Security Games for Cyber Resilient Bulk Power Systems

Gael Kamdem De Teyou
Postdoctoral Researcher, Dept. of Modeling, Simulation, and Visualization Engineering, ODU

Sachin Shetty
Associate Professor, Dept. of Modeling, Simulation, and Visualization Engineering, ODU
OUTLINE

• INTRODUCTION

• POWER GRID NETWORK ARCHITECTURE

• VULNERABILITY MULTI-GRAH

• TWO-PLAYER ZERO SUM-MARKOV GAME

• SIMULATIONS

• CONCLUSION
OUTLINE

• INTRODUCTION

• POWER GRID NETWORK ARCHITECTURE

• VULNERABILITY MULTI-GRAH

• TWO-PLAYER ZERO SUM-MARKOV GAME

• SIMULATIONS

• CONCLUSION
• Advanced
  • Attacker adapts to defenders’ efforts
  • Higher level of sophistication
  • Can develop or buy Zero-Day exploits

• Persistent
  • Attacks are objective and specific
  • Will continue until goal is reached

• Threats
  • Entity/s behind the attack
Critical Infrastructure

- Power Grid
- Water supply
- Transportation
- Information and Telecommunications
- Oil and gas
Introduction

December 2015 Ukraine power grid attack

• Hackers compromised corporate networks using spear-fishing emails with BlackEnergy trojan.

• Remotely, hackers took control of the SCADA network, switched off power substations and then disrupted electricity supply to the end customers.

• Destruction of files stored on servers and workstations.

• Denial-of-service attack on call-center to deny up-to-date information on the blackout
Introduction

2\textsuperscript{nd} cyber-attack on Ukraine power grid in December 2016

• Nearly a quarter of million people lost power in the Ivano-Frankivsk region of Ukraine.

• Hackers sent emails with infected attachments to power company employees, stealing their login credentials and then taking control of the power grid system to cut the circuit breakers at nearly 60 substations.

• The blackout lasted several hours
Introduction

2\textsuperscript{nd} cyber-attack on Ukraine power grid in December 2016

- Nearly a quarter of million people lost power in the Ivano-Frankivsk region of Ukraine.

- Hackers sent emails with infected attachments to power company employees, stealing their login credentials and then taking control of the power grid system to cut the circuit breakers at nearly 60 substations.

- The blackout lasted several hours
Introduction

Increase the resilience of Power Grid with R4 framework

- Increase the **Rapidity** by reducing the delay between the intrusion detection of the malware and the response of the defender;

- Increase the **Resourcefulness** by finding the appropriate vulnerable services to shut down

- Increase the **Robustness** by redirecting the malware into part of the system where critical assets are not accessible, and thus minimizing the impact of attacks
OUTLINE

• INTRODUCTION

• POWER GRID NETWORK ARCHITECTURE

• VULNERABILITY MULTI-GRAH

• TWO-PLAYER ZERO SUM-MARKOV GAME

• SIMULATIONS

• CONCLUSION
Power grid network architecture

Recommended defense-in-depth architecture for Industrial Control System [1]

Power grid network architecture

Execution of crafted code via web server

Power grid network architecture

Execution of crafted code via web server
Power grid network architecture

Execution of crafted code via web server
Power grid network architecture

Execution of crafted code via web server
Power grid network architecture

Execution of crafted code via web server
Power grid network architecture

Execution of crafted code via web server
Power grid network architecture

Execution of crafted code via web server
OUTLINE

• INTRODUCTION

• POWER GRID NETWORK ARCHITECTURE

• VULNERABILITY MULTI-GRAH MODEL

• TWO-PLAYER ZERO SUM-MARKOV GAME

• SIMULATIONS

• CONCLUSION
An edge vulnerability $e \in E$ is a directed edge from a node $v_1$ to a node $v_2$ which corresponds to a vulnerability hosted by an application on $v_2$ that the system rules allow to access from node $v_1$.

- $\Phi$ the set of vulnerabilities
- $\varphi(e) \in \Phi$, the vulnerability associated to $e$
- $v_2 = \gamma_{Head}(e)$ is the head of $e$
- $v_1 = \gamma_{Tail}(e)$ is the tail of $e$
Vulnerability Multi-Graph
Direct acyclic graph

Remote attacker

Control system
Control system DMZ
Corporate network
Corporate DMZ
Active vulnerabilities

<table>
<thead>
<tr>
<th></th>
<th>Φ₁</th>
<th>Φ₂</th>
<th>Φ₃</th>
<th>Φ₄</th>
<th>Φ₅</th>
<th>Φ₆</th>
</tr>
</thead>
<tbody>
<tr>
<td>violet</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>turquoise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>green</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>black</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>orange</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>blue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Attacker moves to $v_3$
- Vulnerable service associated to $Φ_1$ is disabled
Vulnerability Multi-Graph

Lateral movement

Active vulnerabilities

- Attacker moves to $v_{10}$
- Vulnerable service associated to $\Phi_2$ is disabled
Vulnerability Multi-Graph

Lateral movement

Active vulnerabilities

- Attack remains at node $v_{10}$
- Vulnerable service associated to $\Phi_3$ is disabled
Vulnerability Multi-Graph

Lateral movement

- Attacker moves to node $v_{14}$
- Vulnerable service associated to $\Phi_4$ is disabled
Active vulnerabilities

- Attacker is isolated at node \( v_{14} \)
- Vulnerable service associated to \( \Phi_4 \) is disabled
Vulnerability Multi-Graph

Lateral movement

- Attacker is isolated at node $v_{14}$
- Vulnerable service associated to $\Phi_4$ is disabled

Six vulnerabilities

<table>
<thead>
<tr>
<th>$\Phi_1$</th>
<th>$\Phi_2$</th>
<th>$\Phi_3$</th>
<th>$\Phi_4$</th>
<th>$\Phi_5$</th>
<th>$\Phi_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Red]</td>
<td>[Red]</td>
<td>[Red]</td>
<td>[Red]</td>
<td>[Red]</td>
<td>[Red]</td>
</tr>
</tbody>
</table>
OUTLINE

• INTRODUCTION

• POWER GRID NETWORK ARCHITECTURE

• VULNERABILITY MULTI-GRAH

• TWO-PLAYER ZERO SUM-MARKOV GAME

• SIMULATIONS

• CONCLUSION
Two-player zero-sum Markov Game

definition

A two-player zero-sum Markov game is defined as a 6-tuple \((S, A, O, P, R, \gamma)\) where:

- \(S = \{s_1 \ldots s_l\}\) is a finite set of game states;
- \(A = \{a_1 \ldots a_n\}\) is the set of actions of the maximizer (row player);
- \(O = \{o_1 \ldots o_m\}\) is the set of actions of the minimizer (column player);
- \(P\) is a Markovian transition model, with \(P(s, a, o, s')\) being the probability that \(s'\) will be the next game state when players take actions \(a\) and \(o\) respectively;
- The function \(R(s, a, o)\) specifies the immediate reward (or cost) of players for taking actions \(a\) and \(o\) in state \(s\);
- \(\gamma \in ]0, 1]\) is the discount factor for future rewards;
# Two-player zero-sum Markov Game

## Game matrix

### Immediate reward matrix for state $s \in S$

<table>
<thead>
<tr>
<th>Row player</th>
<th>Column player</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$R(s, a_1, o_1)$</td>
</tr>
<tr>
<td>$a_2$</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$a_n$</td>
<td></td>
</tr>
</tbody>
</table>

- $R(s, a_n, o_m)$

### Transition probabilities

- $P(s_0, a_1, o_1, s_1)$
- $P(s_0, a_1, o_2, s_1)$
- $P(s_0, a_2, o_2, s_2)$
Two-player zero-sum Markov Game

Player’s Policy

- A policy $\pi_A: S \rightarrow \Omega(A)$, for the row player (maximizer) is a function that gives for each state $s$ a probability distribution $\pi_A(s)$ over the maximizer actions $A = \{a_1..a_n\}$. For any policy $\pi_A$, $\pi_A(s, a)$ denotes the probability to take action $a$ in state $s$.

- For any policy $\pi$, $Q^\pi(s, a, o)$ is the expected sum of discounted reward of the row player:

$$Q^\pi(s, a, o) = \mathcal{R}(s, a, o) + \gamma \sum_{s' \in S} P(s, a, o, s') \min_{o' \in O} \sum_{a' \in A} Q^\pi(s', a') \pi'(s, a')$$

- Optimal policy $\pi$ and two Bellman functions:

$$W(s) = \max_{\pi_A(s) \in \Omega(A)} \min_{o \in O} \sum_{a \in A} Q(s, a, o) \pi'(s, a)$$

$$Q(s, a, o) = \sum_{s' \in S} P(s'| a, o, s) [\mathcal{R}(s, a, o, s') + \gamma W(s')]$$
Two-player zero-sum Markov Game

Value iteration algorithm

Value iteration \((S, A, O, P, \mathcal{R}, \gamma)\)

\[
W \leftarrow 0 \\
l \leftarrow 0 \\
\text{Repeat} \\
l + + \\
\text{For each} \ s \in S \text{ do} \\
W_{t+1}(s) = \max_{\pi_A(s) \in \Omega(A)} \min_{o \in O} \sum_{a \in A} \pi(s, a) \sum_{s' \in S} P(s'|a, o, s)[\mathcal{R}(s, a, o, s') + \gamma W_t(s')] \\
\text{Until} \ \forall s \in S, |W_{t+1}(s) - W_t(s)| < \epsilon \\
\text{For each} \ s \in S \text{ do} \\
\pi(s) \leftarrow \pi(s): \max_{\pi_A(s) \in \Omega(A)} \min_{o \in O} \sum_{a \in A} \pi(s, a) \sum_{s' \in S} P(s'|a, o, s)[\mathcal{R}(s, a, o, s') + \gamma W_t(s')] \\
\text{Return} \ \pi, W_{t+1}
Two-player zero-sum Markov Game

Application to lateral movement

- $S$ is a set of finite games, the attacker is the maximizer and the defender the minimizer.

- A unit game $s \subseteq S$ is completely defined by:
  - A node $v_s \subseteq V$ indicating the position of the attacker
  - A set of edges $A_s \subseteq E_{v_s} \subseteq E$ adjacent to $v_s$
  - And a set of active vulnerabilities $O_s \subseteq \Phi$.

- $n_s = |A_s|$ is the number of active edges of state $s$.
- $m_s = |O_s|$ is the number of active vulnerabilities of state $s$.
Two-player zero-sum Markov Game

The attacker exploits edge $e_i \in A_s$ and the defender shut down the application associated to vulnerability $\varphi_j \in O_s$.

- $\varphi_j = \varphi(e_i)$:
  - The efforts of the attacker are in vain
  - The immediate reward of the attacker is $R_A(s, e_i, \varphi_j) = \zeta(\varphi(e_i)) < 0$
  - The attacker stays at the same mode.

- $\varphi_j \neq \varphi(e_i)$:
  - The attacker exploits successfully edge $e_i$ and moves forward to next node
  - The immediate reward of the attacker is $R_A(s, e_i, \varphi_j) = \zeta(\varphi(e_i)) + A_t[Y_{Head}(e_i)] \geq 0$
  - The attacker moves to the next node.
Simulation
Exploit cost of vulnerabilities

Base metrics of the Common Vulnerability Scoring System (CVSS) [3]

<table>
<thead>
<tr>
<th>Access Vector $A_v(\varphi)$</th>
<th>Describes how close the attacker must be to exploit the vulnerability $\varphi$</th>
<th>Local</th>
<th>0.395</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Adjacent network</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Remote network</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Access Complexity $A_c(\varphi)$</th>
<th>Describes how easy or difficult it is to exploit the vulnerability $\varphi$</th>
<th>High</th>
<th>0.395</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Medium</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low</td>
<td>0.71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Access Authentication $A_a(\varphi)$</th>
<th>Describes the number of time an attacker must authenticate to exploit the vulnerability $\varphi$</th>
<th>Multiple</th>
<th>0.45</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Single</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>None</td>
<td>0.704</td>
</tr>
</tbody>
</table>

| CVSS Score $20.A_v(\varphi).A_c(\varphi).A_a(\varphi)$ | By construction, $1.4 \leq CVSS(\varphi) \leq 10$ |       |      |

The edge cost is a function $\zeta$ over the set of edges $E$ which measures the amount of effort required to exploit an edge vulnerability:

$$\zeta: E \rightarrow [-11, -1]$$

$$e \rightarrow \zeta(e) = CVSS(\varphi(e)) - 11$$

![Graph showing exploitability vs edge cost](image-url)
The attractiveness of a node $v \in G$ measures its appeal to cyber attack.

<table>
<thead>
<tr>
<th>Typical nodes</th>
<th>Layer</th>
<th>Severity of cyber-attacks</th>
<th>Impact on power grid</th>
<th>Features</th>
<th>Attractivity $A_t(v)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application servers</td>
<td>Control system</td>
<td>Critical</td>
<td>Availability, Integrity, Confidentiality</td>
<td>▪ Access to substation’s controllers in real time</td>
<td>100</td>
</tr>
<tr>
<td>Database servers</td>
<td>Control system</td>
<td>High</td>
<td>Confidentiality, Integrity</td>
<td>▪ Control algorithms and control commands</td>
<td></td>
</tr>
<tr>
<td>Engineering workstations</td>
<td>Control system</td>
<td>Low</td>
<td>Confidentiality</td>
<td>▪ Power transmission planning</td>
<td></td>
</tr>
<tr>
<td>Historian database</td>
<td>Control system</td>
<td>Low</td>
<td>Confidentiality</td>
<td>▪ Power grid sensor’s data</td>
<td></td>
</tr>
<tr>
<td>Web servers</td>
<td>Corporate network</td>
<td>Medium</td>
<td>Confidentiality</td>
<td>▪ Copy of control system data</td>
<td></td>
</tr>
<tr>
<td>Authentication servers</td>
<td>Corporate network</td>
<td>Medium</td>
<td>Confidentiality</td>
<td>▪ Business data (billing, power consumption, etc...)</td>
<td></td>
</tr>
<tr>
<td>Business servers</td>
<td>Corporate DMZ</td>
<td>Low</td>
<td>Confidentiality</td>
<td>▪ Data centers</td>
<td></td>
</tr>
<tr>
<td>Business workstations</td>
<td>Corporate DMZ</td>
<td>Low</td>
<td>Confidentiality</td>
<td>▪ Copy of corporate data</td>
<td>12.5</td>
</tr>
<tr>
<td>Web servers</td>
<td>Corporate DMZ</td>
<td>Low</td>
<td>Confidentiality</td>
<td>▪ Copy of corporate data</td>
<td></td>
</tr>
<tr>
<td>Authentication server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web servers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTP servers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Simulation
Simulation setup

- All nodes have the same operating system.
- Only vulnerabilities published in the last month are considered as unpatched (August 2017)
- Vulnerabilities depend on type of products and manufacturers
- For each position, the attacker chooses one edge vulnerability to exploit
- At each time step, the defender choses a vulnerable application to shut down. This automatically cut all edges corresponding to that application.
- To capture security policies, links between layers are generated with a Bernoulli trial probability law of parameter $p$ (Some users, some devices and some protocols may not be allowed to establish connections)
- Number of nodes at each layer:

<table>
<thead>
<tr>
<th>Layer</th>
<th>Corporate DMZ</th>
<th>Corporate</th>
<th>Control DMZ</th>
<th>Control system</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>6</td>
<td>64</td>
<td>4</td>
<td>26</td>
<td>100</td>
</tr>
<tr>
<td>Percentage</td>
<td>70%</td>
<td></td>
<td>30%</td>
<td></td>
<td>100%</td>
</tr>
</tbody>
</table>
Simulation

Rapidity

The convergence speed is affected by the discounted factor.
If the attacker uses a deterministic strategy, the optimal defense strategy is also deterministic.

<table>
<thead>
<tr>
<th>Attacker strategy</th>
<th>Optimal defense strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shortest path</td>
<td>Vulnerabilities corresponding to the shortest path</td>
</tr>
<tr>
<td>Least cost edges</td>
<td>Vulnerabilities corresponding to least cost edges</td>
</tr>
<tr>
<td>Movement toward next most attractive node</td>
<td>Vulnerabilities corresponding to most attractive node</td>
</tr>
</tbody>
</table>
Simulation
Robustness

Statistical distribution of the final location of the attacker with 100 Monte Carlo trials

**DEFENDER**

Markov

**ATTACKER**

Randomly uniformly

\[
\text{ATTRACTION} = 35
\]

\[
\text{LOCATION} = \text{CORPORATE DMZ}
\]

\[
\text{ATTRACTION} = 20
\]

\[
\text{LOCATION} = \text{CORPORATE DMZ}
\]
Simulation

Robustness

Statistical distribution of the time needed by the attacker to reach the control system layer with 100 Monte Carlo trials

21% of attacks ended in the control system

\[ \text{TIME} = 4.5 \]

11% of attacks ended in the control system

\[ \text{TIME} = 4.7 \]
OUTLINE

• INTRODUCTION

• POWER GRID NETWORK ARCHITECTURE

• VULNERABILITY MULTI-GRAH

• TWO-PLAYER ZERO SUM-MARKOV GAME

• SIMULATIONS

• CONCLUSION
Conclusion and Perspective

• Markov improves the system resilience:
  • by increasing the rapidity of the response (response delay of few seconds with no human in the loop)
  • by increasing the robustness the attack (critical asset are protected and the impact is minimized)
  • By increasing the resourcefulness (providing the optimal response actions at each point of the system)

• The game is built on known vulnerabilities that an attacker can exploit to move laterally from host to host until reaching an attractive target.

• Need to consider the physical layer