade-off between compression and precision, and for consistency. The resulting mean is our reference value for normal behaviour of the given signal. Furthermore, we determine minimum and maximum values contained in the signal before the attack.

Next, we turn our attention to the attack interval starting at $t = t_a$. We split the interval into two halves, and examine each one separately. We first compute a mean value for the first half, and compare the result to the previously determined reference value. Note that depending on the duration of the attack, the data window used for averaging can be larger or smaller than the one we use for the reference value. This is a necessary adaptation, as we cannot rely on attack intervals being always perfectly dividable by our averaging interval.

Reference comparison is done by calculating the difference between the reference value and the mean value. Our algorithm further requires error bounds to serve as confidence intervals, meaning that they have to be exceeded so that we acknowledge a reaction. If the computed difference is greater than the error then the sign of the difference determines the type of reaction, which can be either positive or negative. Otherwise we assume that there is no reaction. Consequently, we distinguish three possible responses for the first half of the attack: *positive*, *negative*, and *none*.

Scale and range of sensor signals differs greatly among process measurements. Therefore we require each sensor signal to have individual scale-adjusted error bounds. To achieve this goal we first down-sample all the data, then determine the minimum and maximum value in each signal series, compute the absolute distance between them, and divide the result by two. As this is done for every signal, the final outcome constitutes a vector of scale-adjusted and noise-reduced error values for all sensors. Each error value represents by how much a signal may vary in both positive and negative directions, such that the deviation is considered insignificant.

Both data down-sampling and error training are lightweight procedures. Our down-sampling is based on computing average values, where only two values need to be stored: the sum and the number of samples that the sum contains. Each average computation further requires s summations and one division, per averaging interval. On the other hand, error training requires a standard procedure of finding minimum and maximum values in a data series, further one subtraction and one division.

The second half of the attack serves for verifying the response found in the first half of the sensor signal, for several reasons. We have experimentally found that many signals have large dead-time, meaning that they manifest delayed responses. This means that there might be no discernible reaction during the first half of the attack. On the contrary, responses may also manifest themselves immediately but only for a very short time. In this case, there would be no reaction during the second half. Splitting the attack interval into two distinct parts allows us to detect and consider these properties of signal behavior in our response classification and clustering decisions.

Furthermore, while some signals show very clear positive or negative reactions to certain impulses, for other signals it is hard to decide whether the response is increasing or decreasing, e.g. when a signal shows oscillations with changing magnitudes. In that case we regard the oscillation slope with the greatest magnitude to represent the nature of the response, e.g. the signal oscillates but there is an increasing trend, or the strongest oscillation has a positive sign, then we say that the response is positive. Doing so makes sense also for the attacker; even if the response is theoretically different in a chemical interpretation, the strongest oscillations should be the focus as they can hit process constraints and trigger potential alarms.

If we find no response during the first half of the attack we cannot say yet whether the signal does not respond at all, or whether the response is delayed. However, if there is no response during the second half either then we assume that the signal does not react to the given impulse. In the case there eventually is a response during the second half we argue that the second half represents the actual reaction. We apply the same argumentation for the case when there is a response during the first half but not during the second, where we regard the first half as representative. Additionally, if each of the attack halves contains a distinct response we consider the one with the greatest magnitude as being representative of the actual response.

If no reaction is found in both halves of the attack then we use the previously determined minimum and maximum reference values of the signal, and compare them to minimum and maximum values occurring during the attack. If the latter exceed the reference values in magnitude, we assume that highly frequent oscillations occur during the attack. Our assumption is based on the fact that computing mean values over high frequency oscillations results in a straight line showing no reaction. However the presence of values of greater magnitude during the attack contradicts the idea of no reaction, and rather suggests that the reaction is masked by our averaging computations (signal smoothing). While being aware of the limitations of this error-prone way of detecting or estimating oscillations, we would still like to emphasize how a few simple computations combined with logically relating differing results can suggest the existence of a response that would otherwise require extensive data processing to be found.

We further treat oscillations as a secondary response property to avoid pattern explosion that would occur if we regarded all derived properties of signal responses as equally important. Having secondary properties also allows a simple and clean distinction of additional sub-categories, while avoiding an exponential increase of possible patterns through combinations of all defined properties. This supports our idea of a lightweight approach.

Finally, we determine the maximum percentage deviation from reference of the analysed signal during the attack, and move our time window forward step by step until we reach the end of the data. For each time window we compute a mean value and compare it to our reference value to determine whether the signal settles back at its original state or transitions into a new one. In the former case we say that the signal is *resilient* as it is able to recover from the attack and return to its normal state. When the signal reaches and remains in a different state even after the attack ends then we say it is *stabilizing*, i.e. the signal does not properly recover but at least it is able to settle in a new state.

3.2.2 Post-Reactions

In certain cases signals do not show any significant responses during an attack, however they have strong post-reactions, i.e. there is a clear process disturbance when the attack stops. We are able to detect these post-reactions by computing a mean value for the entire time, from the beginning of the attack to the end of the data, and comparing it to reference values. We further use minimum and maximum values to underline the conclusion drawn from this mean comparison. Additionally, we determine the maximum percentage deviation from reference of the signal caused by the post-reaction.

This simple approach to finding post-reactions is possible since we first conclude whether a signal has a reaction during the attack or not. If it does, we classify it as either resilient or stabilizing with corresponding information about the deviation. However if there is no reaction during the attack, it could mean that there is no reaction at all, or that the response is delayed until the attack is over. In that case post-reactions are easily spotted using only one mean com-

parison, and verifying them by further assessing minimum and maximum values, and comparing them to reference values as well.

If a signal has a reaction during the attack and also after it, then we regard its first reaction as the definitive one, because it occurs as a direct response to the attack, whereas the post-reaction is likely to solely manifest process recovery. Signals which have post-reactions only are much more interesting as to why a variable manifests reactive recovery while having no significant reaction in the first place.

Since we regard oscillations as secondary response properties, we treat post-reaction in the same way because, as argued before, our main focus is on direct responses. Post-reactions constitute an additional sub-class of reactions that are interesting if there is no significant direct response beforehand.

It is important to note that although this detection of post-reactions is effective in most cases, it is still error-prone, e.g. unpredictable disturbances not caused by attacks can obscure the results and easily lead to wrong conclusions. However, a significant amount of useful information can be obtained by applying such relatively simple and imprecise methods. Especially because we theoretically intend to have Programmable Logic Controller (PLC) execute our algorithms, which means that we have to work with strictly limited resources.

3.2.3 Recovery Times

Lastly, we use another sliding mean window to compute recovery times for resilient signals, i.e. the amount of time these signals require to return to their normal states. For an attacker this can be crucial information in the context of avoiding long-lasting abnormal post-attack states, which can potentially trigger safety interlocks. Having determined that a signal is resilient, we apply a sliding mean window starting at the time of the attack, and attempt to find a predefined number of consecutively connected time windows which match the reference state, i.e. the differences of means are inside the confidence interval. If the algorithm finds time windows that do not match the reference state it increases a counter for non-matching states. If this counter hits a pre-defined limit, the counter of previously found matching states is reset. When the counter for matching states cannot be hit, the numerical value of recovery time stays zero, indicating that it was not possible to determine a signal's recovery time.

In other words, if the algorithm finds a reference-matching state it keeps on moving the mean window a predefined number of times to verify that the signal settles indeed, and that the matching state is not simply an accidental finding. If it finds a non-matching state during this verification period it does not immediately assume that verification fails, but instead tries to assess whether the non-matching state extends over a longer period of time, or whether it is a random short disturbance. Extended disturbances can be part of a response, and indicate that the signal has not settled yet.

In most cases, this procedure allows the algorithm to accurately determine recovery times of signals when all possible post-reactions or post-disturbances have ceased, while also minimizing the biasing impact of short and random data outliers. However, in certain situations the point of recovery time cannot always be precisely determined either because of strong post-reactions or due to high signal variability. Moreover, it is not always clear how much time is generally required for the entire process to fully recover, especially when examining one signal at a time. The challenge of the attacker is to empirically derive counter variables which work best for the process under analysis.

Determining efficacious values for our state counters is a trade-off between the amount of inaccurately computed recovery times and the inability to compute recovery times at all. Low counter values result in short verification periods, potentially causing the algorithm to assume that signals have already reached their reference states, while they are actually still in recovery oscillations. High counter values imply longer verification periods, which in certain cases can lead to the inability to hit the counter for reference-matching states because there is not enough data left, or due to randomly extended noise constantly resetting the counter for reference-matching states. We use a counter value of 1500 for matching states, and 1250 for non-matching states. Additionally, we enforce a simulation time of 100 hours and perform attacks early into simulation to minimize the inability to compute recovery times due to lacking data. However, we noticed that increased simulation time may cause some resilient signals to be classified as stabilizing because of random attack-unrelated process fluctuations occurring towards the end of the simulation time.

3.2.4 Summary

The major advantages of our sliding mean approach are simple implementation and resource-friendly fast execution. Computing mean values is not a resource-demanding operation, further storing and comparing these values does not require extensive resources either. As shown in more detail in Section 4, a lot of information can be extracted by using this simple approach. On the other hand, the underlying simplicity makes the approach sensitive and error-prone to certain disturbances or possibly complex process responses, as it can only detect and classify pre-defined response patterns. If unknown patterns occur they will be misclassified, just like noisy signals can erroneously match defined patterns by chance. However, we have found the majority of relevant occurring patterns in our simulation framework, and we are able to detect and classify them with a high rate of correctness.

3.3 Cluster Verification

Pearson's correlation coefficient is a common measure of linear dependence between two stochastic variables. Its mathematical definition is given in Equation 3, where X and Y represent stochastic variables, μ_X and μ_Y are the corresponding expected values, and $\sigma_X \times \sigma_Y$ is the product of the standard deviations of X and Y.

$$\rho_{X,Y} = \frac{E[(X - \mu_X) \times (Y - \mu_Y)]}{\sigma_X \times \sigma_Y} \tag{3}$$

The coefficient takes values between 1 and -1, where the former case represents perfect linear dependence, and the latter indicates perfect inverse linear dependence. Correlation between two stochastic variables becomes weaker as the coefficient approaches 0, until there is no linear dependence once the coefficient equals 0. However, this does not mean that there is no dependence at all, as the coefficient only detects linear relationships. There might, for instance, be a quadratic dependence between the two examined variables.

However, the Pearson correlation coefficient as described above is not useful for determining linear dependence between time series of sensor signals. This is because the coefficient is a single number which cannot capture the dynamic behavior of sensor measurements with adequate granularity. For instance, it does not allow to assess how correlation between two signals evolves over time. Moreover, two signals might show exactly the same reaction to some impulse but not at the same time, i.e. one of the signals reacts immediately while the other has a delayed reaction. Pearson's correlation coefficient would fail to properly identify this linear relationship due to the shift in time. In fact, the vast majority of cyber-physical sensor signals cannot react simultaneously as the physics they measure are locally separated, such that state changes propagate with time delay.

To address this shortcoming of the Pearson coefficient we propose a time-windowed approach which defines a sliding interval on the given data series, and where the Pearson coefficient is computed in each of these intervals individually. The result is a series of time-dependent Pearson correlation coefficients, altogether representing the development of linear relationship between two signals over time. We use this approach to verify that our resulting signal clusters are indeed correlated.

4. EXPERIMENTAL RESULTS

In the following we present and discuss the clustering capabilities of sliding mean algorithms in the context of chosen controller attacks. We further present a plant-wide visualization technique to capture obtained results in a convenient form. Our analysis assumes that the attacker has gained access to the control system of the TE process, and that our previously presented Sliding Mean Algorithm has been injected into the system, e.g. in the form of malicious code running on a PLC(s).

In the process of developing and verifying both of our approaches we performed approximately 500 distinct simulations generating over 2000 plots. Due to space limitation, we include only a selected set of representative results. We chose the controller responsible for the reactor purge valve (XMV(6)). The purge valve is used to release chemicals from the reactor, primarily to control reactor pressure. Attacks on this controller has plant-wide impact with a wide range of different types of process responses and therefore is representative.

We perform a simulation of 100 hours and start an integrity attack at $t = 20$ hours for a duration of 20 hours. The normal mean value of the purge valve state is 25.74 %, but to achieve our desired noise factor in the controlled signal, which is the purge rate, we require an attack value of 47.00 %. Unlike suggested for controller tuning, this high step change of almost twice the normal value is necessary because the mass flow of the purge valve is comparatively small compared to the reactor mass flow of 1476.0 kgmol/h. Therefore only a significant step change is able to achieve a noticeable deviation in the controlled signal.

Figure 3 depicts the purge rate during a simulation. The black signal represents raw sensor data, which is marked red while the attack is active, and the yellow line represents the same signal down-sampled with an averaging interval of 100 samples. We include this smoothed signal in our plots for visualization only, because in some cases it is helpful for distinguishing signal behavior from noise.

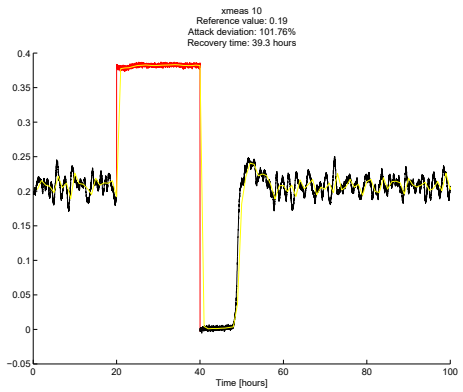

Figure 3: Spoofed purge rate

As mentioned earlier, the dynamic behavior of valves is not modeled in TE simulation model. The purge rate thus jumps to a new state immediately at $t = 20$ hours, and drops to a less than normal value right when the attack is over at $t = 40$ hours. Otherwise the behavior of sensor signals matches reality. Hence, the new induced state of the purge rate is not represented by a straight line but contains fluctuations just as before the attack. This is because the attack only changes the set-point of the purge valve. The corresponding signal of the purge rate is continuously controlled and adjusted to match this new set-point, and it naturally contains fluctuations caused by the underlying physics.

We also observe that there is no proportional relation between the purge valve and the purge rate, because the maximum deviation of the purge rate caused by the attack is 101.76 %, although we increased the purge valve set-point by 82.6 %. Furthermore, once the attack is over and the authentic set-point of the purge valve is no longer overwritten by our integrity attack, the control system notices the highly abnormal state of the purge valve and reacts by completely closing it at $t = 40$ hours. This manifests itself in the purge rate dropping to zero at that time. At about $t = 60$ hours the purge rate returns to its normal state.

Note how the increase of the purge rate around $t = 50$ hours does not manifest itself as a perfectly straight line, like at the beginning our attack. This is another hint at the underlying feedback mechanism of the responsible control loop, which adjusts the purge rate continuously by adjusting the purge valve. In contrast, the malicious set-point induced by our attack is a sudden and external influence on the process, which does not involve the feedback mechanism.

Our sliding mean approach correctly classifies the purge rate as a resilient signal with a positive response. This means that the purge rate increases as a result of the attack, but is able to return to its normal state after recovery. The determined recovery time of 39.3 hours is also very accurate as can be seen in Figure 3. After $t = 50$ hours the purge rate is near its reference state, and behaves accordingly. However, it is still in recovery, as there is a small but discernible overshoot caused by the gradual opening of the valve after it has been nearly closed for a while. Our approach is able to distinguish this recovery behaviour from the previously determined reference state.

Our Sliding Mean Algorithm determines three main clusters for the described attack against the purge valve. These

clusters are formed by resilient signals with a positive response, resilient signals with a negative response, and resilient signals with no response. One signal with a negative response is misclassified as stabilizing instead of resilient.

4.1 Resilient Positive Cluster

Figure 4 and 5 show signals from the cluster of resilient signals with a positive response, as determined by our Sliding Mean Algorithm. All of the contained signals increase in reaction to the attack and return to their original state once the attack is over. Most recovery times are correctly computed and fluctuate around 30 hours. Noisy signals tend to be challenging for accurate computation of recovery times. E.g. the recovery time of signal XMEAS(3) was determined by our algorithm as being $t = 53.5$ hours, a few hours too short, because the normal state was reached around $t = 60$ hours.

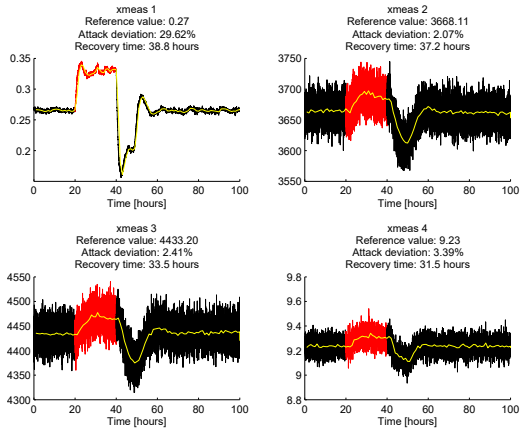

Figure 4: Resilient signals with positive response

Also, the overshoot reactions of the system are likely to obfuscate recovery times. For instance, recovery time of signal XMEAS(23) in 5 was estimated as 30.2 hours instead of actual 35 hours. The slight but discernible negative overshoot reaction at $t = 50$ hours misleads our algorithm into believing that the reference state is reached, because averaging over a period with abnormally high and abnormally low measurements produces a reference-matching state by chance. The fact that we use error bounds as confidence intervals to estimate whether a reference state is reached or not only enforces this effect, because slight deviations are ignored inside these error bounds.

Interestingly, signals XMEAS{1, 2, 3, 4} in Figure 4 are all measurements of reactor feed streams of different components. Since our attack on the purge valve increases the purge rate of the reactor, the system needs to compensate for the loss of chemical components from the reactor. Therefore it makes sense that these signals all have a positive response. Moreover, the response of XMEAS(1) is highly similar to the behaviour of XMEAS(10) in Figure 3, the previously discussed purge rate. At first sight, this could indicate that XMEAS(1) represents the dominating feed component, because a reaction almost identical to the increase of the purge rate could mean that XMEAS(1) compensates for most of the loss. However, considering the comparatively small volume flow rate of XMEAS(1), this cannot be true. Clearly, XMEAS{2, 3} are dominating feed streams in terms of mass.

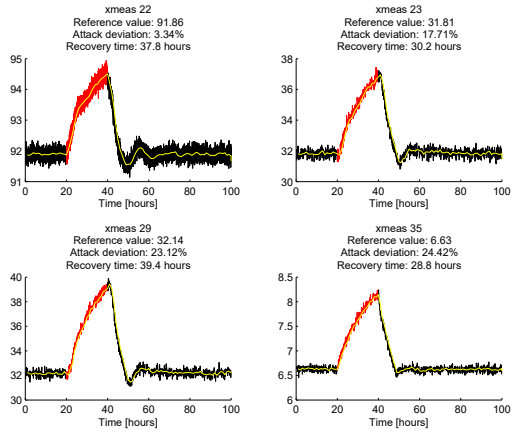

Figure 5: Resilient signals with positive response

We give these chemical details of the process only for verification purposes, to show that the behaviour of the system makes sense and that we can detect it. However, our goal is to present an abstract approach for detecting correlated signal clusters which is detached from the underlying chemistry of the process. Our approach does not verify whether the clusters it finds are meaningful in terms of the involved chemistry. However, by detaching our approach from specific process chemistry, and also from our specific test bed, we are able to provide a more generic approach that can be applied to different processes with minimal modification.

4.2 Resilient Negative Cluster

Signals from the next cluster found by our Sliding Mean Algorithm are shown in Figures 6, 7 and 8. All involved signals are resilient and have a negative response to the attack against the purge valve. While some signals have a clearly visible decreasing reaction, e.g. XMEAS(7), noisy signals like XMEAS(5) have a less distinguishable deviation. Nevertheless, our algorithm is able to correctly classify these noisy signals.

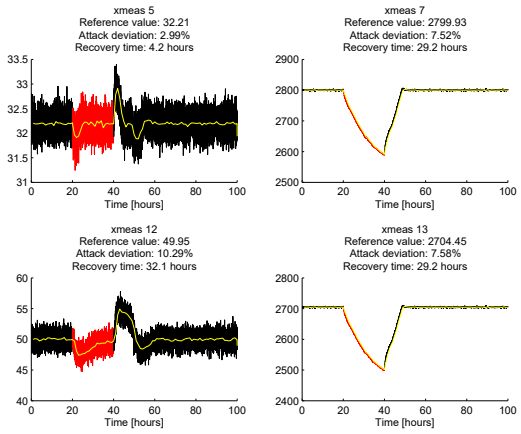

Figure 6: Resilient signals with negative response

Signals XMEAS{5, 12, 15} further manifest post-reactions, i.e. once the attack is over there is an additional increase and subsequent decrease reaction. What all these signals have

in common is that they recover quickly and return to their normal state even before the attack is over. This indicates that these signals are rather independent from the attacked variable, in this case the purge valve, as they are able to recover on their own before the attack is over.

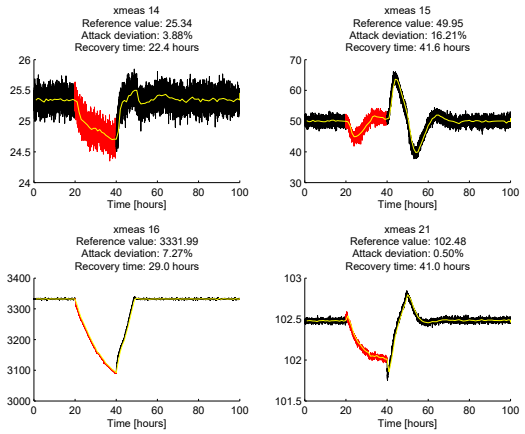

Figure 7: Resilient signals with negative response

However, once the attack is over and the directly affected signals start to recover, the recovery process has a plant-wide impact and also affects independent signals. Therefore we can observe additional oscillations in certain loops which we call post-reactions. For instance, XMEAS(15) represents the stripper level. As can be seen, this signal recovers on its own before the attack is over. However, due to a plant-wide post-attack compensation for the loss of chemical component caused by the attack, plant recovery has an oscillating effect on the stripper level.

The occurrence of post-reactions is a risk to the attacker, who cannot predict beforehand whether they will occur or not. A crucial alarm-sensitive signal might be completely unaffected by a crafted impulse, however the following recovery of affected signals can potentially disturb the entire system. If signals have stringent operational and safety constraints, even small deviations can trigger alarms. For instance, note how the maximum deviation of the post-reaction for XMEAS(15) is greater than the maximum deviation during the attack. This means that the system's recovery process has a greater impact on the stripper level than the performed attack against the purge valve. The same holds true for XMEAS(12), which is the product separator level. Also, signal XMEAS(5), which is the recycle flow, has a similar post-reaction.

Apart from the reactor, stripper and separator are the only tank-like sub-components of the TE process. It makes sense that these three signals show strong post-reactions for the performed attack. Our attack against the purge valve results in high reagent loss, which the control system compensates for by drastically increasing reactor feed streams. This measure quickly restores normal states for stripper, separator, and recycle flow. The physical nature of these components implies that they have a significant amount of chemical components contained in them all the time, which reduces the impact of our attack in the first place.

Therefore we observe short and insignificant direct reactions in XMEAS{5, 12, 15}. However, the control system remains at abnormally high feed streams until the attack

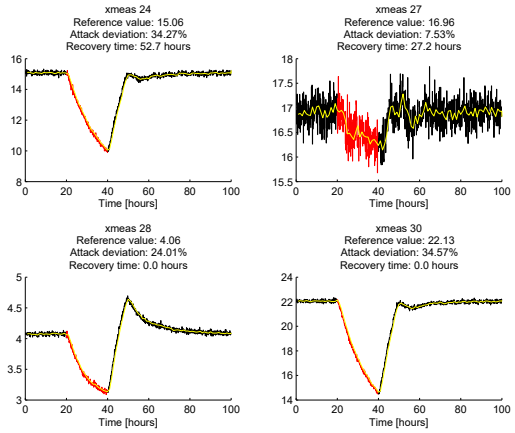

Figure 8: Resilient signals with negative response

is over. Once it is over, the purge valve is temporarily closed, effectively setting the purge rate to zero. This is the time where the still abnormally high feed streams result in accumulation of the physical components in the stripper, separator, and recycle flow, manifesting themselves as post-reactions.

Figure 9 depicts the time-windowed Pearson correlation for XMEAS{12, 15}. The relationship is not strong, but there is a clear correlation. Note, the correlation during the post-reaction is significantly higher than during the attack.

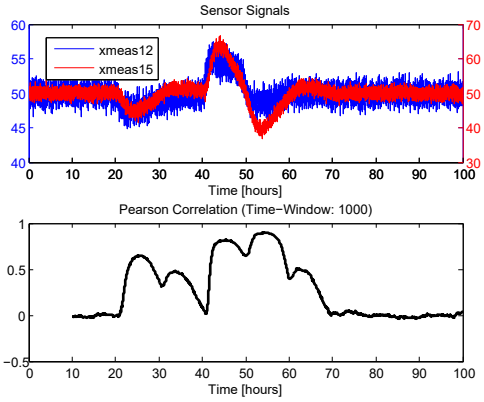

Figure 9: Time-windowed Pearson correlation

Although post-reactions are important to detect, in our approach we classify the mentioned signals as resilient with a negative response, meaning that they react with a decrease and return to their original state after the attack. The cataloging of post-reactions is omitted for a reason. As described before, distinguishing too many process response properties would result in a high number of different clusters with fewer correlated signals in each. Our algorithm acknowledges post-reactions only if a signal has no reaction during the attack, as direct reactions are our primary focus.

Also note how the cluster contains signals of highly divergent scales, similarly to the previously discussed cluster with positive responses. Our algorithm is able to identify these signals as similar due to our implementation of scale-

adjusted and signal-specific error bounds. Since we require these error bounds for our sliding mean approach to function, when we present results of the Sliding Mean Algorithm it means that we perform two simulation instances. Firstly, we run a simulation without any attacks or modifications. After the simulation is over, we compute our vector of error bounds based on the output of the TE model. Next, we run a simulation affected by an attack, and apply the Sliding Mean Algorithm to the generated output data, using our error bounds for decision making. In real life, the attacker would perform error bounds estimation, attack(s) and process response analysis in sequential fashion.

4.3 Resilient Inactive Cluster

The last cluster found by our Sliding Mean Algorithm is depicted in Figure 10 and 11. It consists of sensor signals that do not react to the performed attack. However, signals XMEAS{6, 8, 17} are misclassified due to their small scale and high noise level. As described before, signal noise can obscure the results of our Sliding Mean Algorithm.

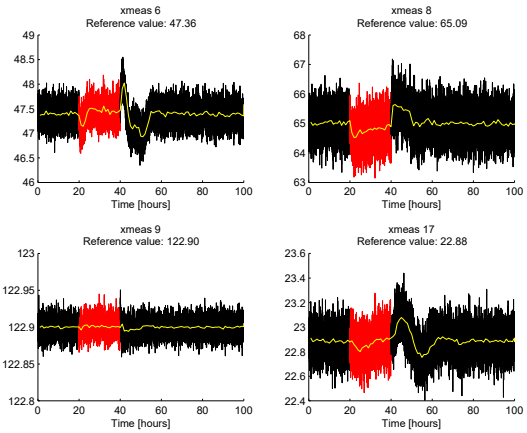

Figure 10: Resilient signals with no response

Figure 12 shows the aforementioned cluster of stabilizing signals created by a misclassified signal XMEAS(34). When performing attacks against controllers, we expect all signals to react in a resilient manner due to the fast process recovery that follows these attacks.

The reaction of signal XMEAS(34) is highly similar to XMEAS(28) in Figure 8, where the latter is correctly classified as being resilient with a negative response. While XMEAS(28) represents the reactor feed stream for component F, XMEAS(34) measures the purge rate for this component. These signals should be in the same cluster. However, by taking a closer look at the graph of XMEAS(34) it can be seen that, at the end of the simulation, the signal is in a slightly higher state than it was before the attack had started. Our algorithm erroneously interprets this behavior as XMEAS(34) not returning to its original state. Considering the similarity between XMEAS{28, 34}, the fact that they are in separate clusters shows how signal noise can have obscuring effects on our Sliding Mean Algorithm.

4.4 Plant-Wide Response Visualization

Figure 13 depicts visualization of the performed attack and classification results for the entire system. Such visualization can be done, e.g., over P&ID or process flow dia-

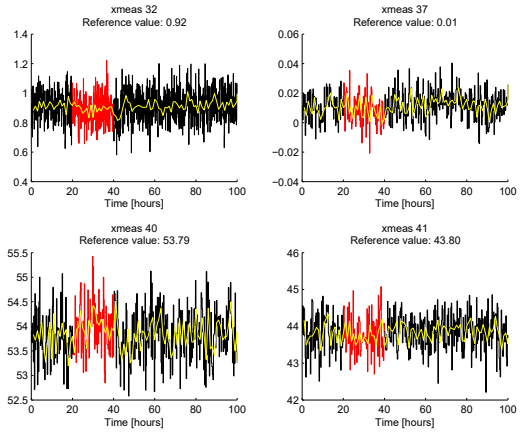

Figure 11: Resilient signals with no response

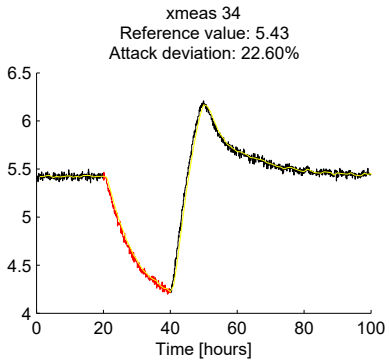

Figure 12: Stabilizing signal with negative response

grams. Our idea for the plant-wide response visualization is to give a condensed overview of the system's response without providing extensive details. Therefore, we do not distinguish resilient and stabilizing signals. Furthermore, deviation magnitudes and recovery times are not included. We also regard oscillating reactions and post-reactions as equivalent to a no response. The latter accounts for the logic of our Sliding Mean Algorithm, since oscillations and post-reactions are secondary properties of signals that have no directly detectable reaction in the first place. In summary, Figure 13 distinguishes three different response patterns: *positive*, *negative*, and *no response*.

Let's look at the results. Note the consistent behaviour of the involved chemical components denoted by XA, XB, etc. The deviations that affect these compnents are identical for the reactor feed rates and the reactor purge rate. Only the amount of components G and H contained in the final product is inconsistent with the purge rate. As a result of the attack, the amount of products G and H released through the purge valve increases, but does not change in the final product composition.

Since G and H are the two desired products of the TE process, it is of economic meaning that the increased purge rates have no influence on the final product composition, apart from decreased secondary components E and F. We

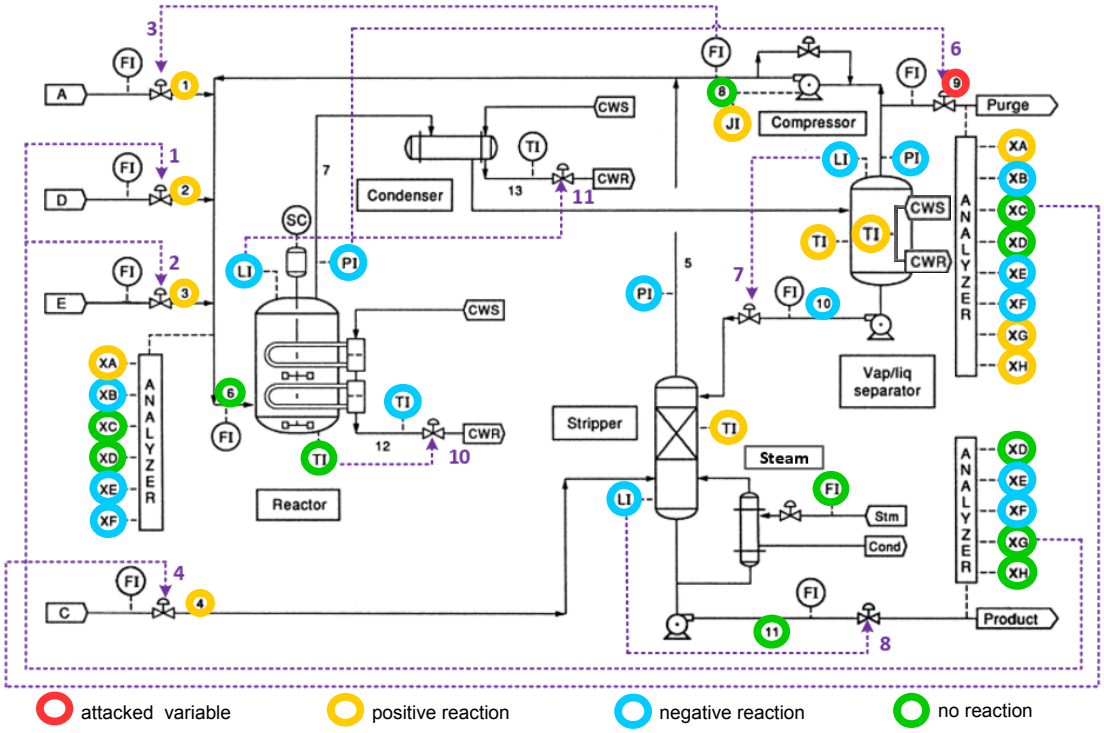

Figure 13: Plant-wide response visualization

can conclude that the performed attack against the purge valve, although of disturbing nature, has no direct impact on the process in terms of financial damage through product loss. The final product composition actually improves due to the attack, since the concentration of the secondary and unwanted components E and F decreases. However, financial damage is caused by the loss of reagents removed from the process due to the abnormally high purge rate. It is therefore crucial to measure process performance not only in terms of product purity, but also regarding the efficiency of reagent conversion. A more detailed discussion on attacks strategies on reaction rate and associated forensic analysis can be found in our previous work [13].

Simple visualization techniques of mapping process responses over P&ID (or similar) diagram, like the one we propose, can help to learn and catalogue dynamic behaviour of any physical process. An exhaustive set of identified response patterns can serve as a simplified model for preparing more sophisticated attacks. Although Figure 13 contains only a fraction of the information found by our sliding mean approach, it allows for convenient reasoning about process behaviour and even economy with regard to attacks. Moreover, our plant-wide response visualization can be of use to attackers as well as defenders, since in both cases knowledge about the underlying process is crucial for success.

In a real scenario however, the amount of possible response patterns paired with the huge number of process variables can be overwhelming both to attackers and defenders. In that regard, simplicity is a key when trying to model the behaviour of a complex physical process. Being able to extract, identify, and visually compress dominating aspects of dynamic process behaviour enables a tailored and adequately exhaustive analysis.

4.5 Attacks on Sensors

There are sixteen attackable sensors in the TE model. Attacking sensors is however substantially different from attacking controllers. Attacks against controllers directly force the system to go into a deviated state due to changed set-points of valve petitioners. In contrast, spoofed sensor measurements cause the system to assume that it is already in a deviated state, and that it has to recover the process. Such process state recovery is iterative, meaning that the valve positions are adjusted gradually with small set-point changes at each control cycle (as opposed to a single set-point change in controller attacks).

It is still possible to probe the process silently with a help of data integrity attacks on sensor signals. However, no recommendations based on control theory methods exist to support this process. One should start with small scale data integrity attacks and gradually increase them as the observed response allows (heuristic approach).

5. SUMMARY AND CONCLUSIONS

An attacker facing an unknown process is challenged by the lack of knowledge about its control loops' tuning and responses. Consequently, trying to catalogue the dynamic behavior of the process by sending crafted impulses is a non-trivial task. To account for the situation of adversaries facing an unknown process we have derived a unified attacking approach inspired by controller tuning procedures for silently probing a process to determine attack and timing parameters while keeping the risk of alarms and detection at a minimum. We have applied all of the above mentioned in terms of interval-based integrity attacks against a representative set of controllers and sensors of the TE process.

In our approach we identify clusters of correlated sensor signals using a generic algorithm that does not rely on the mathematical notion of correlation. Instead, our algorithm extracts behavioral patterns from measurement series with the help of lightweight data approximation procedures paired with estimation methods. In addition, our algorithm is detached from the underlying physics or chemistry of the process, which renders our approach applicable to a vast variety of different processes with distinct control strategies. Out of two tested algorithms, the sliding mean approach proved to be more effective and produce more convenient results, while simultaneously being simpler both in implementation and execution.

We have visualized the classification results of our Sliding Mean Algorithm on the process flow-sheet of the TE model as exemplary means of a visually convenient summary for cataloguing plant-wide process responses in the context of behavioral modelling. We have further introduced the metric of time-windowed Pearson correlation for verification purposes, as the standard notion of Pearson correlation does not provide an adequate granularity for capturing dynamic process behavior.

In summary, our work indicates that it is feasible to reverse-engineer a controlled physical process from observations of responses to crafted impulses. Even with light-weight approximative algorithms running on resource-restricted low-level field devices, an attacker is able to obtain a significant amount of information regarding dynamic process behavior, and potentially controller tuning. With additional knowledge about the underlying control structure, which is easily obtainable in the form of process flow-sheets, attackers can further make conclusions about flaws of the employed control system and prioritize attack targets (e.g. exclude sensitive controllers). The acquired knowledge enables adversaries to prepare sophisticated attacks, tailored to the individual dynamic behavior of the victim process. As a result, these attacks are more likely to be successful, and less likely to be detected in the short term. In fact, adversaries might even focus on obtaining behavioral models of physical processes without having the intention of actually attacking these processes afterwards. Instead, these models can be sold to interested third parties, effectively separating efforts and responsibilities of the discovery and control stage of cyber-physical attacks against PCS.

We would like to conclude with the encouragement to further develop process-aware cyber security mechanisms that do not solely rely on IT aspects. Once IT barriers are compromised, which Stuxnet and Havex have shown to be possible, adversaries are potentially able to take control over the physical process [5]. While the latter is a non-trivial task and requires expert knowledge, it would be negligent to assume that attackers are unable to obtain or apply the required knowledge, and thus to rely on security by obscurity in such a crucial applications as Industrial Control Systems. Instead, process discovery must be actively opposed to deprive attackers of the ability to simulate and prepare sophisticated attacks.

6. REFERENCES

[1] E. H. Bristol. Swinging Door Trending: Adaptive Trend Recording? *ISA, Paper #90-493*, 1990.

[2] A. Maier, O. Niggemann, M. Koester, C. P. Gatica. Automated Generation of Timing Models in Distributed Production Plants. In *2013 IEEE International Conference on Industrial Technology (ICIT)*, pages 1086 – 1091, 2013.

[3] M. Krotofil, A. Cárdenas, B. Manning, and J. Larsen. CPS: Driving Cyber-physical Systems to Unsafe Operating Conditions by Timing DoS Attacks on Sensor Signals. In *Proceedings of the 30th Annual Computer Security Applications Conference*, ACSAC '14, pages 146–155, 2014.

[4] M. Krotofil and A. Isakov. Damn Vulnerable Chemical Process - Tennessee Eastman. http://github.com/satejnik/DVCP-TE.

[5] R. Langner. To Kill a Centrifuge, 2013. goo.gl/uX3mJG.

[6] G. Chen, L. Li. An Optimized Algorithm for Lossy Compression of Real-Time Data. *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), Vol. 2*, 2010.

[7] A. Vodencarevic, H. Kleine Büning, O. Niggemann, A. Maier. Identifying Behavior Models for Process Plants. In *2011 IEEE 16th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–8, 2011.

[8] Symantec Security Response. Dragonfly, 2014. goo.gl/yUEn8d.

[9] N. L. Ricker. Tennessee Eastman Challenge Archive. http://depts.washington.edu/control/LARRY/TE/download.html.

[10] J. Smuts. Cohen-Coon Tuning Rules. http://blog.opticontrols.com/archives/383.

[11] J. Smuts. Ziegler-Nichols Open-Loop Tuning Rules. http://blog.opticontrols.com/archives/477.

[12] J. J. Downs, E. F. Vogel. A Plant-Wide Industrial Process Control Problem. *Computers & Chemical Engineering*, 17(3):245 – 255, 1993.

[13] D. Gollmann, P. Gurikov, A. Isakov, M. Krotofil, J. Larsen, A. Winnicki. Cyber-Physical Systems Security - Experimental Analysis of a Vinyl Acetate Monomer Plant. In *1st ACM Workshop on Cyber-Physical System Security(CPSS)*, pages 1–12, 2015.