Compromising Security of Economic Dispatch in Power System Operations

Devendra Shelar†, Pengfei Sun*, Saurabh Amin† and Saman Zonouz*
Civil and Environmental Engineering, *Electrical and Computer Engineering
†Massachusetts Institute of Technology, *Rutgers University
{*shelard, amins*}@*mit.edu*, {*pengfei.sun, saman.zonouz*}@*rutgers.edu*

***Abstract*—Power grid operations rely on the trustworthy operation of critical control center functionalities, including the so-called Economic Dispatch (ED) problem. The ED problem is a large-scale optimization problem that is periodically solved by the system operator to ensure the balance of supply and load while maintaining reliability constraints. In this paper, we propose a semantics-based attack generation and implementation approach to study the security of the ED problem.[1] Firstly, we generate optimal attack vectors to transmission line ratings to induce maximum congestion in the critical lines, resulting in the violation of capacity limits. We formulate a bilevel optimization problem in which the attacker chooses manipulations of line capacity ratings to maximinimize the percentage line capacity violations under linear power flows. We reformulate the bilevel problem as a mixed integer linear program that can be solved efficiently. Secondly, we describe how the optimal attack vectors can be implemented in commercial energy management systems (EMSs). The attack explores the dynamic memory space of the EMS, and replaces the true line capacity ratings stored in data regions with the optimal attack vectors. In contrast to the well-known false data injection attacks to control systems that require compromising distributed sensors, our approach directly implements attacks to the control center server. Our experimental results on benchmark power systems and five widely utilized EMSs show the practical feasibility of our attack generation and implementation approach.**

I. Introduction

Critical national infrastructure has become increasingly complex. The power grid exemplifies a cyber-physical infrastructure, with data collected from its physical components and processed by control algorithms running on computers to provide for accurate and safe monitoring and control. Such a large-scale trusted computing base introduces a hard-to-protect attack surface. Events such as proliferation of the Stuxnet worm [10], the coordinated attack on the Ukranian power grid [5], and the emergence of new threats that leverage existing weaknesses in these systems [23] demonstrate that cyber-physical infrastructures are unprepared to maintain their safe and secure operation in the face of malicious adversaries.

Despite the failures, the past intrusions had two features: *i)* they mostly required full ownership of the target controllers (e.g., Siemens Step7 server compromise by Stuxnet [10]) to perform the attacks; and *ii)* they did not fully optimize their adversarial impact via utilization of the underlying physical model. A semantics-based attack can do a lot more using much less resources. For instance, an attacker with access to only few power system parameters can leverage its dynamical model to calculate the malicious replacing parameter values such that the ultimate damage to the power system is maximized.

In the literature, there has been an extensive body of work on false data injection attacks [17], where the compromised sensors send corrupted measurements to mislead the operators regarding the power system state. Such attacks assume the attacker can compromise a large number of geographically and logically distributed set of sensors remotely. In addition to the scalability barrier, remote malicious access to (analog) sensors with serial connections may not be feasible in practice. Additionally, by design, false data injection attacks target sensors or actuators only, and cannot manipulate core system parameters such as the network topology and line parameters (e.g., capacities). This information often resides within the control center servers and are used for power system operations such as state estimation and operational control. However, almost all the past real attacks (e.g., [5], [10]) against critical infrastructures have targeted control center assets (as opposed to individual sensors or actuators).

A. Our focus

This article presents a semantics-aware attack against a widely used power grid network control functionality, and demonstrates its practical feasibility on well-known *Energy Management System (EMS)* softwares. Specifically, we conduct a vulnerability assessment of an important functionality provided by all EMSs – the so-called *Economic Dispatch (ED)* problem. In critical infrastructures, ED is routinely solved to set the generator output levels over a control area of a regional transmission grid. We show that software security vulnerabilities in power system controllers can be exploited by an attacker (an external hacker or a strategic market participant) to gain a backdoor entry into power grid operations.[2] By utilizing the knowledge of an approximate power flow model – specifically, DC approximation – the attacker can launch a semantic memory attack to change the critical parameters such as transmission line ratings (capacities). A transmission line's rating reflects the maximum amount of power that it can carry without violating safety codes or damaging the line. We design experiments using ED implementation on real-world EMS software packages to demonstrate the economic and safety risks posed by use of manipulated line ratings.

[1] This is a Regular research paper.

[2] Throughout the paper, we use the term *controller* as the ED implementation software packages that solve economic dispatch problem.

The core of our attack generation approach against the power grid infrastructure is a bilevel optimization problem that encodes the attacker's partial knowledge of power system operations to compute the target malicious power system parameters. This physics-aware attack generation approach enables us to identify key features of power system data and software operations whose exposure can significantly increase security risks. The implementation of our optimal attack against power system operation involves targeted manipulation of specific power system parameters that reside within the EMS's dynamic memory space. The exploit performs an online memory data search using lightweight pattern matching to locate the sensitive power system parameters used by the ED software to calculate the generation output levels. The use of manipulated parameter values makes the EMS issue incorrect dispatch (generation and power flow) commands, and consequently drive the power system towards unsafe states. The merit of our overall approach lies in the combination of the semantics-based optimal attack generation and a generic implementation procedure for EMS's memory data corruption.

The bilevel problem for attack generation can be viewed as a sequential game between the attacker (leader) and the follower (grid operator). In the first stage, the attacker chooses power system parameter manipulations with the objective of maximizing the violation of capacity limits; in the second stage, the operator solves the ED to determine generator output levels while facing the manipulated parameters chosen by the attacker in the first stage. We show that the optimal power injections and nodal voltages computed using the manipulated parameters yield suboptimal and unsafe power flow allocations. This significantly increases the possibility of cascading failures and the risk of subsequent emergency actions.

Thus, the main contributions of our paper are as follows:

- We introduce a new domain-specific semantic data attack against power grid controllers. The attack leverages an approximate model of power system to manipulate the controller runtime memory such that the execution of the legitimate controller software, using partially corrupted values, drives the physical plant towards unsafe states.
- We formulate the problem using a game-theoretic framework to optimize the attack strategy in terms of which available data regions in the controller memory space should be modified. The adversary-optimal values are calculated using fast bilevel optimization procedures.
- We implemented working prototypes of the proposed controller attack against real-world large-scale and widely-used energy management systems. Our implementations leverage logical memory invariants to locate the sensitive power system parameters in the controller's memory space. The evaluation results prove the feasibility of domain-specific data corruption attacks to optimize for the physical damage.

In the remaining of this section, we present an overview of our proposed attack. Section II and Section III present the attack model and optimization algorithm to calculate the parameter manipulations that will maximize the ultimate adversarial impact of resulting power flows. Section IV and

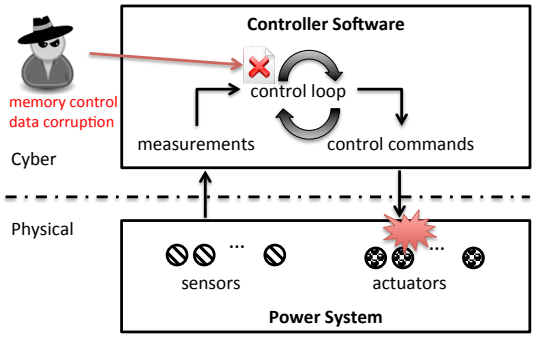


Fig. 1: Physics-aware memory attack on control systems.

Section VI present our empirical experiments with real-world commercial power grid monitoring and control software solutions. Section VII discusses the potential mitigation strategies, and Section VIII reviews the related work.

B. Solution Overview

Our contribution builds on two perspectives that have evolved in the emerging field of cybersecurity of networked control systems. The first perspective involves the analysis of state estimation and control algorithms under a class of attacks to sensor measurements or actuator outputs [24]. These attack models reflect the loss of availability (resp. integrity) of measurements/outputs when the communication network linking the physical system and remote devices is compromised. Recent work has studied how the physical system's performance and stability can be compromised by such attacks [17]. Typically the attacker is assumed to be a resource-constrained adversary with only partial (or possibly full) knowledge of system, and a resilient control design problem is to ensure a reliable and safe performance against arbitrary actions that can be performed by the attacker. These results are grounded in the theory of robust and intrusion tolerant control, which provides a quantitative framework to study the tradeoffs between efficiency in nominal conditions and robustness during non-nominal ones including the attacker-induced failures. In contrast, as illustrated in Figure 1, our attack model considers direct data corruption (specifically, manipulation of power system critical parameters) in the live memory of EMS software, where all distributed sensor measurements are received and processed, i.e., single point of compromise. Hence, individual infections of distributed sensors are not required unlike previous work on false data injection attacks [18]. This allows us to study how the vulnerabilities in control software implementations and in their links to external data sources can be exploited by the attackers.

A second perspective has emerged in the vulnerability assessment of large-scale power grids against physical attacks [6]. Here the objective is to find worst-case disturbance or an *adversary-optimal attack* to physical components that can maximize the impact on grid functionality, even under perfect observability and best response by the operator (defender). Various classes of failures have been considered, for e.g., line failures, sudden loss of generation, and load disconnects. Typically, these problems are formulated as bilevel optimization problems, and involve explicit consideration of both physical

constraints (e.g., power flows, generation constraints, and line capability limits) as well as resource constraints of the attacker. Examples of physical security problems that have been considered using this framework include $N-k$ contingency analysis problem [7], network interdiction under line failures, and modeling of cascading failures that originate due to local component failures in one sub-network and progressively propagate to other sub-networks of the grid. However, existing work on adversary-optimal attack does not consider how such an attack can be executed in controller software. In our work, we combine the computation of adversary-optimal attack with analysis of EMS software to execute the attack.

Threat model. Our adversary model is concerned with stealthy memory data corruption of EMS (that typically sits within the control center); thus, we require a compromised controller process within the EMS server. This is a realistic assumption, because it requires lesser privileges compared to the past real incidents such as Stuxnet [10] and BlackEnergy [5] that took complete control of the servers. With the access to EMS dynamic memory, the exploit targets the true memory-resident power system critical parameters, and implements calculated adversary-optimal incorrect values in EMS memory.

We emphasize two aspects of our model: Firstly, our attack generation and implementation approach is *generalizable*. However, to concretely illustrate our approach and to evaluate its feasibility, we assume that the attacker is concerned with generating "optimal" dynamic line ratings (DLRs) to maximize capacity violations. Indeed, other variations of attack generation are possible, for e.g. manipulation of other parameters such as generator/loads/voltage bounds, etc. Secondly, our implementation approach is motivated by server-side attacks to EMS software and emphasizes the *stealthiness* of the attack. Specifically, the in-memory parameter manipulations are still within acceptable limits and hence pass the typical out-of-bound checks for false data injections. Thus, they can remain dormant in controller's memory and can produce the intended consequences (e.g. thermal overloading, or even physical damage) before the last line of defense (i.e., physical fail-safe mechanisms) are triggered. Again, other ways of implementing our attack are possible, for e.g. intercepting network communication and injecting false data.

Implementations. We perform off-line binary analysis to locate the power system parameters in the controller's memory space. We use this information to extract logic-based structural pattern signatures (invariants) about the memory around power system parameter value addresses. The signature predicates are checked during attack-time to identify the real parameters on the victim controller memory space. Such pattern-based search (as opposed to absolute memory address-based search) is required because analysis-time (offline) and attack-time (online) parameter value addresses in memory often differ. This is because of unpredictable execution paths (due to potentially different workloads) across different runs that result in different heap memory allocation function call/return sequences, and hence different allocated memory addresses. Finally, the attack achieves a certain level of stealthiness by ensuring that the incorrect parameters reflect similar general trends as the true ones.

II. Optimal Attacks to Economic Dispatch

In this section, we describe how the attacker generates a semantic attack that utilizes the knowledge of an approximate model of power flow to manipulate the model parameters used by the ED software. We choose DC model as the approximate model known by the attacker, and line capacities as the targeted model parameters.

We show that under our adversary model, the allocation generated by the ED implementation under the manipulated capacity ratings, causes the power flows on the transmission lines to exceed the actual line capacity ratings. Specifically, its implementation on the power system will lead to the violation of safe thermal limits of the lines. This can cause the lines to rapidly deteriorate or degrade, increasing their likelihood of tripping. The sudden disconnection of power lines can cause an outage. It may cause a short circuit between two lines that can ignite a fire. Coming in contact with a line that is live, can also kill people, seriously injure them. Thus, such a semantic attack increases both reliability and safety risks in power system operations to a significant degree.

In our attack model, the attacker chooses the DLR manipulations in a way such that his actions are not obvious to the System Operator (SO). If the effect of the attack is not visible to the SO (for e.g., via line flow measurements or emergency signals), the SO will not invoke generation curtailment and/or line disconnect operations. In fact, under partial network observability, the operator may not be able to implement the necessary preventive actions in a timely manner. As a result, the SO will implement the false ED solution that will violate the line limits.

A. Attacker Knowledge

We first describe the attacker's system knowledge which consists of DC-approximation of the actual nonlinear AC power flow equations. The topology of a transmission network can be described as a connected graph with the set of nodes $\mathcal{V}$ and the set of edges $\mathcal{E}$. In power systems terminology, each node refers to a bus and each edge refers to a transmission line. We let $n = |\mathcal{V}|$. Let $\{i, j\}$ denote the line joining the nodes i and j, and its susceptance (inverse of reactance) be denoted as β_{ij}. The set of generators at a bus i is denoted as $\mathcal{G}_i$. The set of all generators is denoted by $\mathcal{G} := \mathcal{G}_i$. For each $i \in \mathcal{G}$, p_i^{min} and p_i^{max} are the lower and upper generation bounds that are specific to the $i-$th generator. The generation bounds can be expressed as constraints on individual p_i:

$$p_i^{min} \leqslant p_i \leqslant p_i^{max}. \tag{1}$$

Following the standard formulation of economic dispatch, the cost of power generation for the $i-$th generator is modeled as a convex quadratic function $C_i(p_i)$ in p_i. Let $p \in \mathbb{R}^{\mathcal{G}}$ and $d \in \mathbb{R}^{\mathcal{V}}$ denote the generation and demand vectors, respectively. The total cost of generating p is:

$$C(p) = \sum_{i \in \mathcal{G}} C_i(p_i), \tag{2}$$

where

$$C_i(p_i) = a_i {p_i}^2 + b_i p_i + c_i. \tag{3}$$

$a_i, b_i, c_i \in \mathbb{R}_+ \ \forall \ i \in \mathcal{G}$. a_i and b_i are not simultaneously zero, i.e., the cost of generation is an increasing function of power (MWs) supplied.

The power flow f_{ij} from node i to node j can be expressed as a linear function of the difference between the voltage phase angles at nodes i and j [6]:

$$f_{ij} = \beta_{ij}(\theta_i - \theta_j), \tag{4}$$

where $\theta \in \mathbb{R}^{\mathcal{V}}$ is the vector of voltage phase angles.

The conservation law for the power flows is:

$$\sum_{j:\{i,j\}\in\mathcal{E}} f_{ij} = \sum_{k\in\mathcal{G}_i} p_k - d_i, \tag{5}$$

which states that the net generation at a node i is equal to the sum of outflows from node i to its neighbors. The DC power flow (4)-(5) is said to be feasible if and only if total supply is equal to total demand (see [6]), i.e.,

$$\sum_{i\in\mathcal{G}} p_i - \sum_{j\in\mathcal{V}} d_j = 0. \tag{6}$$

The power flows satisfy the capacity line constraints, i.e.,

$$|f_{ij}| \leqslant u_{ij}. \tag{7}$$

Thus the DC-optimal power flow problem faced by the SO can be posed as follows:

$$\min_{p,\theta} \quad C(p) \qquad \text{s.t. } (1)-(6),(7). \tag{8}$$

B. Attacker Resources

The true capacities of the transmission lines dynamically vary over time due to weather conditions (ambient temperature, wind, etc.) [9], and are, in fact, greater than the static line ratings assumed by the SO for economic dispatch problem (Figure 2). Dynamic Line Rating (DLR) lines are the transmission lines with DLR sensors that report the true line capacities to the system operator.

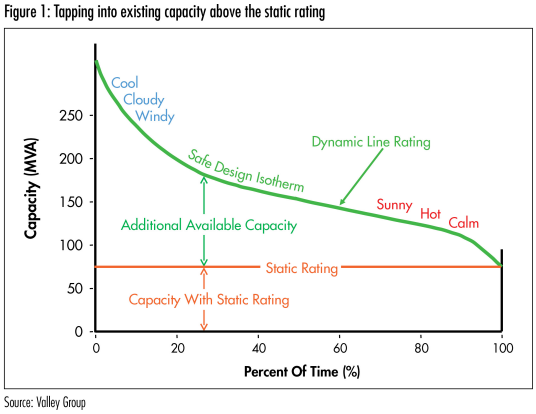


Fig. 2: Static vs Dynamic Line Rating

Let $\mathcal{E}_D \subset \mathcal{E}$ denote the set of lines that are equipped with DLR devices. The complementary set $\mathcal{E}_S = \mathcal{E}\backslash\mathcal{E}_D$ denotes the set of lines that are not equipped with DLR technology, and hence their rating will be fixed to the respective static line capacity values. Given that DLR deployments are done as part of government sponsored smart grid projects [8], [9], the set of lines $\mathcal{E}_D$ equipped with DLR technology is public knowledge. These lines will be the ones that are routinely prone to congestion and hence receive priority DLR implementation by the operator.

For a line $\{i,j\} \in \mathcal{E}_D$, we denote u^d_{ij} as the actual line rating computed by the DLR software using measurements collected from the Supervisory Control and Data Acquisition (SCADA) system.

$$u_{ij} = \begin{cases} u^s_{ij} & \text{if } \{i,j\} \in \mathcal{E}_S \\ u^d_{ij} & \text{if } \{i,j\} \in \mathcal{E}_D, \end{cases} \tag{9}$$

where

$$\forall \ \{i,j\} \in \mathcal{E}_D \quad u^{min}_{ij} \leqslant u^d_{ij} \leqslant u^{max}_{ij} \tag{10}$$

i.e. the DLRs can only take values between a certain range.

Thus the DC-optimal power flow problem faced by the SO can be posed as follows:

$$\min_{p,\theta} \quad C(p) \qquad \text{s.t. } (1)-(6),(7),(9). \tag{11}$$

We assume an *informed attacker*. Specifically, the attacker's knowledge includes the network topology, line susceptances, set of generators, and their corresponding generation limits, and the cost of generation. The attacker also knows the nominal demand d_j at each node j and the nominal generator output p_i for each $i \in \mathcal{G}$. In power systems terminology, with this knowledge, the attacker can solve for an DC ED solution which is an approximation of AC ED solution that the EMS implements on the power system. Note that our assumption on attacker's knowledge is not unrealistic given that all major ISOs publicly disclose historical generation and demand patterns and the locational marginal prices in day ahead and hourly power markets.

Since the SO knows the static line ratings and these are fixed in ED software implementations, we assume that the attacker cannot compromise them in ED implementation's memory. Any compromise to static line ratings can be overridden by simple built-in checks in power flow implementations. Also, since the static ratings are typically calculated for constant (worst-case) weather conditions over an extended period of time (few months to years), we assume that the attacker knows their values. This assumption can be justified by the fact that the manufacturers of transmission line conductors supply static line ratings in their product specifications. Thus, under the aforementioned constraints, the set of lines $\mathcal{E}_D$ constitutes the attacker's constraint since the attacker only targets DLR ratings and not the static ones.

C. Attack Objective

Now, we present the constraints faced by the attacker so that the attack remains stealthy, and the SO's ED software admits the DLR ratings manipulated by the attacker. Then, we formulate the attack policy of the attacker as a bilevel optimization problem.

Under our attack model, the attacker accesses the actual DLR values u^d_{ij} for lines $\{i,j\} \in \mathcal{E}_D$ in ED's dynamic memory and replaces them with incorrect values u^a_{ij} (Section VI). The attacker knows u^d_{ij} and computes u^a_{ij} in order to maximize

the violation of line ratings by the resulting power flows. To avoid detection by in-built checks, each u^a_{ij} is constrained by minimum and maximum permissible limits of line ratings, denoted as u^{min}_{ij} and u^{max}_{ij}, respectively. These limits are also known by the attacker. For ease of presentation, we introduce an auxiliary variable $\widehat{u}_{ij}$ to denote the manipulated dynamic line rating for the line $\{i,j\} \in \mathcal{E}_D$. The attacker is subject to following constraints:

$$\forall\ \{i,j\} \in \mathcal{E}_D \quad \begin{cases} \widehat{u}_{ij} = u^a_{ij} \\ u^{min}_{ij} \leqslant \widehat{u}_{ij} \leqslant u^{max}_{ij}, \end{cases} \tag{12}$$

because, the attacker wants to maintain a level of stealthiness, so he does not choose an out-of-bound DLR value, that may set off an alarm.

If the DLR value of a line $\{i,j\} \in \mathcal{E}_D$ is manipulated by the attacker, then the EMS software will obey the following constraint for the power flow on the line:

$$\forall\ \{i,j\} \in \mathcal{E} \quad |f_{ij}| \leqslant \widehat{u}_{ij}. \tag{13}$$

We pose the problem of optimal attack generation – from the attacker's viewpoint – as the following bilevel optimization problem:

$$\max_{u^a} \quad U_{cap}(f; u^d) \quad = \max_{\{i,j\} \in \mathcal{E}_D} 100 \left(\frac{|f_{ij}|}{u^d_{ij}} - 1 \right)_+ \tag{14a}$$

$$\min_{p,\theta} \quad C(p) \quad \text{s.t. } (1)-(6), (12), (13), \tag{14b}$$

where $a_+ := \max(a, 0)$. This problem is equivalent to a 2-stage sequential (Stackelberg) game, in which the attacker (leader) chooses his strategy assuming a best response from the defender (follower). Specifically, in the first stage, the attacker chooses the incorrect DLR ratings u^a (or equivalently $\widehat{u}_{ij}$) that are subsequently implemented in runtime by localizing and corrupting true DLR values in the nonlinear ED controller's memory (Section VI). The attacker's objective is to maximize the maximum percentage capacity bound violation of the power flows f_{ij} on lines $\{i,j\} \in \mathcal{E}_D$ over the true DLR values u^d_{ij} after the defender responds optimally in the second stage. This objective can be expressed as $U_{cap}(f; u^d)$ in (14a). In the second stage, the defender chooses the generator outputs p and voltage phase angles θ that achieves min-cost solution to DC-ED, i.e., minimize the generation costs (2) subject to the constraints (1)-(6),(12),(13). The attacker ensures that under the manipulated DLR ratings $\widehat{u}_{ij}$ for lines $\{i,j\} \in \mathcal{E}_D$ and given static ratings u^s_{ij} for lines $\{i,j\} \in \mathcal{E}_S$, there exists a feasible flow allocation that minimizes the generation cost (2), otherwise the SO will be require to setting off an alarm causing the SO to initiate other actions such as load curtailment.

Note that the actual generation cost faced by the operator when incorrect u^a are used in the SO's nonlinear ED formulation will be different than the defender cost obtained in the stage 2 subgame. In fact, the nonlinear ED is likely to be infeasible in the sense that the power flows on certain lines can exceed the permissible line ratings.

The attack model can be summarized as follows. The physical system consists of the physical components, e.g., generators, transmission network, and the loads. Each of these components send data to the EMS via means of SCADA, which is part of the attacker knowledge. The generators submit the cost functions, the transmission network submits the topology and the line ratings, and the loads submit the demand. The attacker uses this data to compute a DLR manipulation based on his attack policy, and then compromises the DLR values utilized by the EMS while solving the ED problem. Finally, the EMS implements the false ED solution by dispatching the new generation set-points to the individual generators.

Next, we present our computational approach to compute the optimal maximin attack.

III. Characteristics of Optimal Attack

The optimal attack generation problem posed in (14) is a linear-quadratic bilevel (LQBP) that is, in general, computationally hard to solve. One of the standard approaches to solve a LQBP is to reformulated it as a Mixed Integer Linear Program (MILP), which can be implemented using commonly available optimization solvers.

Our approach for solving the bilevel optimization problem (14) is as follows. First, we divide the main problem as $2\,|\mathcal{E}_D|$ parallel optimization problems where the attacker's objective is to just maximize the capacity violation of one DLR line, in either flow direction. This converts the attacker's objective function from nonlinear to an affine function. This subproblem can be represented as follows:

$$\begin{aligned} \max_{x \in X} \quad & {g_1}^T x + {g_2}^T y^\star \\ \text{s.t.} \quad & A_1 x + B_1 y^\star \leqslant k_1 \\ y^\star \in \min_{y} \quad & \frac{1}{2} y^T H y + {h_1}^T y + h_2 \\ & \text{s.t.} \quad A_2 x + B_2 y \leqslant k_2, \end{aligned} \tag{15}$$

where x denotes the attacker actions; X denotes the non-negativity and/or integrality constraints. In the subproblem of (14), $x = u^a$, $y = (p, \theta)$, $X = \{u \in \mathbb{R}^{\mathcal{E}_D} : u^{min} \leqslant u \leqslant u^{max}\}$. Also, g_1, B_1 are zero vector and zero matrix, respectively.

Second, we note that, for fixed attacker action x, the inner problem is a convex minimization problem, and therefore strong duality applies. Applying the Karush-Kuhn-Tucker (KKT) conditions for the optimal solution of the inner problem, we can pose the overall bilevel problem as a MILP [35]. Let, for fixed attacker action x, $(y^\star, \lambda^\star)$ denote the optimal primal-dual pair for the inner problem. Then the KKT optimality conditions are as follows.

$$A_2 x + B_2 y^\star \leqslant k_2 \tag{16a}$$

$$\lambda^\star \geqslant 0 \tag{16b}$$

$$H y^\star + {B_2}^T \lambda^\star + h_1 = 0 \tag{16c}$$

$$\lambda^\star \leqslant M(\mathbf{1}_n - \mu)$$

$$A_2 x + B_2 y^\star - k_2 \leqslant M\mu \tag{16d}$$

$$\forall\ i \in \{1, 2, \cdots, m\}, \qquad \mu_i \in \{0, 1\},$$

where $m = length(k_2)$, M is infinity (chosen as a significantly large number). (16a), (16b), (16c) and (16d) are primal feasibility, dual feasibility, stationarity and complementary

slackness conditions. Note that the complementary slackness conditions are reformulated into integrality constraints.

Thus, the bilevel subproblem can be restated as a single-level mixed-integer linear program (MILP).

$$\begin{aligned} \max_{x \in X} \quad & {g_1}^T x + {g_2}^T y^\star \\ \text{s.t.} \quad & A_1 x + B_1 y^\star \leqslant k_1, \text{ and (16).} \end{aligned} \tag{17}$$

Third, we solve for $2\,|\mathcal{E}_D|$ copies of the above MILP (17), and choose the maximum over all DLR lines, the non-negative percentage capacity bound violation, in either flow direction.

Algorithm 1 Optimal security strategy

```
(U*_cap, u^a*) ← GETOPTIMALATTACK()
procedure GETOPTIMALATTACK()
    U*_cap ← 0, u^a* ← 0
    m = GETMILPMODEL() using (17)
    for {i, j} ∈ E_D do                          ▷ for each DLR line
        for dir ∈ {−1, 1} do                     ▷ for each flow direction
            SETOBJECTIVE(m, 100 ((dir × f_ij)/u^d_ij − 1))
            SOLVE(m)
            U_cap ←GETOBJECTIVEVALUE(m)
            u^a ← GETVALUE(m,u^a)
            if U_cap > U*_cap then
                (U*_cap, u^a*) ← (U_cap, u^a)    ▷ update values
            end if
        end for
    end for
    return U*_cap, u^a*
end procedure
```

Our approach is summarized in Algorithm 1. The procedure GETOPTIMALATTACK() initializes the optimal attacker strategy and optimal attacker gain to zero. It constructs the MILP model with the KKT conditions for the inner problem and the feasibility constraints for the outer decision variables, by calling the procedure GETMILPMODEL(). Then, for each DLR line and each flow direction, GETEDGEATTACK sets the objective function as the percentage capacity violation for that line. During each iteration, if the attacker's gain computed is larger than the previously computed value, then the values for the optimal attacker's gain and the corresponding optimal attack strategy are updated. As we will see in Section IV-B, this computational approach is indeed scalable to larger networks.

IV. Computational Results

We discuss the structure of optimal attacks on benchmark power networks with DLRs, and discuss its implications on line capacity violations and increased generation costs.

A. 3-node Example

We now illustrate the optimal attacker strategy with the help of a benchmark example. We consider a 3-node network as shown in Figure 3. It consists of 2 generators G_1, G_2 at bus 1 and 2, respectively, and a load L on bus 3.

The following assumptions enable the computation of optimal attack in closed form. The nominal voltage magnitude is $V^{\text{nom}} = 230\ kV$ and the upper and lower voltage bounds are given by $\overline{V} = 1.1V^{\text{nom}}, \underline{V} = 0.9V^{\text{nom}}$, respectively. The three lines are identical, each with impedance $z = 0.002 + 0.05\mathbf{j}$

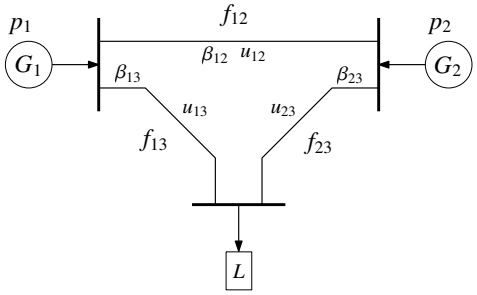


Fig. 3: Three-bus power system.

u^d_{13}	u^d_{23}	u^a_{13}	u^a_{23}	f_{13}	f_{23}	U_{cap} (in 10^5\$)
130	120	100	200	100	200	80
130	150	200	100	200	100	70
160	150	100	200	100	200	50
160	180	200	100	200	100	40

TABLE I: Optimal attacker strategy for three-bus test case.

in per unit system. Thus, the susceptance of each line is the inverse of reactance given by $\beta = \frac{1}{0.05}$. Assume that for the given instance, the active DLR for each of the three lines is 160 MW. The generation output of the two generators must satisfy the bounds $0 \leqslant p_1, p_2 \leqslant 300$ MW. Bus 3 has a constant power load having demand $d = 300$ MW.

Consider, for simplicity, a linear power flow model (4)-(5), and the linear cost of generation given by

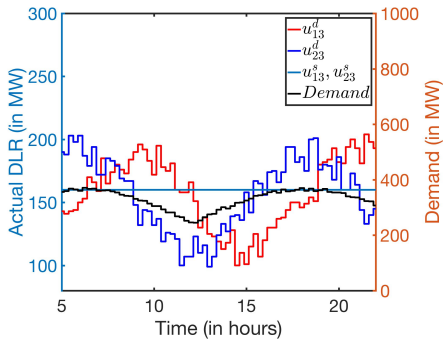
$$C(p) = b_1 p_1 + b_2 p_2, \tag{18}$$

where we choose $b_1 = 2b_2 = 2b > 0$. Simplifying further, we get, $C(p) = b_1 p_1 + b_2(d - p_1) = bp_1 + bd$.

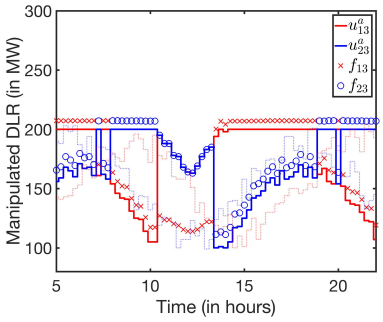
In the "no attack" case, the optimal generation turns out to be $(p_1, p_2) = (120, 180)$. The power flows at this point are $f_{12} = -20, f_{13} = 140$, and $f_{23} = 160$, respectively. As a result, the most congested line among all the three lines is line $\{2, 3\}$. This is expected as the G_2 has lower cost of production, so it generates more causing the congestion in line $\{2, 3\}$.

Assume for the sake of illustration that only the DLRs of lines $\{1, 3\}$ and $\{2, 3\}$ can be manipulated. The attacker's strategy will be either to maximize the capacity violation on line $\{2, 3\}$ (strategy A) or that on line $\{1, 3\}$ (strategy B). The attacker's optimal strategy is the one which leads to larger of these two violations. Assuming that the demand is fixed at 300, under strategy A (resp. strategy B), the optimal manipulated DLRs will be $u^a_{13}, u^a_{23} = (100, 200)$ (resp. $(200, 100)$). Table I lists some possible combinations for the actual DLR values of lines $\{1, 3\}$ and $\{2, 3\}$, and the corresponding optimal attacker strategies. For example, if $(u^d_{13}, u^d_{23}) = (120, 120)$, then the optimal attacker strategy is strategy A, i.e. $(u^a_{13}, u^a_{23}) = (100, 200)$, which yields attacker objective value as $U_{cap} = 80$.

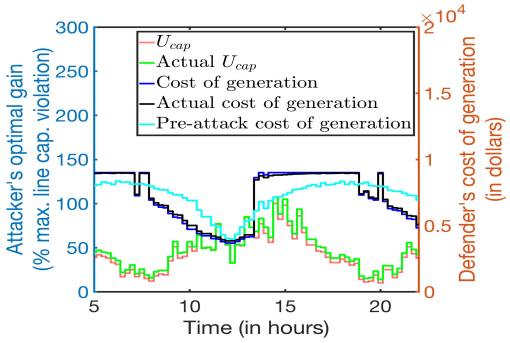
Now let us use the aforementioned approach to generate optimal DLR manipulations when the demand and DLRs vary over time, and OPF calculations account for manipulated line ratings to generate power flow allocations. For the 3-node network (Figure 3), consider the demand pattern at node 3 and the representative DLR for two lines $\{1, 3\}$ and $\{2, 3\}$ as



(a) Possible DLR and demand pattern over 24 hour horizon.



(b) Time of attack. The actual DLR ratings u^d are shown as lightly dashed lines in the background for comparison.



(c) Attacker's gain and SO's cost of generation as predicted by the bilevel model (14), and as computed by MATPOWER.

Fig. 4: Results for three-node power grid.

shown in Figure 4a. We instantiate the OPF models at every 15 minutes using this demand pattern. The aggregate demand curve has two peaks corresponding to the morning and evening peak periods. We chose the lower and upper bounds for the DLR values to be 100 and 200 MW. Then we varied DLRs between these bounds to generate patterns for 24 hour period. For the sake of illustration, we consider the two DLR curves to have sinusoidal patterns with certain offset between the two. The pattern also models the increased capacity due to favorable conditions (e.g. wind) during certain parts of the day. For these DLR and demand patterns, we determine how the attacker strategy and the attacker's gain varies over time with respect to the true DLRs and the demand.

Figure 4b shows the non-linear power flows along the DLR lines when the attacker's DLR ratings are in effect. We observe that the non-linear power flows are greater than the attacker's DLR ratings because of the presence of the reactive power which is not accounted by the linear power flow model assumed by the attacker in generating the optimal attack.

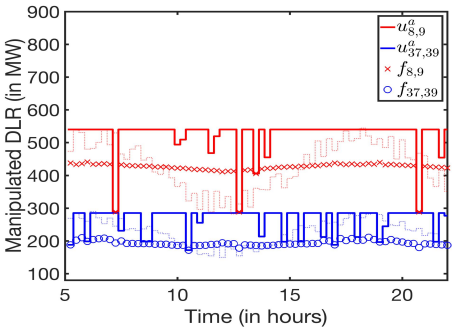
We also note that if the attacker targets line $\{2,3\}$ (strategy A), then the optimal attack can reach to maximum DLR rating, i.e., u^a_{23} can assume the value u^{max}_{23} for certain time periods. Recall that the bilevel formulation is constrained by the supply-demand balance in the defender's response. This constraint becomes tight for a range of time-periods during which the optimal attack u^a_{13} tracks the power flow f_{13} on line $\{1,3\}$. If the true DLRs are such that $u^d_{23} > u^d_{13}$, then the attacker chooses $u^a_{23} = u^{max}_{23}$. To ensure that the supply = demand constraint is met u^a_{23} is just equal to the power flow required to flow on line $\{1,3\}$. On the other hand if $u^d_{23} < u^d_{13}$, then the optimal attacker strategy is to violate the capacity of line $\{1,3\}$ (strategy B).

We evaluated the attacker's gain (U_{cap}) and the defender's cost of generation both estimated by the bilevel formulation (14) and by the nonlinear computations using MATPOWER (see Figure 4c). The respective curves closely follow each other. The actual cost of generation under nonlinear power flows is slightly larger than the cost of generation estimated under linear power flows. The same is also true for the attacker's gain U_{cap}. Comparing the demand and DLR variations in Figure 4a and the objective functions in Figure 4c, we can see that the optimal attacker gain is not achieved when the network experiences heavy demand. Rather, the optimal gain is achieved when the network is heavily congested, i.e., relative to the network's capacity, the aggregate demand is high. This gives an important insight into the optimal time for the attack. For e.g., during the hot summers and low windy conditions, the lines have lower capacities than during the winters. Also, the high temperatures lead to more aggregate demand during the summers. Hence, the attacker is better off manipulating the DLRs in high temperature conditions.

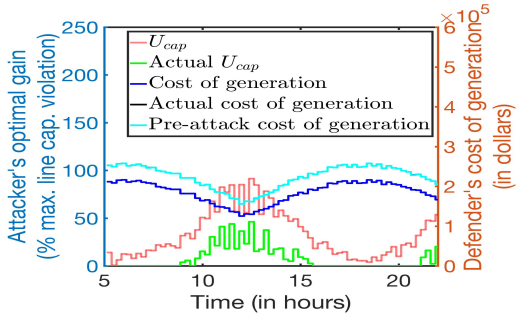
B. Scalability of attack

To demonstrate the scalability of our approach, we implemented Algorithm 1 on an 118-node network. We choose the DLR and demand patterns for the 118-node network similar to the ones in 3-node network, but in contrast to the linear generation cost (18), we adopt the more realistic convex quadratic cost function (3). In this paper, we have used Gurobi which is a state-of-the-art optimization toolbox and has built-in support for solving MILP problems. Figures 5a and 5b show the corresponding computational results for an 118 node network. Due to the fact that actual power flows also consist of reactive power flows in addition to real power flows, there are higher line losses, resulting in more total power generation that increases the cost of generation. However, we see that the actual attacker's gain is lower than the estimate obtained by solving (14) (Figure 4c). This can be explained as follows. The generators have different quadratic curves for the cost of generation. As a result for lower network load, one set of generators may be more contributing to the generation, but for higher loads, other set of generators may be the more contributing ones. This results in lower power flows along the DLR lines during high demand conditions. Hence, in the case of low aggregate demand, the DLR lines are violated to a larger extent than in the case of high demand. Another important observation is that the attacker's gain can be high even if the demand is low, because the actual DLRs may be even lower.

In the next section, we describe how an attacker can implement the optimal attack as computed by the bilevel formulation (14), as a cyberattack targeting the EMS soft-



(a) Time of attack for 118-node power network.



(b) Loss functions for 118-node network.

Fig. 5: Results for 118-node network

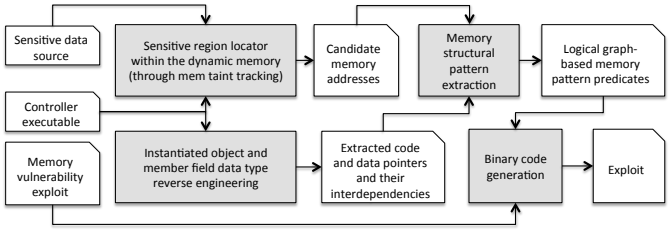


Fig. 6: Flowchart for attack implementation.

wares. Specifically, we will show how an EMS software (e.g., PowerWorld[3]) be targeted such that the values of the DLRs in the memory of the software will change during run-time. This will cause the ED implementation in the EMS to yield a false ED solution.

V. Implementations

We implemented our proposed attack in real controller software packages. Figure 6 shows the stages of the implemented attack. Initially, we assume a controller executable file (vulnerable point) and sensitive data sources (e.g., inputs such as DLRs originating from an external source) are given. Next, through memory taint analysis, we narrow down our search space to identify the the memory regions where the sensitive parameters may reside in memory during the controller execution. Accordingly, all the memory regions affected by the target input are marked (tainted). The tainted areas are then searched for the values of interest (e.g., target DLRs), and candidates are shortlisted. To identify the correct candidate from the set of candidates, we generate structural memory pattern signatures around the correct candidates during the offline binary analysis phase. We use our past work [26] to extract binary-level data type and code, and data pointers and their interdependencies (discussed below). Given the reverse engineered logical memory layout, we create structural patterns of the memory regarding where the target parameters reside. Those patterns are then used to generate the exploit binary. During the attack phase the exploit searches the dynamic memory address space to locate the target parameters using the patterns. Finally, it changes the identified parameter values to the optimal attack values, as discussed in Section III.

Every control algorithm implementation by controller software executables involve code and data. The code instructions encode the algorithm logic (e.g., iterative optimization loops), whereas the data stores the controller parameters such as the OPF constraints and DLRs. Modification of the code instructions are often infeasible due to $W \oplus X$ protections. However, the data regions should be (and are set as) writable, because the EMS operators often update their values dynamically according to the most recent power system configuration.

Maintenance of control-sensitive variable values such as DLRs by the controller software provides an attack surface to modify them in memory space during the attack. Our investigations of EMS software binaries showed heavy use of data structures and class objects to store those values that are used directly by OPF. During the offline phase, we analyzed the EMS software binary to determine its memory's structural layout. We are interested in structural information such as the allocated class instances (objects), the class hierarchy, and the logical interdependencies between the instantiated objects within the memory, e.g., cross-object code and data pointers. We are not interested in exact object memory addresses, because the addresses will likely differ during the attack due to unpredictable (inputs and hence) dynamic execution paths. Instead, by capturing the logical interconnections among the instantiated memory-resident objects, we extracted invariants about their interdependencies that remain the same across different runs. The attacker later uses the invariants during the attack to locate (and corrupt) the DLR values.

Search for a specific DLR value during the attack results in several memory-resident candidates that are mostly (except one) false positives. To identify the correct candidate, our implementation uses the invariants, expressed as propositional logic predicates, that capture the logical memory structural patterns around the target DLR parameters. We use three kinds of memory patterns: address-relative intra-class type patterns, code pointer-instruction patterns, and data pointer-

[3]We have taken the necessary responsible disclosure steps and have informed the vendors about our research findings. It is noteworthy that we are not reporting a security software vulnerability in this paper. Instead, assuming there is a potential exploit, we demonstrate how the adversaries can perform domain-specific data corruption in memory to impact the produced control actuation commands. The steps are not specific to any commercial software package.

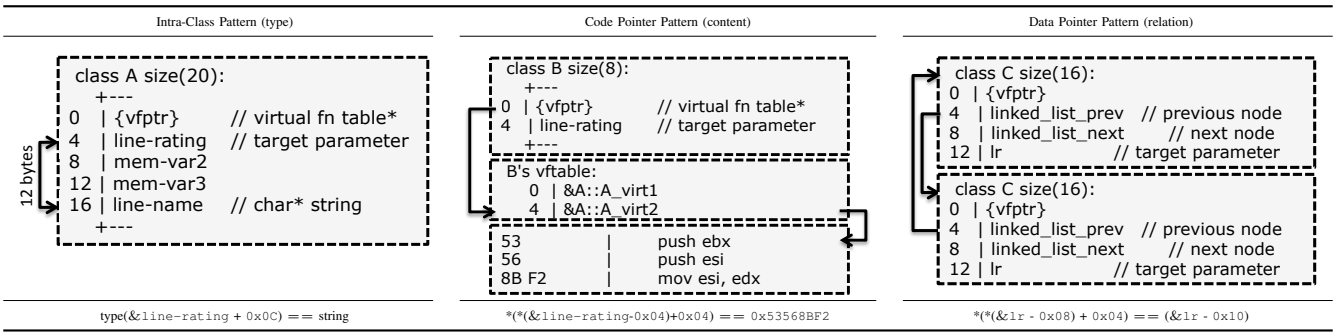
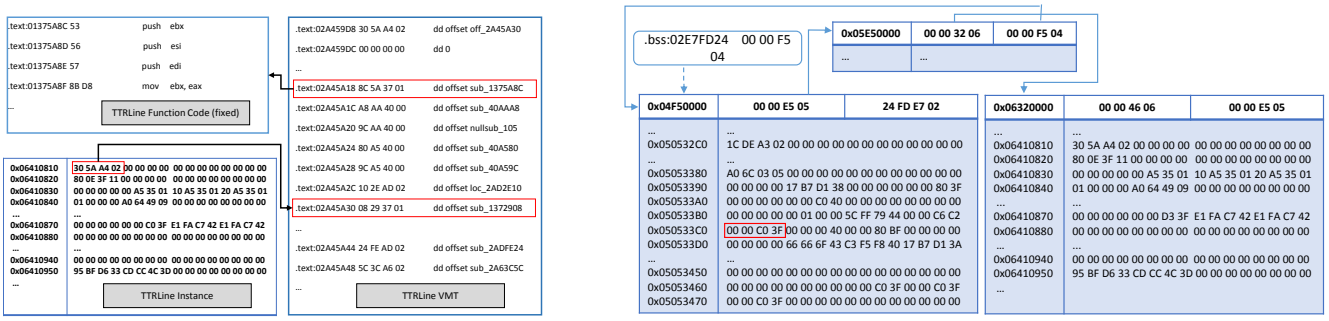


TABLE II: Logical memory structure signatures for critical parameters.



(a) Code pointer-instruction pattern.

(b) Linked-list as data pointer-based pattern.

Fig. 7: Code and data pointer-based structural memory patterns in PowerWorld used for graphical predicate generation.

based patterns (Table II).

Address-relative intra-class type patterns. The attack extracts execution-agnostic memory structural patterns around the target DLR values in memory. We concentrate on intra-class patterns that capture fixed offset relations among members of the same class as the target DLR parameter, and their types and/or values. If the DLR parameter is stored as a member of a class that also contains other variable(s), whose type is (are) easy to identify, we use that information as a local signature for the target parameter. In memory forensics, types such as character strings, pointers [16], and fixed-value member fields can be identified simply. We investigate the vicinity of the target parameter within the same object looking for addresses that store easy-to-identify data types. If one or more of such samples are found, their type/value and corresponding offset from the target parameter address is used to produce the signature. The attack creates simple-to-check logical predicates for each candidate (e.g., "candidate_addr + $0x08$ stores $0x00000001$"). Our implementation aggregates the produced predicates into a single conjunctive logic signature.

Code pointer-instruction patterns. We leverage the code pointer relations within the memory regions to extract invariants (logical predicates) about the structural memory layout around the target DLR parameters. We extract such invariants given the reverse engineered class object pointers, and their logical interdependencies with the corresponding member and virtual functions. We use the fact that code segments (e.g., instructions of member and virtual functions) within the controller software binary are typically set as read-only with fixed content. Table II shows a sample code pointer-based predicate for the illustrated pattern. The signature checks whether the first four byte content of the target parameter's object's second virtual function is equal to the corresponding function prologue. As denoted, the signature does not depend on the absolute address values given the target parameter candidate's location. The attack can automatically generate the code pointer patterns for the object's individual member and virtual functions. Finally, the generated predicates are combined into a single conjunctive logical predicate to check against all the identified candidates within the EMS memory space attack time.

Data pointer-based patterns. The data pointer-based patterns do not often assume fixed data values in memory, and is purely based on memory structure and the relations between various objects. We perform a recursive pointer traversal among the recognized objects on the controller's memory space following its earlier forensics analyses of the allocated objects and the stored pointer values within them (member fields). The algorithm implements a depth-first search starting from individual recognized pointers within the memory space. For each pointer under the consideration, we determine if its destination is an memory-resident object. If so, the attack recursively traverses all the member pointer fields within the destination object. During its recursive search, our implementation generates the corresponding directed graph, where nodes represent allocated objects, and the outgoing edges indicate the member pointer fields within the source object. The generated directed graph represents the inter-object dependencies within the memory space. Once the generation of the graph in completed, our implementation searches for cycles. Such cycles are very

Param. values	#Hits	#Relevant	#Recognized	Accuracy
0x3FC00000	143	3	3	100%
0x02A45A30	2038	4	4	100%
0x06410570	30	1	1	100%
0x06410810	30	1	1	100%
0x06410810	28	1	1	100%

TABLE III: The target parameter value recognition accuracy.

EMS Software	vfTable	Line	Bus	Gen.	Accuracy
PowerWorld	8527	3	3	2	100%
NEPLAN	6549	51	30	5	100%
PowerFactory	110	34	39	10	100%
Powertools	3	185	118	53	100%
SmartGridToolbox	194	79	57	4	100%

TABLE IV: Memory layout (object) forensics accuracy. The instances were correctly marked with their types.

popular in widely used data structures such as linked lists (the rightmost entry on Table II). The attack turns each cycle within the graph into a logical predicate that corresponds to a data pointer-based signature.

VI. Empirical Attack Deployment Results

To assess the proposed attack feasibility in practice, we implemented it against widely-used commercial and open-source industrial controller software packages. The implemented attack involves the following steps: *i)* during the offline phase, we reverse engineer the EMS software binary to locate DLR parameters within the controller and create the corresponding invariants that hold true regardless of their absolute memory addresses; *ii)* during the online phase (attack time), the exploit searches the controller memory for the known legitimate DLR values and collects the candidates; *iii)* the attack recognizes the only true candidate by applying the invariants on the collected set of candidates; and *iv)* our implementation modifies the value maliciously according to the optimal attack generation algorithms discussed in the previous section. We now explain the results for our empirical validation.

A. EMS Software Attack

We validated the proposed attack on real-world widely-used industrial controller software packages. We first present the detailed results on PowerWorld, and later compare the attack's performance for other controllers (NEPLAN, PowerFactory, PowerTools, and SmartGridToolbox).

Figure 7a shows a generated code pointer-based memory signature in PowerWorld. The corresponding pattern predicate for runtime memory search was "*(*(candidate_addr - 0x54) - 0x24) == 0x5356578B", where 0x5356578B is the hex representation of the sub_1375A8C function's first four instruction bytes. The rating of every transmission line is stored in offset 0x24 of the corresponding TTRLine object. The information about the transmission lines of the power system is stored as a doubly linked list of TTRLine objects in PowerWorld memory space. The attack used "*(*(candidate_addr - 0x24) + 0x04) == (candidate_addr - 0x24)" as the pattern predicate for line ratings. Let us call the linked list node that stores the target line rating A. The pattern predicate above essentially verifies the following linked list invariant: whether A's previous node's next pointer points to A. More complex patterns can be extracted if needed; however, our empirical studies on PowerWorld shows simple patterns always suffice to identify and isolate the exact candidate uniquely.

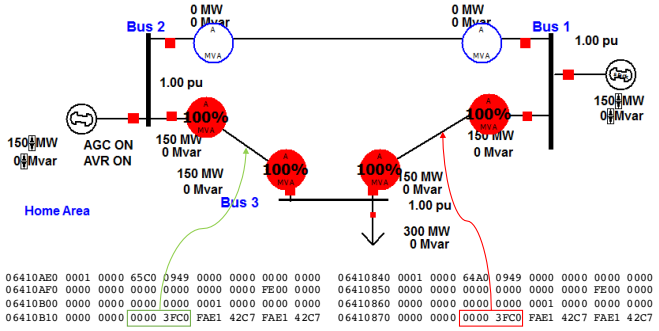
Figure 7b shows another PowerWorld data pointer pattern for line ratings. PowerWorld allocates linked list nodes (0x13FFF0 sizes each) allocated by VirtualAlloc for objects instances of different classes (e.g., TGen, TBus and TTRLine). Only three nodes are shown. If our objective is to look for line rating 0x3FC00000, its corresponding pattern predicate will encode the offset to get the node's initial member value 0x05E50000 that points to the next node shown (summarized) on the top of the figure. The second element of each node (0x04F50000 in the top node) points to the previous node. A relatively more complex second-degree predicate would be "*(*(*(*(candidate_addr - 0x1033C0)) + 0x04)+ 0x04) == candidate_addr - 0x1033C0", i.e., $A \rightarrow next \rightarrow next \rightarrow previous \rightarrow previous == A$, where A represents the data structure that stores the line rating 0x3FC00000.

The attack payload checks for patterns on the identified candidates before corrupting their values. The code searches for the specific value in memory, and modifies the identified candidate. Table III shows how many hits our implementation finds for individual target power system parameter values on PowerWorld memory space. The number empirically proves the infeasibility of memory corruption attacks without the use of signature predicates. The next column shows how well the signatures dismiss the irrelevant candidates and identify the true target values. Table IV shows the forensics analysis accuracy for five different EMS software packages. Through the use of the code pointer signatures and its extracted knowledge about the class hierarchies, our implementation was able to correctly recognize the class types of all object instances within the EMS memory. The payload initializes the OPF algorithm in its corresponding thread. Once it changes the identified memory addresses, it restarts the control loop through the call to CreateThread function within kernel32.dll that is loaded by almost all windows processes.

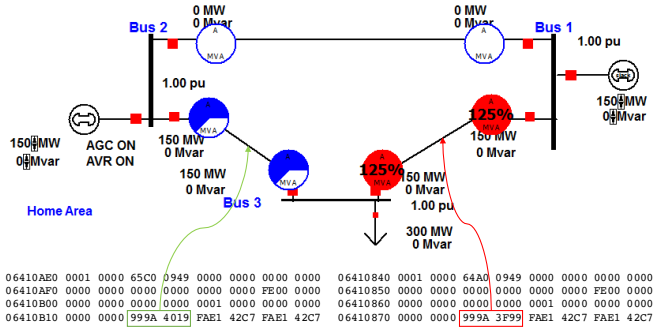
B. Case-study Demonstration

As a concrete example, we show how the state of underlying power system (the same model used in Section IV) gets affected once the memory corruption is completed (Figure 8[4]). Before the corruption (Figure 8a), the EMS GUI visualizes the safe state of power system operation, where the transmission lines are mostly fully utilized; however, no line rating (capacity constraints) are violated. The optimal attack generation algorithm computes the adversary-optimal values for the line ratings, and chooses to *i)* modify the $B1-B3$ transmission line to $120MW$ from $150MW$; and *ii)* modify the line rating for the $B2-B3$ transmission line to $240MW$ from $150MW$. While implementing the optimal attacker strategies that we obtain from the maximin solution, we need to translate the line rating values to higher values using basic power flow

[4]The pie charts on the transmission lines represent the used percentages of the line power flow capacities in that particular state.



(a) PowerWorld pre-attack power system state (safe).



(b) PowerWorld post-attack power system state (unsafe).

fbus	tbus	r	x	b	rateA	rateB	rateC	ratio	angle	status	angmin	angmax
1	3	0.0	0.05	0.0	150.0	9999.0	9999.0	0.0	0.0	1	-30.0	30.0
1	2	0.0	0.05	0.0	150.0	9999.0	9999.0	0.0	0.0	1	-30.0	30.0
2	3	0.0	0.05	0.0	150.0	9999.0	9999.0	0.0	0.0	1	-30.0	30.0

016B2AE0 0001 0000 0000 0000 2AC8 016B 0000 0000
016B2AF0 0000 0000 0000 3FF8 0000 0000 0000 0000
016B2B00 0000 0000 0000 3FF0 0000 0000 0000 0000
016B2B10 0000 0000 0000 0000 999A 9999 9999 3FA9
016B2B20 0000 0000 0000 0000 FFFF FFFF FFFF C033
016B2B30 0000 0000 0000 3FF0 0000 0000 0000 0000
016C0500 0003 0000 0000 0000 95B8 016B 0000 0000
016C0510 0000 0000 0000 3FF8 0000 0000 0000 0000
016C0520 0000 0000 0000 3FF0 0000 0000 0000 0000
016C0530 0000 0000 0000 0000 999A 9999 9999 3FA9
016C0540 0000 0000 0000 0000 FFFF FFFF FFFF C033
016C0550 0000 0000 0000 3FF0 0000 0000 0000 0000

(c) Powertools memory image of the sensitive parameters.

Fig. 8: PowerWorld and Powertools controller software attack results as the result of targeted adversary-optimal line rating manipulation.

calculations. For example, for the implementation of optimal attack, we use $\widehat{u}_{13} = 120\ MVA$ and $\widehat{u}_{23} = 240\ MVA$. These values are higher than the values $\widehat{u}_{13} = 100$ and $\widehat{u}_{23} = 200$ calculated by the bilevel optimization.

This increase in optimal line rating manipulations is necessary to account for the fact that the AC OPF implementation is constrained by the line rating bounds on apparent power flows (with both real and reactive power components) while the optimal attack generation procedure calculates manipulated line rating assuming that only real power flows are subject to line ratings. As the consequence, the power system enters an unsafe state after the OPF control algorithm uses the corrupted line ratings and hence produces wrong control outputs to the power generators; see Figure 8b. Optimal and physics-aware corruption of the sensitive values through a controller attack allows the intruders to maximize the physical impact on the power system operations without having to compromise a large number of sensors as required in false data injection attacks. We also performed the same memory data corruption attack on Powertools [1] package. In this scenario, the attacker changed the line rating for two of the branches as shown in Figure 8c. Similar to the PowerWorld case, the exploit locates the sensitive parameters (line ratings) and modifies them during the program execution. As the result, the memory corruption impacted the power flow iterations of DC-OPF performed by the Powertools software that consumed the modified memory regions, and made it converge to a different wrong value. In terms of the attack implementation approach, the attacks against PowerWorld and powertools were identical.

VII. Discussions and Potential Mitigation

Our attack and similar domain-specific memory data corruption attacks can be mitigated through several potential solutions: *i)* Protection of sensitive data: fine-grained data isolation mechanisms such as hardware supported Intel SGX can be leveraged to store and process sensitive data such as power system parameters within protection enclave regions. This protects sensitive data against access requests by other irrelevant instructions in the same memory space. A more fine-grained version of such memory-based data protection can distinguish between data that are often fixed during the operation (e.g., power system topological information) vs. regularly updated data regions (e.g., sensor measurements) to facilitate lower-overhead protection such as read-only memory pages for the fixed data once they are loaded on memory initially. *ii)* Control command verification: controller output verification mechanisms such as an extended version of TSV [19] can be used to ensure the safety of the (maliciously) issued control commands by an infected control system software before they are allowed to reach the actuators. Monitoring of the control channel, however, does not ensure the correct functionality of the control system software. Instead it just ensures its outputs (even though corrupted) are within the safety margins of the physical plant. *iii)* Intrusion-tolerant replication: a more traditional approach is to use redundancy such as N-version programming by maintaining a redundant controller software that is different from the main one used. The replica controller can monitor the dynamic behavior of the physical plant (e.g., power system) as well as the main controller's output to the actuators. The replica can rerun the control algorithm to calculate and compare its calculated control outputs with those of the main controller. Hence, the main controller infection (misbehavior) can be identified if a mismatch is detected; *iv)* Algorithmic redundancy: Carefully designed algorithmic tools (e.g., attack-aware optimal dispatch) can provide safe operating regimes to limit the impact of successful attacks. Indeed, this is a topic of future research.

VIII. Related Work

We review the most related recent work on control system security. The existing solutions to protect the control networks' trusted computing base (TCB) are insufficient as software patches are often applied only months after release [22], and new vulnerabilities are discovered on a regular basis [21], [28]. The traditional perimeter-security tries to keep adversaries out of the protected control system entirely. Attempts include regulatory compliance approaches such as the NERC CIP

requirements [31] and access control [11]. Despite the promise of information-security approaches, thirty years of precedence have shown the near impossibility of keeping adversaries out of critical systems [13] and less than promising results for the prospect of addressing the security problem from the perimeter [14], [15], [20]. Embedded controller software from most major vendors [14], [32] and popular human machine interfaces [20] have been shown to have fundamental security flaws. Offline control verification solutions [19] implement formal methods using symbolic execution of the controller program to verify the safety of the code before it is let execute on the controller device. Not surprisingly, those methods face scalability problem, caused by state-space explosion.

One specific related line of research is proposed false data injection (FDI) attacks [17], [30], [33] that have been explored over the past few years. FDI assumes compromised set of sensors and make them send corrupted measurements to electricity grid control centers to mislead the state estimation procedures. The authors propose a system *observability* [17] analysis to determine the required minimal subset of compromised sensors to evade the electricity grid's bad data detection algorithms [18]. The power system stability has also been studied under corrupted real-time pricing signals [29]. As a fundamental domain-specific monitoring tool for cyber-physical platforms, state estimation is to fit sensor data to a system model and determine the current state [2], [3]. Existing real-world solutions to analyze power system stability [12] run every few minutes [25]. These solution do not consider the cyber-side controllers and/or adversarial settings [4], [34]; hence they may miss malicious incidents such as the controller code execution attacks. Risk assessment techniques, e.g., contingency what-if analyses [27] investigate potential power system failures speculatively. However, enumeration of all *possible* incidents is a combinatorial problem and does not scale up efficiently in practical settings [7].

Acknowledgements

This work was supported by the NSF grants CNS-1239054, CNS-1453126, CNS-1453046, and the ONR grant N00014-15-1-2741.

References

[1] Powertools; available at http://hhijazi.github.io/PowerTools/, 2017.

[2] A. Abur and A. Expósito. *Power System State Estimation: Theory and Implementation*. Marcel Dekker, 2004.

[3] O. Alsac, N. Vempati, B. Stott, and A. Monticelli. Generalized state estimation. *IEEE Trans. on Power Systems*, 13(3):1069–1075, 1998.

[4] J. Arrillaga and B. Smith. *AC-DC Power Systems Analysis*. The Institution of Electrical Engineers, 1998.

[5] M. Assante. Confirmation of a Coordinated Attack on the Ukrainian Power Grid. SANS Industrial Control Systems Security Blog, 2016.

[6] D. Bienstock. *Electrical transmission system cascades and vulnerability - an operations research viewpoint*, volume 22 of *MOS-SIAM Series on Optimization*. SIAM, 2016.

[7] C. M. Davis and T. J. Overbye. Multiple element contingency screening. *Power Systems, IEEE Transactions on*, 26(3):1294–1301, 2011.

[8] Department of Energy. Dynamic Line Rating Systems for Transmission Lines; available at https://www.smartgrid.gov/files/SGDP_Transmission_DLR_Topical_Report_04-25-14_FINAL.pdf, 2016.

[9] Department of Energy. Improving Efficiency with Dynamic Line Ratings; available at https://www.smartgrid.gov/files/NYPA_Improving-Efficiency-Dynamic-Line-Ratings.pdf, 2016.

[10] N. Falliere, L. O. Murchu, and E. Chien. W32.Stuxnet Dossier. Technical report, Symantic Security Response, Oct. 2010.

[11] D. Formby, P. Srinivasan, A. Leonard, J. Rogers, and R. Beyah. Who's in control of your control system? device fingerprinting for cyber-physical systems. In *NDSS*, 2016.

[12] J. Glover, M. Sarma, and T. Overbye. *Power System Analysis and Design*. Cengage Learning, 2011.

[13] V. M. Igure, S. A. Laughter, and R. D. Williams. Security issues in scada networks. *Computers & Security*, 25(7):498–506, 2006.

[14] E. V. Kuz'min and V. A. Sokolov. On construction and verification of plc-programs. *Modelirovanie i Analiz Informatsionnykh Sistem [Modeling and Analysis of Information Systems]*, 19(4):25–36, 2012.

[15] T. G. Lewis. *Critical infrastructure protection in homeland security: defending a networked nation*. John Wiley & Sons, 2006.

[16] Z. Lin, X. Zhang, and D. Xu. Automatic reverse engineering of data structures from binary execution. In *Proceedings of Information Security Symposium*, page 5. CERIAS-Purdue University, 2010.

[17] Y. Liu, P. Ning, and M. K. Reiter. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC)*, 14(1):13, 2011.

[18] Z. Lu and Z. Zhang. Bad data identification based on measurement replace and standard residual detection. *Automation of Electric Power Systems*, 13:011, 2007.

[19] S. McLaughlin, S. Zonouz, D. Pohly, and P. McDaniel. A trusted safety verifier for controller code. In *NDSS*, 2014.

[20] T. H. Morris, A. K. Srivastava, B. Reaves, K. Pavurapu, S. Abdelwahed, R. Vaughn, W. McGrew, and Y. Dandass. Engineering future cyber-physical energy systems: Challenges, research needs, and roadmap. In *North American Power Symposium (NAPS)*, pages 1–6. IEEE, 2009.

[21] D. G. Peterson. Project Basecamp at S4. http://www.digitalbond.com/2012/01/19/project-basecamp-at-s4/, January 2012.

[22] J. Pollet. Electricity for Free? The Dirty Underbelly of SCADA and Smart Meters. In *Black Hat USA*, 2010.

[23] F. Rashid. Ics-cert: Response to cyber incidents against critical infrastructure jumped 52 percent in 2012. *Security Week*, 10, 2013.

[24] H. Sandberg, S. Amin, and K. H. Johansson. Cyberphysical security in networked control systems: An introduction to the issue. *IEEE Control Systems*, 35(1):20–23, Feb 2015.

[25] H. Singh and F. Alvarado. Network topology determination using least absolute value state estimation. *Power Systems, IEEE Transactions on*, 10(3):1159–1165, 1995.

[26] P. Sun, R. Han, M. Zhang, and S. Zonouz. Trace-free memory data structure forensics via past inference and future speculations. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 570–582. ACM, 2016.

[27] Y. Sun and T. J. Overbye. Visualizations for power system contingency analysis data. *IEEE Trans. on Power Systems*, 19(4):1859–66, 2004.

[28] L. Szekeres, M. Payer, T. Wei, and D. Song. Sok: Eternal war in memory. In *IEEE Symposium on Security and Privacy*, pages 48–62, 2013.

[29] R. Tan, V. Badrinath Krishna, D. K. Yau, and Z. Kalbarczyk. Impact of integrity attacks on real-time pricing in smart grids. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 439–450. ACM, 2013.

[30] R. Tan, H. H. Nguyen, E. Y. Foo, X. Dong, D. K. Yau, Z. Kalbarczyk, R. K. Iyer, and H. B. Gooi. Optimal false data injection attack against automatic generation control in power grids. In *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pages 1–10, 2016.

[31] U.S. Department of Energy Office of Electricity Delivery and Energy Reliability. North american electric reliability corporation critical infrastructure protection (nerc-cip), 2015.

[32] S. E. Valentine. *PLC code vulnerabilities through SCADA systems*. PhD thesis, University of South Carolina, 2013.

[33] Y. Wang, Z. Xu, J. Zhang, L. Xu, H. Wang, and G. Gu. Srid: State relation based intrusion detection for false data injection attacks in scada. In *European Symposium on Research in Computer Security*, pages 401–418. Springer, 2014.

[34] A. J. Wood and B. F. Wollenberg. *Power generation, operation, and control*. John Wiley & Sons, 2012.

[35] B. Zeng and Y. An. Solving bilevel mixed integer program by reformulations and decomposition. 2014.