A Case Study of a Systematic Attack Design Method for Critical Infrastructure Cyber-Physical Systems

D. Grimsman, V. Chetty, N. Woodbury, E. Vaziripour, S. Roy, D. Zappala, and S. Warnick*

Abstract— **As cyber-physical systems continue to become more prevalent in critical infrastructures, security of these systems becomes paramount. Unlike purely cyber systems, cyber-physical systems allow cyber attackers to induce physical consequences. The purpose of this paper is to design a general attack methodology for cyber-physical systems and illustrate it using a case study of the Sevier River System in Central Utah (United States). By understanding such attacks, future work can then focus on designing systems that are robust against them.**

I. Introduction

As cyber-physical systems continue to become more prevalent in critical infrastructures, security of these systems becomes paramount. Unlike purely cyber systems, cyber-physical systems allow cyber attackers to induce physical consequences. Some examples of these types of attacks include:

- The Maroochy sewage management system in Australia was attacked by a disgruntled former employee, flooding some parks and a hotel with sewage waste [1]
- A Polish teenager modified his TV remote to switch the train tracks, causing 4 wrecks and multiple injuries [2]
- The Stuxnet worm was discovered breaking equipment in Iranian nuclear plants [3]

The purpose of this paper is to design a general attack methodology for cyber-physical systems and illustrate it using a case study of the Sevier River System in Central Utah (United States). By understanding such attacks, future work can then focus on designing systems that are robust against them, for example, using the mitigation strategies discussed in [4].

II. Related Work

In recent years, attack modeling of cyber-physical systems has become an important area of research [5]. Since cyber attacks can cause physical damage, security of these systems requires that traditional cybersecurity be maintained while also designing a robust system architecture in case this security layer breaks down. Increasing security requires that more sophisticated attack models be built to guard against. Therefore, new models have been developed recently in the literature. For instance, in [6], Zhu and Basar use game theory to model how the cyber and physical components of the system interact, along with how the attacker and system administrator interact. In [7], a general attack space is presented along with a corresponding attack policy using state space models. This work differs from these two because it uses the signal structure of the system to simulate the model from the attacker point of view.

In [8] attacks are modeled from a control-theoretic perspective. While this work is similar to [8], our attack methodology looks for the link with the highest vulnerability in the signal structure, rather than the state space. The choice of using the signal structure instead allows us to represent the system at the level of abstraction seen by the attacker (More on this in Section III). Additionally, [8] has a similar river system application, however, the system model is different and leads to different kinds of attacks and vulnerabilities than shown in this work.

III. Attack Design Methodology

We present a six-step methodology for designing attacks on critical infrastructure systems. In brief, these steps are as follows:

Step 1: Define a Model Class of the System

As is the case with any strategic attack design, a model of the system is required to understand how to design an attack. The specific model class depends on the application, and each application may have several model classes from which to choose. In the case study for this paper, we use a parametrized mass-balance model with a PI controller.

Step 2: System Identification with Available Data

Once a model class has been chosen, the parameters of that model class can be assigned based on the data given. The literature is rich with system identification techniques and theories, therefore, this paper will only cover the topic to the extent that we show our system identification approach for the case study.

Step 3: Identify the Exposed Variables

From an attacker's perspective, not all system states or variables are exposed, or directly observable. Identifying which system states an attacker can access allows the model to describe the damage an attacker can do.

*D. Grimsman, V. Chetty, N. Woodbury, and S. Warnick are with the Information and Decision Laboratories, Brigham Young University, Provo, UT. E. Vaziripour and D. Zappala are with the Internet Research Laboratory, Brigham Young University, Provo, UT. S. Roy is with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA. This material is based on research sponsored by the Department of Homeland Security (DHS) Science and Technology Directorate, Homeland Security Advanced Research Projects Agency (HSARPA), Cyber Security Division (DHS S&T/HSARPA/CSD), LRBAA 12-07 via contract number HSHQDC-13-C-B0052. All material in this paper represents the position of the authors and not necessarily that of DHS.

Step 4: Model the Attack Surface

Using only the exposed variables identified in Step 3, a new model of the system can be created that represents the attacker's perspective, meaning the set of exposed variables and the dynamic relations between them. This can be done by leveraging the signal structure as represented by the dynamical structure function (DSF) [9] [10] [11].

Step 5: Analyze the System Vulnerability

Equipped with the DSF found in the previous step, we can now leverage the results in [12] and [4] to find vulnerabilities in the system. These results use the small gain theorem to find which of the links–or the dynamic relationships between manifest variables–are most sensitive to perturbations. If no links are vulnerable, then the system is completely secure from all destabilizing attacks, assuming the attacker utilizes the attack surface modeled.

Step 6: Design an Attack

When the most vulnerable link–i.e. the link that is most sensitive to perturbations–in the DSF has been established, the next step is to design a perturbation on that link that achieves the attack objectives. The specific design and implementation of this attack will vary depending on the application.

IV. The Sevier River System: A Case Study

In order to showcase the methodology laid out in the previous section, we apply it to a real-world cyber-physical system–a segment of the Sevier River System in the state of Utah.

Background: The Sevier River System

The Sevier River is an essential natural resource in Utah, USA. It is managed by the Sevier River Water Users Association (SRWUA) in collaboration with the US Bureau of Reclamation and is used extensively for irrigation. As a whole, the Sevier River irrigates over 286,600 acres of farmland (about one quarter the size of Rhode Island) in a semiarid desert region [13].

This system belongs to a class known as supervisory control and data acquisition (SCADA) systems. It has some built-in automation, including semi-automated gates and active flow sensors that communicate their values wirelessly to a central server. These values are reported publicly on the SRWUA website.

For the purpose of this paper, we focus on a segment of the Sevier River in the Lower section of the Sevier River Basin that stretches between the Yuba Reservoir and the DMAD Reservoir (see Fig. 1). The water flow leading into this portion of the river is controlled by a water commissioner through gates at the Yuba Reservoir. Each canal branching off the main river is governed by an independent canal company, which in turn services requests made by farmers needing irrigation water downstream. At each point in the river leading to a canal, a gate controls the water flow into the canal. Each gate is also monitored by a sensor measuring the water flow. The remaining water not diverted into a canal then flows into the DMAD Reservoir. The water flow along the main stretch of river is about an order of magnitude higher than that of any of the canals that branch off.

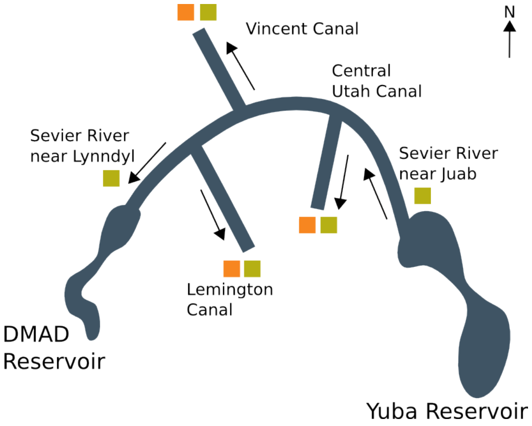

Fig. 1 Lower part of Sevier River. Green boxes show the sensors for water flow and orange boxes represent the gate height sensors. This figure is not to scale.

The water commissioner works with the canal companies to receive aggregated requests via SMS or phone calls. The water commissioner then schedules releases from dams to meet the demand represented by these requests. Most commonly, requests are made during the spring and summer months (Apr - Sep) while farmers are irrigating.

A comprehensive list of security threats to water systems like the Sevier River System has been cataloged in [14]. Also see [15] for a more detailed examination of cyber attacks on SCADA systems in general.

Step 1: Define a Model Class of the River System and the Water Commissioner

The first step of the attack design is to build a model of both the river system and of the water commissioner controlling the system.

Physics-Based Model Class of the River System: We use a simple mass-balance equation to account for how all water entering the system will ultimately leave. We call this model "open-loop" since it does not yet incorporate the control policies implemented by the water commissioner. Previous research used a mass-balance model to represent a different portion of the Sevier River, and basic parameters in the model were identified from historical flow data [16]. Here, we adopt a similar model, given by:

$$y[k] = a_1 w_1[k-l_1] - w_2[k-l_2] - w_3[k-l_3] - w_4[k-l_4], \quad (1)$$

where

$$\begin{aligned} w_2[k] &= a_2 u_2[k], \qquad w_3[k] = a_3 u_3[k], \\ w_4[k] &= a_4 u_4[k] + a_5 u_5[k]. \end{aligned} \quad (2)$$

and y represents the measured downstream flow (positive flow being out of the system) near Lynndyl, and k is time measured in hours.

Controlled input w_1 is chosen by the water commissioner and represents the flow into the system near Juab. Other controlled inputs are u_2, u_3, u_4, and u_5 defining the gate heights (measured in feet) on the Central Utah, Vincent, and Leamington canals, respectively (the Leamington canal uses two gates in parallel). Measured states w_2, w_3, and w_4 are the corresponding flows out of the system via these canals, where all flows in the system are measured in cubic feet per second (cfs).

In (1) the constant a_1 represents the effect of possible unregulated and unmeasured inflows and outflows to the river. These could include small streams that enter or leave the main flow of the river, evaporation, or rainfall. Additionally, each canal flow w_i, $i = 2, 3, 4$ can also be modeled as a proportionality constant a_i multiplied by the gate height, as shown in (2).

The delays in the system–denoted by l–represent the time it takes (in hours) for the water to flow from a given point along the river to the point where y is measured. For example, l_4 is the time it takes for water to flow from the head of the Leamington Canal to the Sevier River at Lynndyl.

All values of $y[k]$, $u_i[k]$, and $w_j[k]$ are saturated to be at least zero at all times k. Likewise, the gate heights $u_i[k]$ are saturated maximally with the maximally observed values in the data. Due to this saturation, the model is not linear. Lags l_i and parameters a_j are all determined through a system identification technique, which will be covered in Step 2.

Model Class of the Water Commissioner: We now develop a model class of the water commissioner's control policies managing the flow of water through this segment of the river. By observing the data for the water commissioner's control w_1, we can see the data points where the water commissioner changes the gate heights in the reservoir. We can also offset y by the delay l_1 to make an approximation as to what the desired output would have been at that time.

This results in a piecewise constant reference signal $r_1[k]$ that can account for the factors for which there is no data. A standard PI controller is then used to approximate the behavior of the water commissioner, given by:

$$w_1^{pred}[k] = k_p(r_1[k] - y[k]) + k_i \sum_{i=0}^{k}(r_1[i] - y[i]), \quad (3)$$

where k_p and k_i are parameters to be tuned in order to minimize the error between w_1^{pred} and the actual observed w_1.

State Space Representation of the Closed-Loop System: We can express the closed loop system consisting of (2) and (3) in the form

$$x[k+1] = Ax[k] + B \begin{bmatrix} r[k] \\ u[k] \end{bmatrix}, z[k] = Cx[k] \quad (4)$$

where

$$\begin{aligned} z[k] &= \begin{bmatrix} y[k] & w_1[k] & w_2[k] & w_3[k] & w_4[k] \end{bmatrix}^T \\ r[k] &= \begin{bmatrix} r_1[k] & r_2[k] & r_3[k] & r_4[k] \end{bmatrix}^T \\ u[k] &= \begin{bmatrix} u_2[k] & u_3[k] & u_4[k] & u_5[k] \end{bmatrix}^T \end{aligned} \quad (5)$$

Step 2: System Identification Using Open-Source Data

We now use a system identification method to learn the model parameters in (4) for this segment of the river. Much of the data for the river is available at the SRWUA website (`http://sevierriver.org`). This site hosts sensor logs measuring gate heights and water flows at an hourly rate. Some data was also taken from the U.S. Geological Survey (USGS) website (`http://www.usgs.gov`). Roughly 4.5 years of hourly data (Jan 2009 - Jun 2014) were extracted from these sources, resulting in more than 47,000 measurements. The model was trained on two thirds of the data and was tested on the remaining third.

System Identification of the River System: We first learn the parameters l_i, with the simplifying assumption that they are time invariant and with a focus on accuracy during high flow periods. Since the volume of water flow down the main section of the river is roughly an order of magnitude higher than that of the branches, $y[k]$ is most affected by $u[k-l_1]$. Thus, l_1 can be inferred by measuring the time interval measuring the difference in the appearance of features (such as spikes) in w_1 and y. We found that $l_1 = 29$ hours was the best fit to align the features of w_1 and y. Using this information, we use linear interpolation with respect to distance to compute the remaining lags, yielding $l_2 = l_3 = 12$ and $l_4 = 11$.

Next, we learn a_i. Assume, for a moment, that $a_1 = 1$. Since y and all w_i are known from public data, we can define a known signal w_1' as follows:

$$w_1'[k-l_1] = y[k] + w_2[k-l_2] + w_3[k-l_3] + w_4[k-l_4]. \quad (6)$$

Signal w_1' is the value of w_1 if the system had no unmeasured inflows and outflows. We then use a_1 to adjust w_1 in order to account for these unmeasured inflows and outflows. Ideally, we would have that $w_1'[k] = a_1 w_1[k]$ for all k; however, this is unlikely to occur. Therefore, the best we can do is measure an error $e_1 = w_1' - a_1 w_1$ and choose a_1^* such that

$$a_1^* = \arg\min_{a_1} \|e_1\| = \arg\min_{a_1} \|w_1' - a_1 w_1\|, \quad (7)$$

over some choice of norm $\|\cdot\|$. For our purposes, we choose the 1-norm, and (7) can then be solved using $L1$ regression.

We repeat this procedure for finding the remaining a_i. As a result, we find that $a_1 = 1.0709$, $a_2 = 5.1212$, $a_3 = 18.63$, $a_4 = 1.059$ and $a_5 = 1.1467$. The range in values for these parameters can be attributed to the varying gate and canal sizes.

Using this model with our validation data, a predicted outflow to the system is computed, as shown in Fig. 2. The largest discrepancies that occur between predicted and actual output are due to time periods where there was initially missing data. The places where there is consistent data (year 1 and the beginning of the irrigation season in year 2) are also the most consistent with the model.

System ID of the Water Commissioner: We now develop a model of the water commissioner's control policies managing the flow of water through this segment of the river.

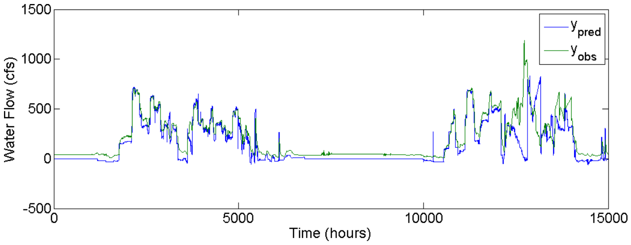

Fig. 2 The predicted outflow values for the open-loop learned model of the river system vs. the observed values.

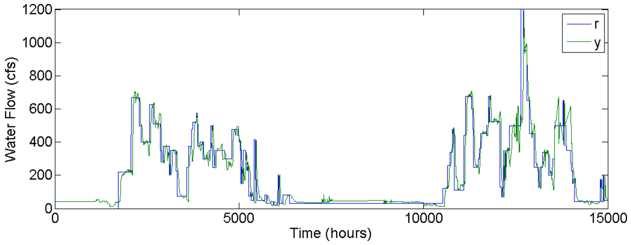

Fig. 3 The reference signal r_1 modeling the requests for the river outflow vs. the actual outflow y.

Observing the w_1 data, we can see the data points where the water commissioner changes the gate heights in the reservoir. Combining this with the computed delay, we can look at what happened at y at these data points to make an approximation as to what the desired output would have been at that time. This results in a piecewise constant reference signal r_1 that can account for the factors for which we have no data, such as downstream demand. In Fig. 3, observe that the magnitude of r_1 is close to y, but that there is a delay between them, as one would expect.

Equipped with signal $r[k]$, we can tune (3) to develop a model of the water commissioner's control policy, finding that $k_i = 0.014$ and $k_p = 0.01$ are good fits. As a test, we compared the predicted control inputs for w_1 with the actual control inputs submitted by the water commissioner, as shown in Fig. 4.

The Closed-Loop System: With these learned parameters, a state space model of the closed loop system in the form of (4) can now be built (noting again that this system is not linear since the states are all saturated to be at least zero). In order to represent the learned delays, this state space equation utilized 66 states in total, though the resulting A, B, and C are sparse. In Fig. 5, we show the predicted y values for this closed-loop model compared to that of the observed values of y.

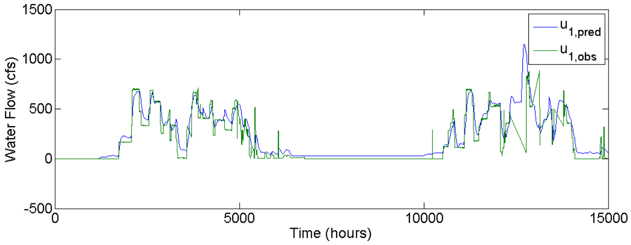

Fig. 4 The inferred inflow (output from water commissioner) w_1^{pred} compared with the observed values w_1.

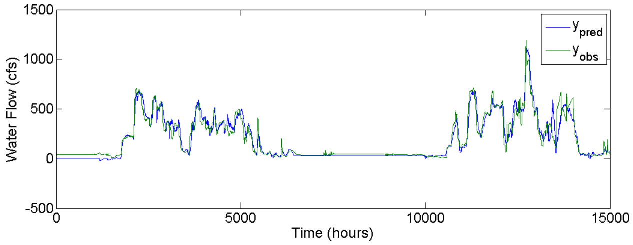

Fig. 5 The predicted outflow values y for the closed loop model of the river and the water commissioner vs. the observed values.

Step 3: Identify the Exposed Variables

We now define the set of variables which can be observed and manipulated by an attacker. In this situation, we define the set of exposed variables as precisely those that are available publicly through the internet. In particular, y and all w_i will be considered exposed variables.

Note that the model identified in the previous sections includes many more states than those included in the set of observed variables. The reason is that most of the states in the state space model from Step 2 represent previous data values. Since the attacker has access to the same data, we can just say that these 5 variables are the exposed variables. If the river system had more sensors that were not available to the attacker, these would be identified and excluded in this step.

Step 4: Model the Attack Surface

We define the attack surface as a signal structure representing the set of exposed variables defined in the previous section combined with the causal dynamic interactions between them. These interactions can be represented through the use of the DSF, which is defined as the matrix pair $(Q(z), P(z))$ such that

$$Y(z) = Q(z)Y(z) + P(z)U(z) \tag{8}$$

where the vector $Y(z)$ is the $\mathcal{Z}$-transform of the state vector exposed to the attacker and $U(z)$ is the $\mathcal{Z}$-transform of the input vector. The matrix $Q(z)$ is a matrix where each entry is a SISO transfer function, and models how each manifest state affects the other manifest states in the system, possibly through hidden states. Similarly, the matrix $P(z)$ has SISO transfer function entries, but models how the input affects the state of the system, possibly through hidden states. For more information on the formulation of DSF, see [17].

The DSF can be viewed as a left factorization of the behavioral transfer function representation $G(z)$ of the system, where $G(z) = (I - Q(z))^{-1} P(z)$. While the signal structure representation may not capture the structure at the same resolution as a state space representation, it nonetheless provides more structural information than a behavioral model. This corresponds to the attacker perspective of the

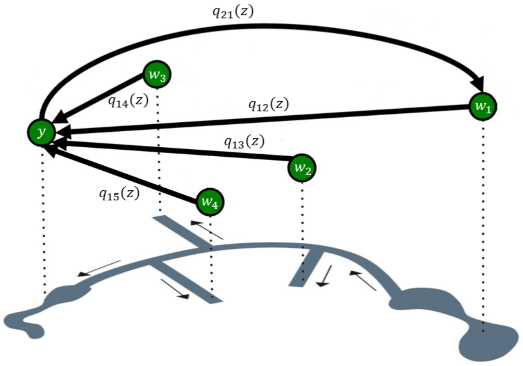

Fig. 6 A graphical representation of the attack surface of the river system as given by the DSF.

system, which is in general more sophisticated than merely understanding the input-output response of the system, but may not be attuned to every nuance of the model created in Step 2.

To transform the state equations learned in Step 2, we first relax the saturation constraints on the state variables, allowing the model to become linear. Since the purpose of this analysis is to highlight locations in the system which are sensitive to attack, rather than to find the most efficient attack available, such a relaxation will not have a profound impact on the results. Now let

$$\begin{aligned} Y(z) &= \mathcal{Z}\{\begin{bmatrix} y[k] & w_1[k] & w_2[k] & w_3[k] & w_4[k] \end{bmatrix}^T\} \\ U(z) &= \mathcal{Z}\{\begin{bmatrix} r[k]^T & u[k]^T \end{bmatrix}^T\} \end{aligned}$$

Following the procedure outlined in [17], we arrive at the following DSF representation of the water system:

$$Q(z) = \begin{bmatrix} 0 & \frac{10709}{10000z^{29}} & -\frac{1}{z^{11}} & -\frac{1}{z^{10}} & -\frac{1}{z^{8}} \\ -\frac{12z-5}{500z^2-500z} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Note that, as we will explain in the subsequent section, $P(z)$ is not used in the vulnerability analysis; therefore we do not include it here.

Fig. 6 gives a graphical representation of $Q(z)$ showing the causal dynamic relationships between the manifest states in the river system.

Step 5: Analyze System Vulnerability

We now use the results outlined in [12] and [4] to highlight which link in the signal structure representation of the system is most vulnerable. Note that, for the purposes of this paper, only single-link attacks will be considered.

We define vulnerability here in the sense of vulnerability to destabilizing attacks. Let $q_{ij}(z)$ be a link in the system. It has been shown that $q_{ij}(z)$ is vulnerable if and only if the closed-loop transfer function seen by that link–given by $h_{ji}(z)$ where $H(z) = (I - Q(z))^{-1}$–is non-zero. In other words, link $q_{ij}(z)$ is vulnerable if and only if it is in a cycle. Note that, by definition, no links in $P(z)$ are in a cycle; therefore, no links in $P(z)$ are vulnerable. For this reason, we limit our vulnerability analysis to links in $Q(z)$.

The vulnerability v_{ij} of link $q_{ij}(z)$ can also be expressed in terms of $h_{ji}(z)$. Using the small gain theorem, we can find that the minimum size of perturbation $\|\Delta\|$ on $q_{ij}(z)$ required to destabilize the entire system is precisely $\|h_{ji}(z)\|$. Therefore,

$$v_{ij} = \frac{1}{\|\Delta\|} = \frac{1}{\|h_{ji}(z)\|}. \tag{9}$$

Note that the ∞-norm is typically used in (9). Thus the vulnerability of a link becomes the inverse of the magnitude of the smallest perturbation on that link required to destabilize the entire system.

Notice from Fig. 6 that only two links in $Q(z)$ are in a cycle, particularly link $q_{12}(z)$ corresponding to the dynamics from w_1 to y as well as link $q_{21}(z)$ corresponding to the dynamics from y to w_1.

Using (9), we compute the vulnerability as $v_{12} = 2.512$ and $v_{21} = 298.6$. Therefore, link $q_{21}(z)$–corresponding to the feedback link from y to w_1–is the most vulnerable link in this system, meaning that perturbations on this link will have the most impact on the system. This is where we design our attack.

Step 6: Design an Attack

The feedback link identified by the vulnerability analysis is the link representing the water commissioner. Since we are dealing with a human in the loop, there are many ways to perturb a human: threats, bribery, or social engineering, for example. We consider here a simple perturbation where information across this link is delayed, which could be effective against both human and automated controllers.

To implement a delay on this system, an attacker could infiltrate the VHF radio network that moves the sensor data from the river to the internet, since currently no encryption or security is used. Likewise, the website reporting the data could be hacked to show erroneous values using a man-in-the-middle attack, since no encryption is used on the website.

In order to illustrate the possible effects of such an attack, consider a thought experiment. The value at y is high, so the controller releases less water. But, since the signal at the bottom is delayed, the controller sees that although it released less water, the level is still high, so it releases even less. Depending on the severity, the controller could shut the water flow down to extremely low levels, thinking that weather or some other phenomenon is making up for the deficit. By the time the controller gets the signal that in fact the water is low, it must make up for that by releasing more water. However, since this is just the delayed response, the controller believes that this has somehow led to less water downstream, so it releases more. Following this pattern, one can see that the system could become unstable (see Fig. 7).

V. Simulated Results and Estimated Impact

We implemented a delay attack on a simulation of the water system as described in Step 6. Fig. 7 shows the water

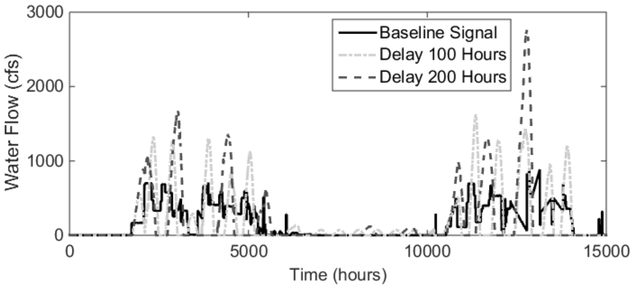

Fig. 7 Perturbation attack using delays in information. Notice the dramatic spikes in the delayed systems

flow that would result in delays of 100 hours and 200 hours. As can be seen, the attack causes the flow of water to oscillate wildly, indicating that the attacks are causing system instability even though the saturation on the states prevents the oscillations from growing unbounded.

In order to further show the impacts of these attacks, we present a rough financial estimate of potential losses. As mentioned, the Sevier River and its tributaries are responsible for irrigating about $186,600$ acres of farmland. According to [13], most of this irrigated cropland is alfalfa, which is harvested in multiple cuts per season. In Utah 4 cuts for a season is common. Also, in an average season, one can expect to harvest 10 tons per acre, or an average of 2.5 tons per acre per cut [18]. At the time of publishing, the price for alfalfa is $150 per ton for average quality according to the United States Department of Agriculture.

Suppose that the attack destroys one cut of alfalfa (which could happen one of the downward oscillations corresponds to a phase in the growth of alfalfa in which water is critical). This would lead to a loss of $375/acre. Using alfalfa as an indicator crop (since it's grown on a majority of the irrigated land) and extrapolating the attack to the entire irrigated acreage in the Basin shows that potential monetary losses would amount to approximately $70 million for the season.

Note that anomaly detection could potentially detect such a delay attack early, preventing the attack from destabilizing the system. However, attacks that are unsuccessful in destabilizing the system can still cause large amounts of damage by releasing unneeded water or withholding it in seasons of need. This could be particularly impactful in the drought-prone desert region of Central Utah.

VI. Conclusion and Discussion

In conclusion, we have presented a systematic methodology for designing an attack on an existing critical infrastructure system which includes cyber, physical, and human components. By using publicly available data, we can construct a view of an attack surface–of the components of the system seen by the attacker combined with links defining the causal dynamic relations between these components. These links can then be analyzed, one-by-one, to determine which is most sensitive to perturbations caused by the attacker. In this particular case, the most sensitive link happened to be the human in the loop. Equipped with this knowledge, an attacker now has many options available which can be used to perturb the system in order to cause considerable economic damage to the agricultural economy of Central Utah.

VII. Acknowledgments

The authors would like to thank Roger Hansen, David Ainscough, Bradley Heistermann, and Arlen Hilton of the Provo office of the US Bureau of Reclamation along with water commissioners Jim Walker and Ray Owens. These people provided our research group with valuable information and insight.

References

[1] J. Slay and M. Miller, "Lessons learned from the Maroochy water breach," in *Critical Infrastructure Protection*, ser. International Federation for Information Processing (IFIP). Springer US, 2008, vol. 253, pp. 73–82. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-75462-8_6

[2] J. Leyden, "Polish teen derails tram after hacking train network," *The Register*, vol. 11, 2008.

[3] R. Langner, "Stuxnet: dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[4] A. Rai, D. Ward, S. Roy, and S. Warnick, "Vulnerable links and secure architectures in the stabilization of networks of controlled dynamical systems," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 1248–1253.

[5] A. Cox, S. Roy, and S. Warnick, "A science of system security," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 487–492.

[6] Q. Zhu and T. Basar, "Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: games-in-games principle for optimal cross-layer resilient control systems," *Control Systems*, vol. 35, no. 1, pp. 46–65, 2015.

[7] A. Teixeira, K. C. Sou, H. Sandberg, and K. H. Johansson, "Secure control systems: A quantitative risk management approach," *Control Systems, IEEE*, vol. 35, no. 1, pp. 24–45, 2015.

[8] F. Pasqualetti, F. Dorfler, and F. Bullo, "Control-theoretic methods for cyberphysical security: Geometric principles for optimal cross-layer resilient control systems," *Control Systems, IEEE*, vol. 35, no. 1, pp. 110–127, 2015.

[9] J. Gonalves and S. Warnick, "Necessary and sufficient conditions for dynamical structure reconstruction of lti networks," *IEEE Transactions on Automatic Control*, Aug. 2008.

[10] S. Warnick, "Shared hidden state and network representations of interconnected dynamical systems," in *53rd Annual Allerton Conference on Communications, Control, and Computing*, Monticello, IL, 2015.

[11] V. Chetty and S. Warnick, "Network semantics of dynamical systems," in *Conference on Decision and Control*, Osaka, Japan, 2015.

[12] V. Chetty, N. Woodbury, E. Vaziripour, and S. Warnick, "Vulnerability analysis for distributed and coordinated destabilization attacks," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 511–516.

[13] *A Water-Related Land Use Inventory Report of the Sevier River Basin*, Utah Department of Natural Resources, Division of Water Resources, June 2011.

[14] P. H. Gleick, "Water and terrorism," *Water policy*, vol. 8, no. 6, pp. 481–503, 2006.

[15] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on SCADA systems," in *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*. IEEE, 2011, pp. 380–388.

[16] M. Maxwell and S. Warnick, "Modeling and identification of the Sevier River system," in *American Control Conference, 2006*. IEEE, 2006, pp. 6–pp.

[17] J. Goncalves, R. Howes, and S. Warnick, "Dynamical structure functions for the reverse engineering of LTI networks," in *Conference on Decision and Control*, New Orleans, LA, 2007.

[18] T. G. C.S. Poulson, M.G. Pace and K. Pack, "Irrigated alfalfa variety performance, 2003-2005; Delta, UT," Utah State University Cooperative Extension, Tech. Rep., April 2006.