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

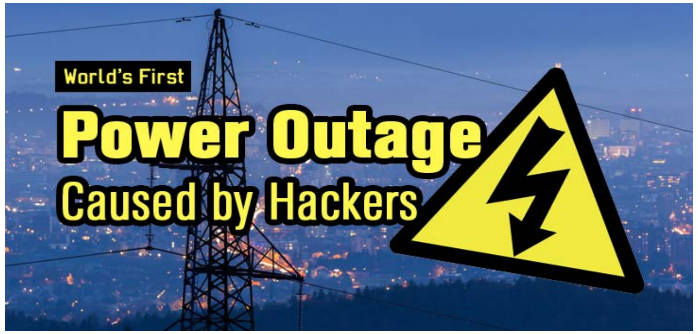
December 2015 Ukraine power grid attack

- 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

2nd 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^{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

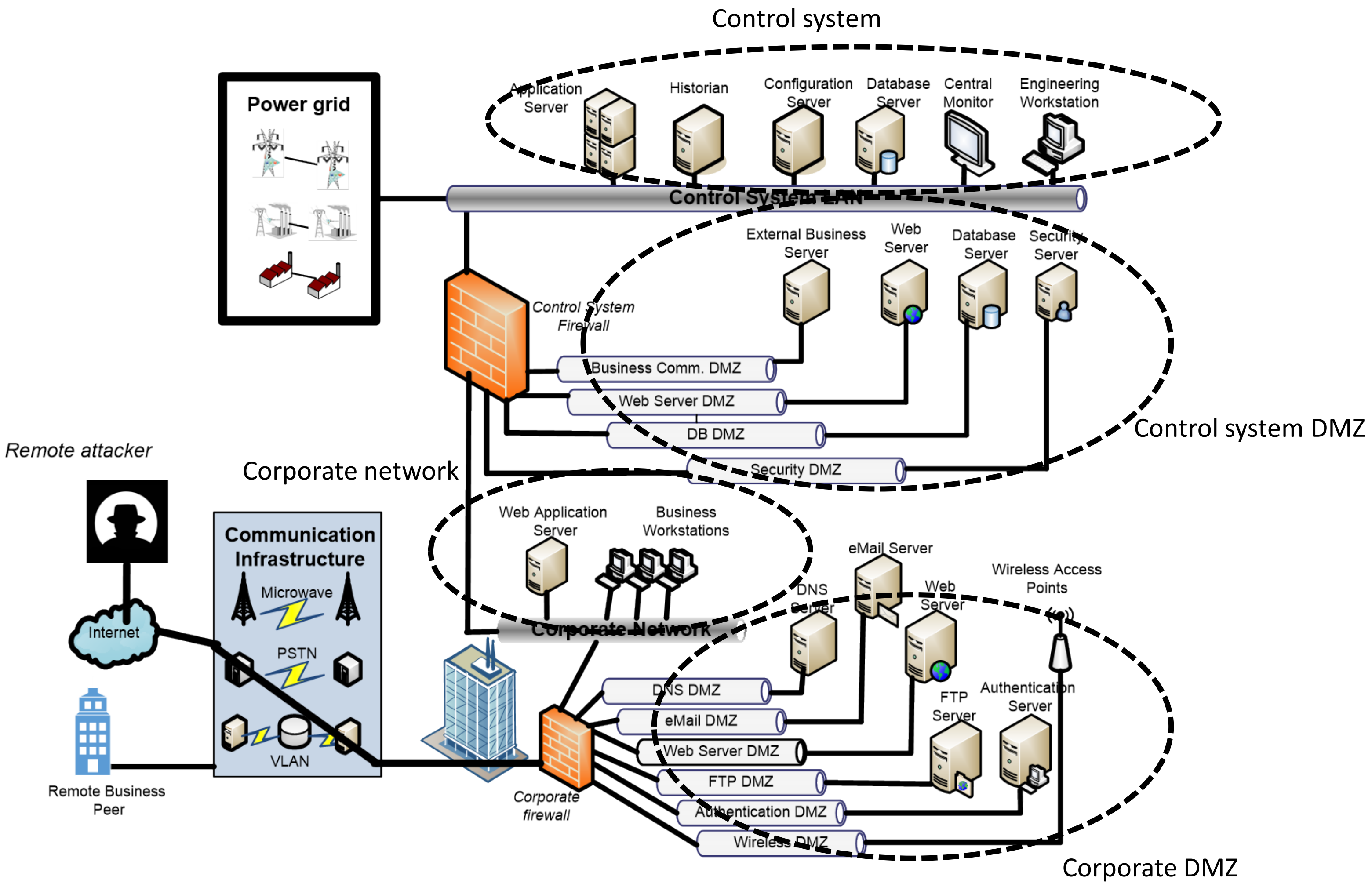
- Increase the **Robustness** by redirecting the malware into part of the system where critical assets are not accessible, and thus minimizing the impact of attacks

OUTLINE

- INTRODUCTION
- POWER GRID NETWORK ARCHITECTURE
- VULNERABILITY MULTI-GRAH
- TWO-PLAYER ZERO SUM-MARKOV GAME
- SIMULATIONS
- CONCLUSION

Power grid network architecture

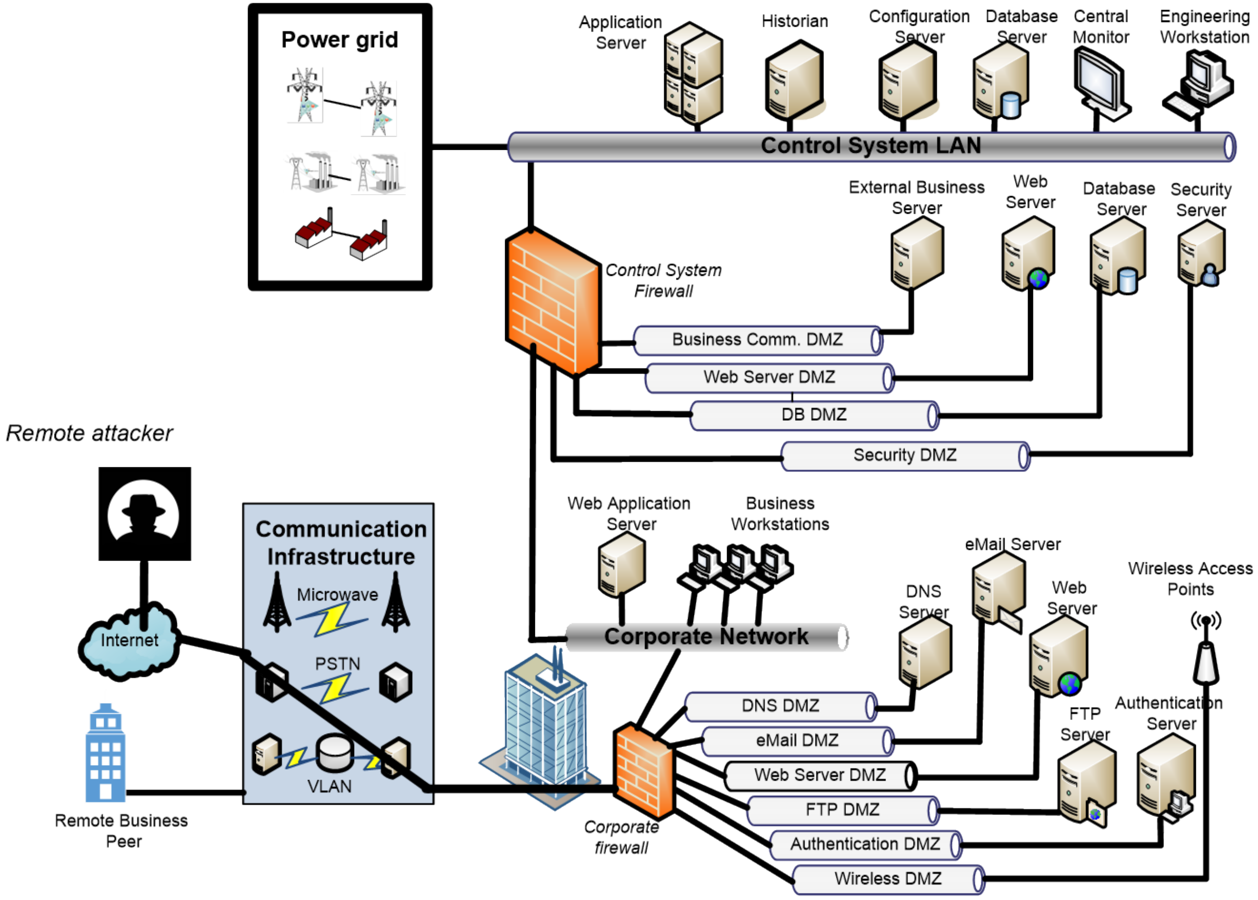
Recommended defense-in-depth architecture for Industrial Control System [1]



[1] Keith Stouffer et al. 'Guide to Industrial Control Systems (ICS) Security', *Sp800-82, NIST, May 2015*.

Power grid network architecture

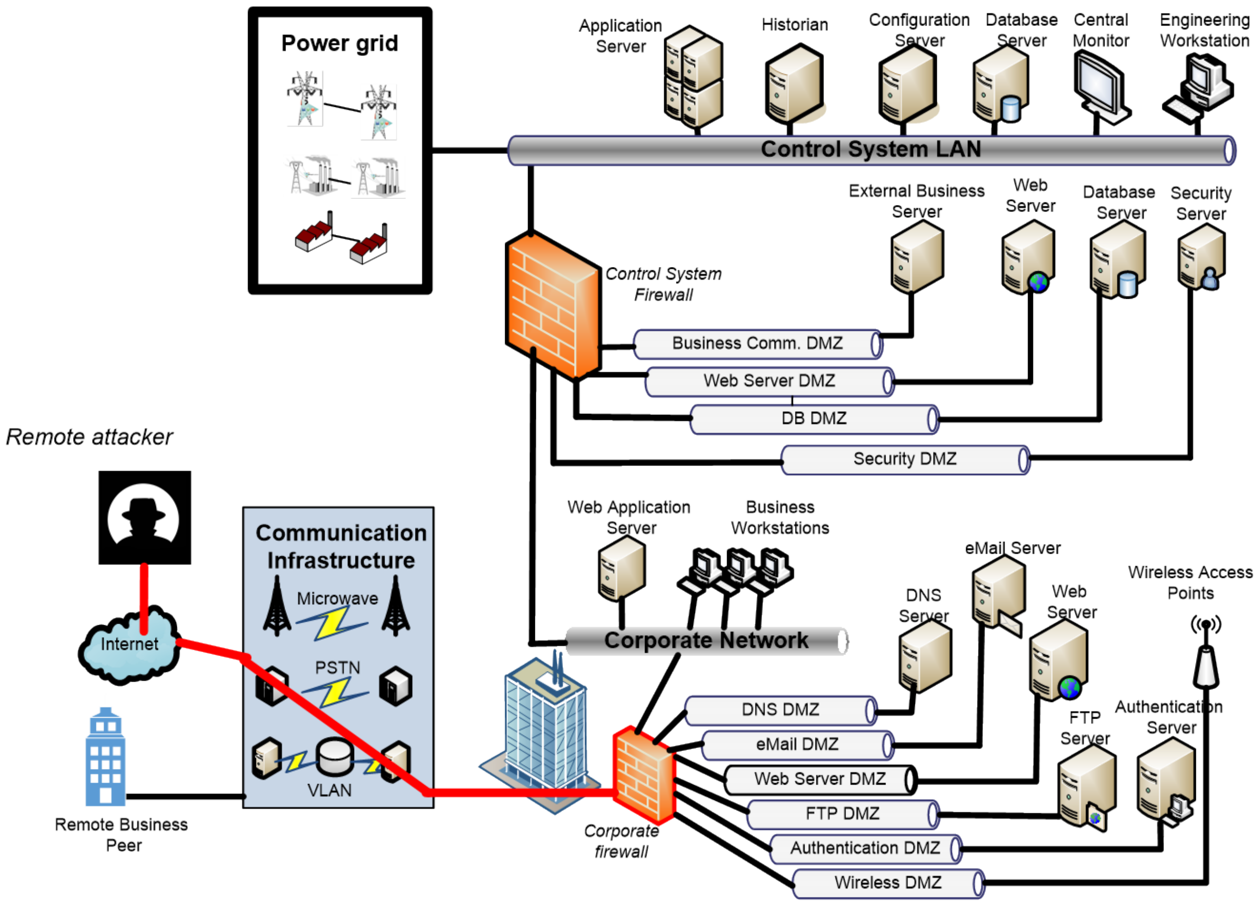
Execution of crafted code via web server



[2] **CVE-2015-1635, https://nvd.nist.gov**

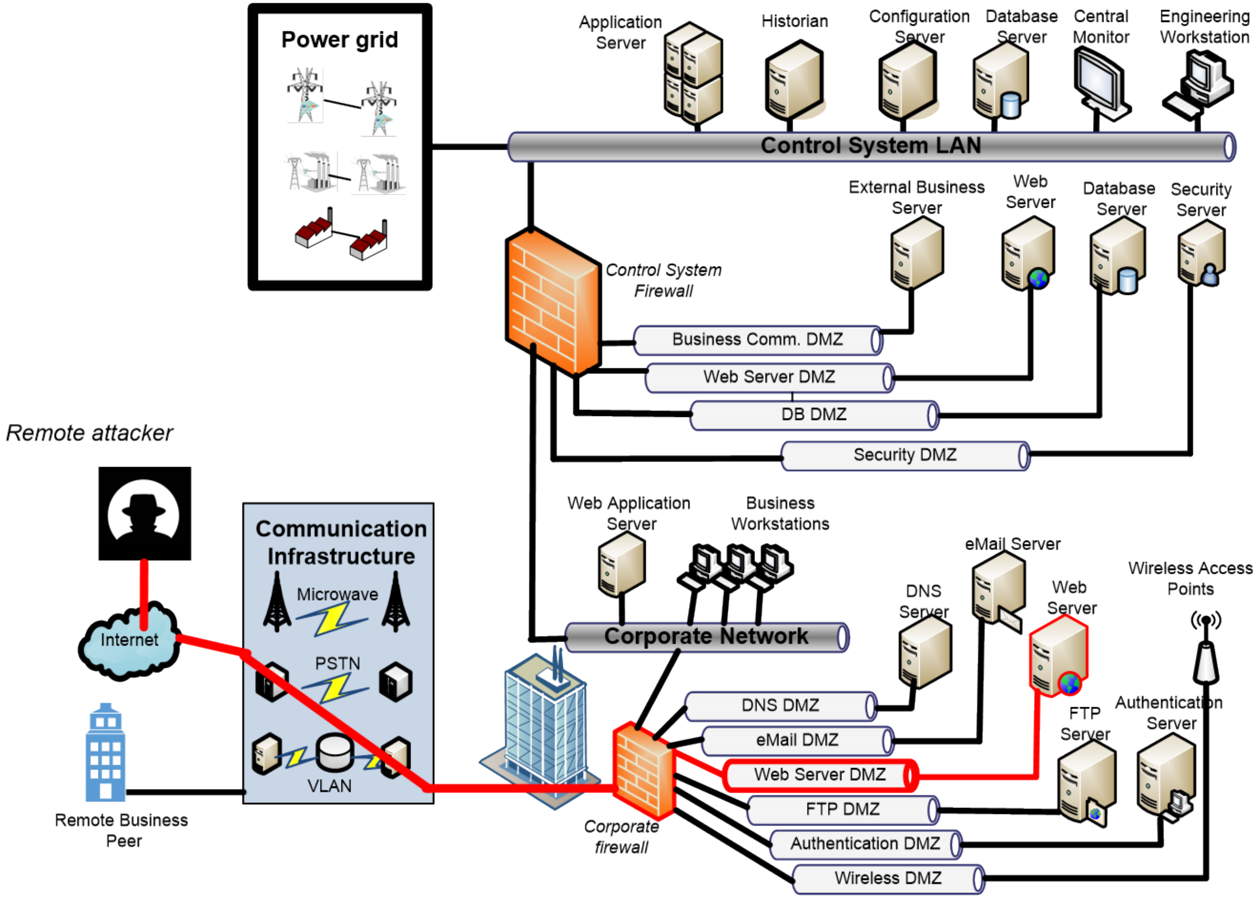
Power grid network architecture

Execution of crafted code via web server



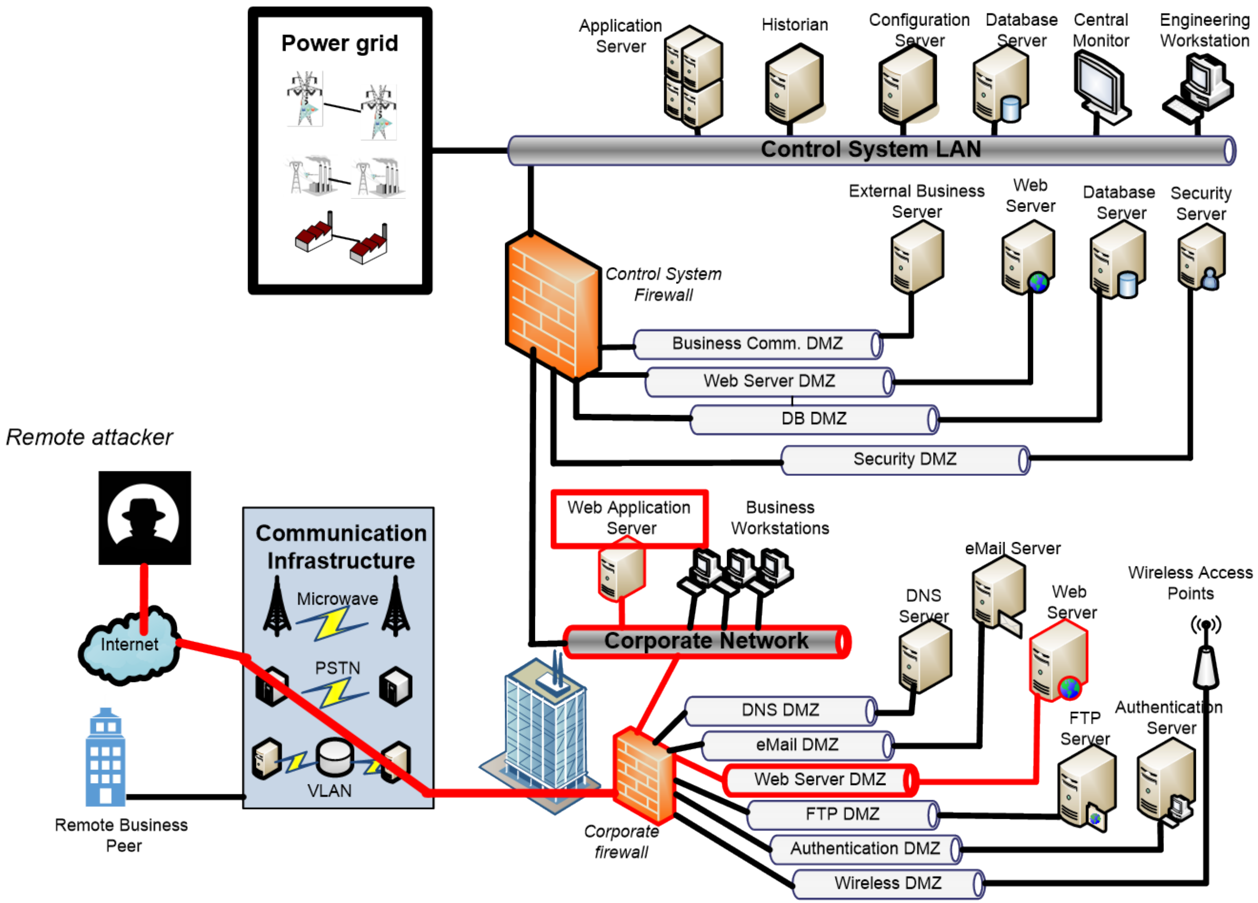
Power grid network architecture

Execution of crafted code via web server



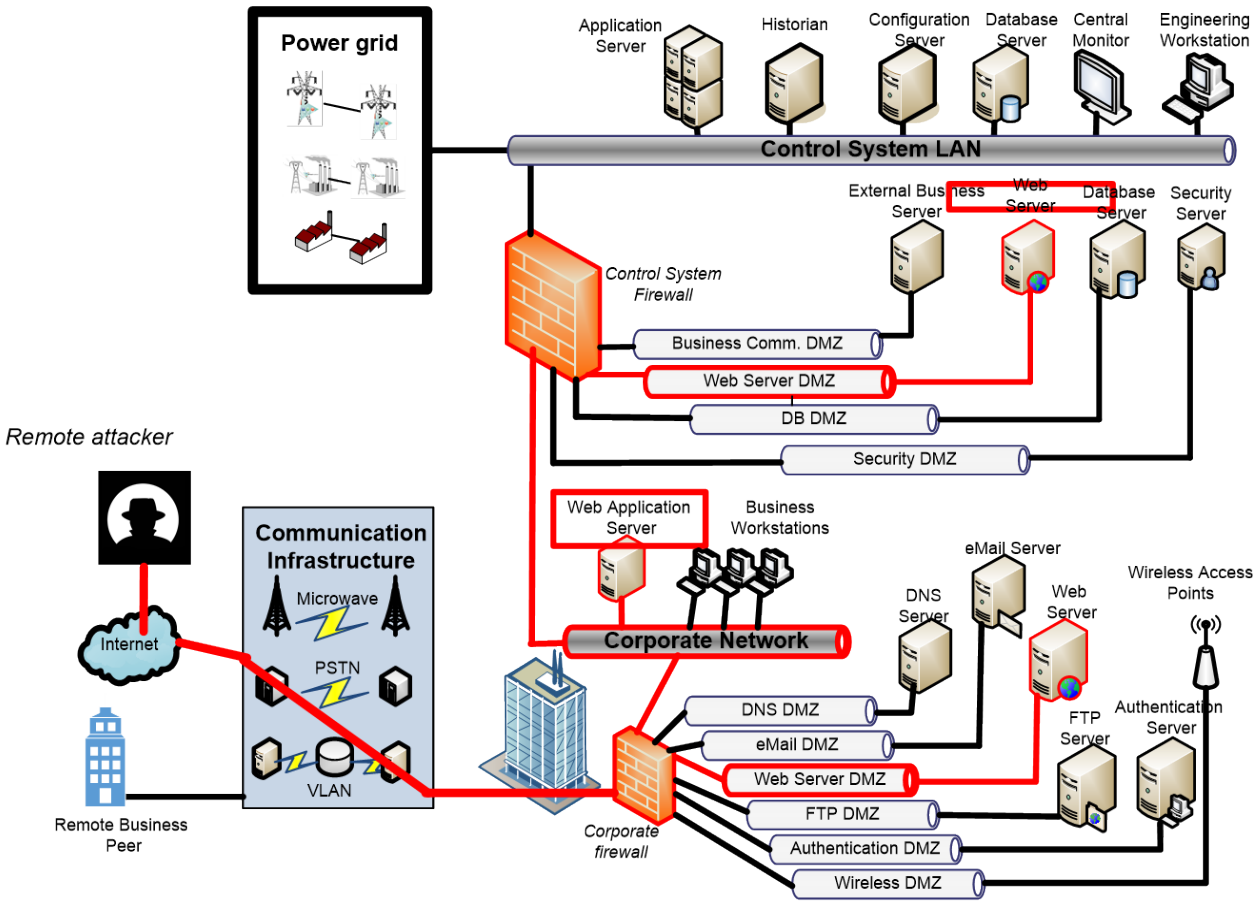
Power grid network architecture

Execution of crafted code via web server



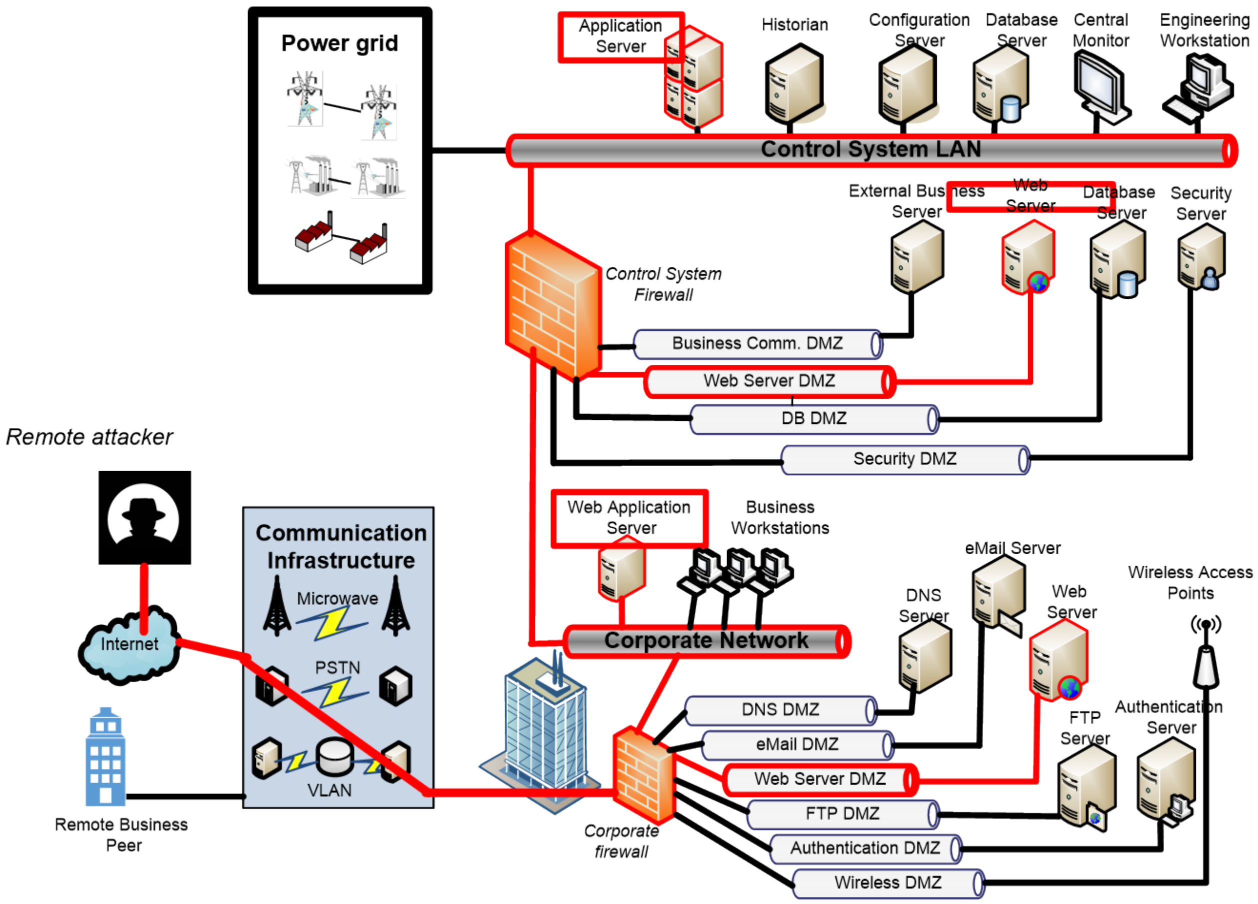
Power grid network architecture

Execution of crafted code via web server



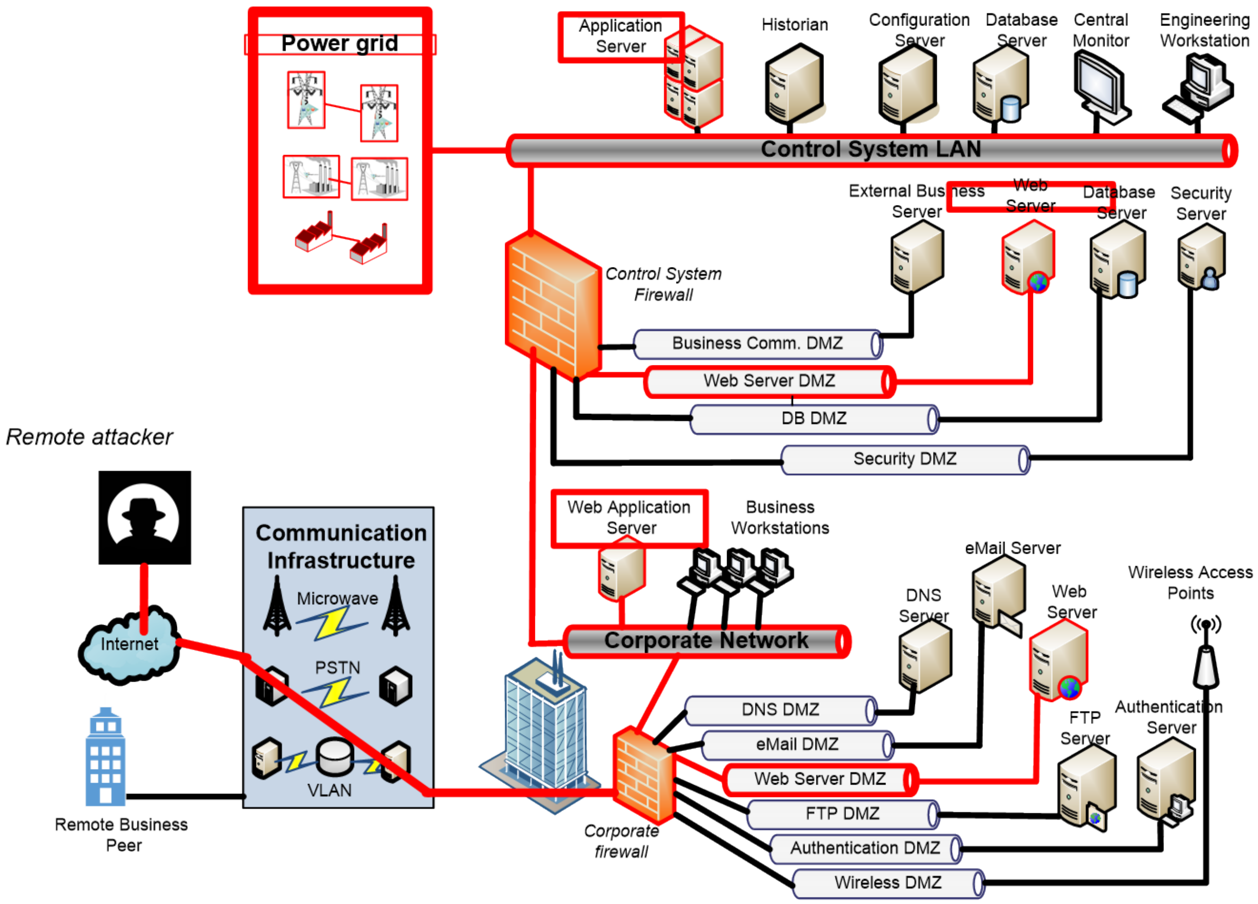
Power grid network architecture

Execution of crafted code via web server



Power grid network architecture

Execution of crafted code via web server



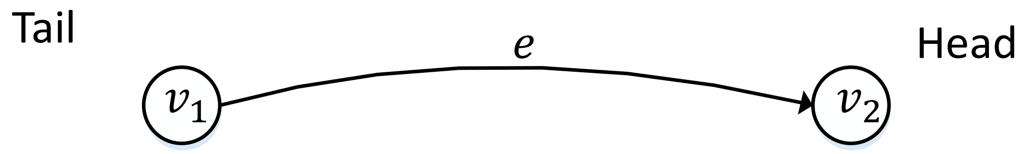
OUTLINE

- INTRODUCTION
- POWER GRID NETWORK ARCHITECTURE
- VULNERABILITY MULTI-GRAH MODEL
- TWO-PLAYER ZERO SUM-MARKOV GAME
- SIMULATIONS
- CONCLUSION

Vulnerability Multi-Graph

Edge vulnerability

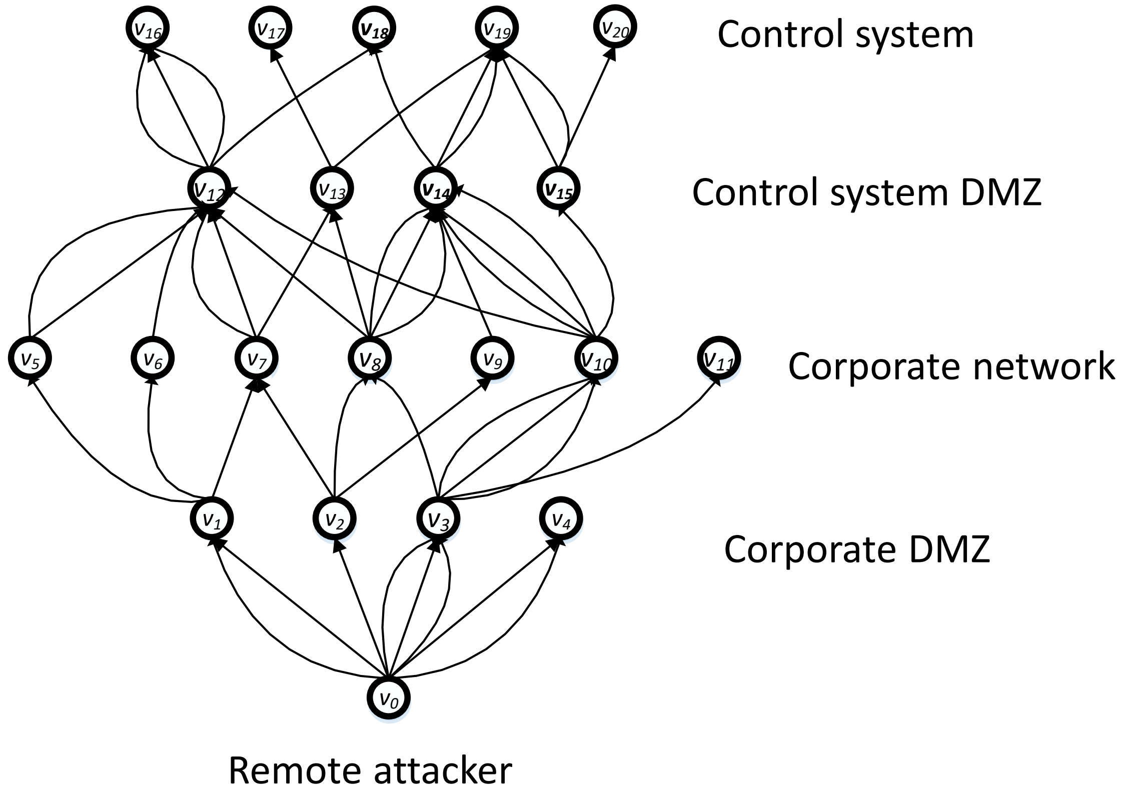
An edge vulnerability *$e \in E$ is a directed edge from a node v_1 to a node v_2 which corresponds to a vulnerability hosted by an application on v_2 that the system rules allow to access from node v_1.*



- Φ the set of vulnerabilities
- $\varphi(e) \in \Phi$, the vulnerability associated to e
- $v_2 = \Upsilon_{Head}(e)$ is the head of e
- $v_1 = \Upsilon_{Tail}(e)$ is the tail of e

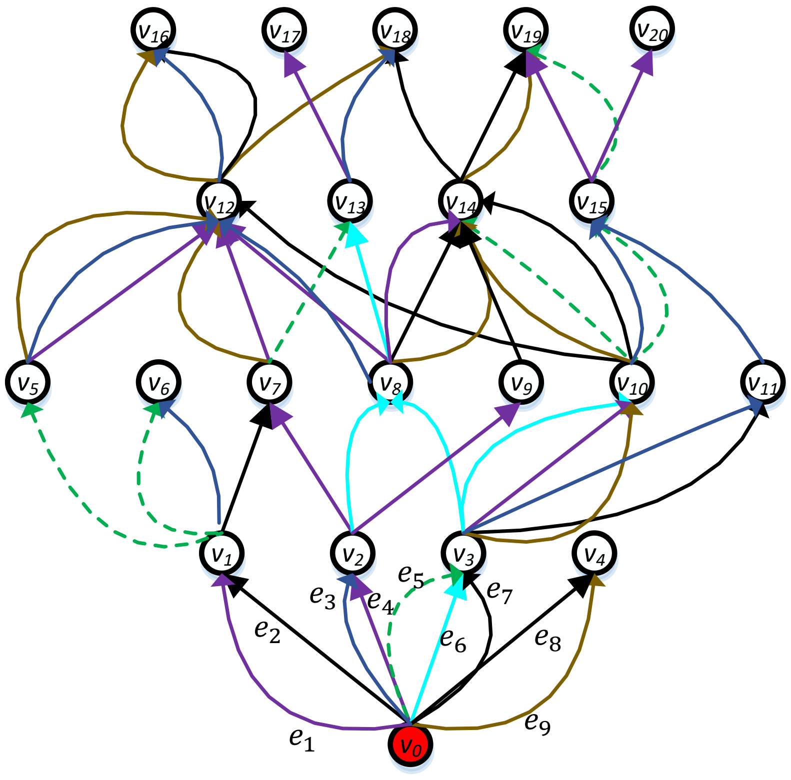
Vulnerability Multi-Graph

Direct acyclic graph



Vulnerability Multi-Graph

Lateral movement



Defender actions

Attacker actions

	Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6
e_1	v_0	v_1	v_1	v_1	v_1	v_1
e_2	v_1	v_1	v_1	v_0	v_1	v_1
e_3	v_2	v_2	v_2	v_2	v_2	v_0
e_4	v_0	v_2	v_2	v_2	v_2	v_2
e_5	v_3	v_3	v_0	v_3	v_3	v_3
e_6	v_3	v_0	v_3	v_3	v_3	v_3
e_7	v_3	v_3	v_3	v_0	v_3	v_3
e_8	v_4	v_4	v_4	v_0	v_4	v_4
e_9	v_4	v_4	v_4	v_4	v_0	v_4

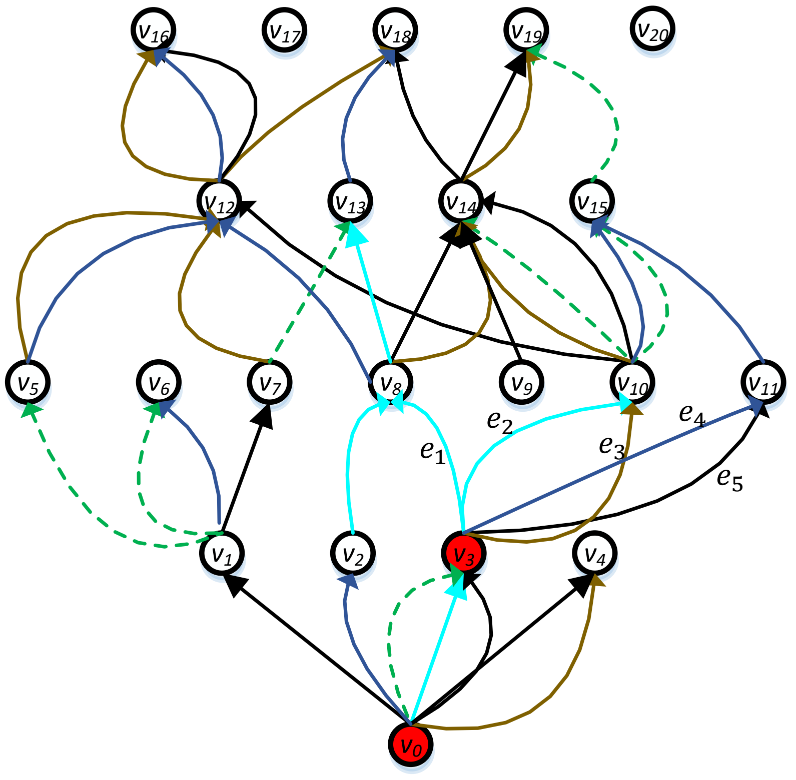
Matrix of actions at v_0

Active vulnerabilities					
Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6

- ❑ Attacker moves to v_3
- ❑ Vulnerable service associated to Φ_1 is disabled

Vulnerability Multi-Graph

Lateral movement



Defender actions

Attacker actions

	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6
e_1	v_3	v_8	v_8	v_8	v_8
e_2	v_3	v_{10}	v_{10}	v_{10}	v_{10}
e_3	v_{10}	v_{10}	v_{10}	v_3	v_{10}
e_4	v_{11}	v_{11}	v_{11}	v_{11}	v_3
e_5	v_{11}	v_{11}	v_3	v_{11}	v_3

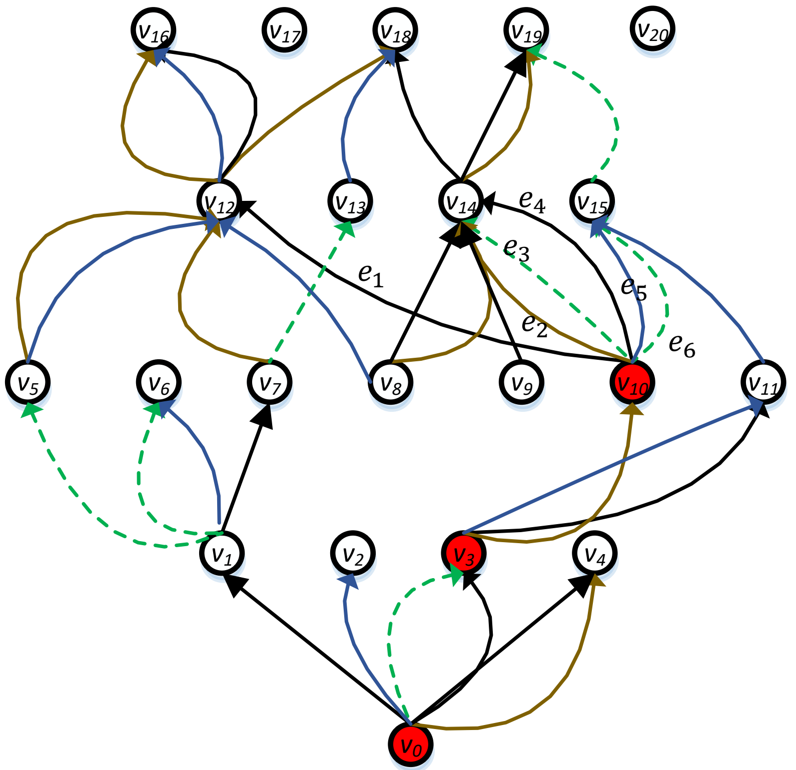
Matrix of actions at v_3

Active vulnerabilities					
~~Φ_1~~	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6

- Attacker moves to v_{10}
- Vulnerable service associated to Φ_2 is disabled

Vulnerability Multi-Graph

Lateral movement



Defender actions

Attacker actions

	Φ_3	Φ_4	Φ_5	Φ_6
e_1	v_{12}	v_{10}	v_{12}	v_{12}
e_2	v_{14}	v_{14}	v_{10}	v_{14}
e_3	v_{10}	v_{14}	v_{14}	v_{14}
e_4	v_{14}	v_{10}	v_{14}	v_{14}
e_5	v_{15}	v_{15}	v_{15}	v_{10}
e_6	v_{10}	v_{15}	v_{15}	v_{15}

Matrix of actions at v_{10}

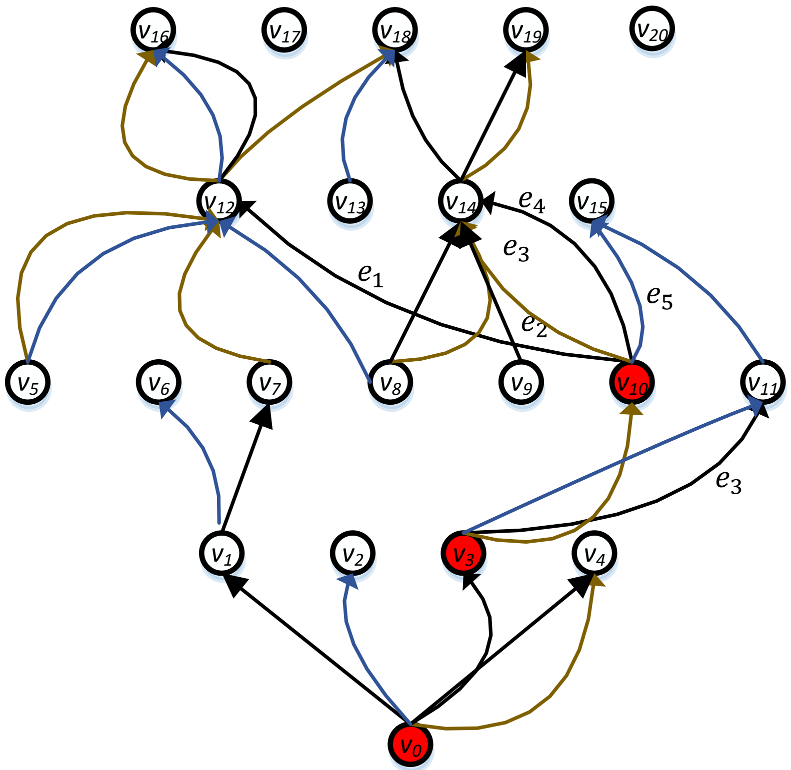
Active vulnerabilities

Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6

- ❑ Attacker remains at node v_{10}
- ❑ Vulnerable service associated to Φ_3 is disabled

Vulnerability Multi-Graph

Lateral movement



Defender actions

Attacker actions

	Φ_4	Φ_5	Φ_6
e_1	v_{10}	v_{12}	v_{12}
e_2	v_{14}	v_{10}	v_{14}
e_3	v_{14}	v_{14}	v_{14}
e_4	v_{10}	v_{14}	v_{14}
e_5	v_{15}	v_{15}	v_{10}

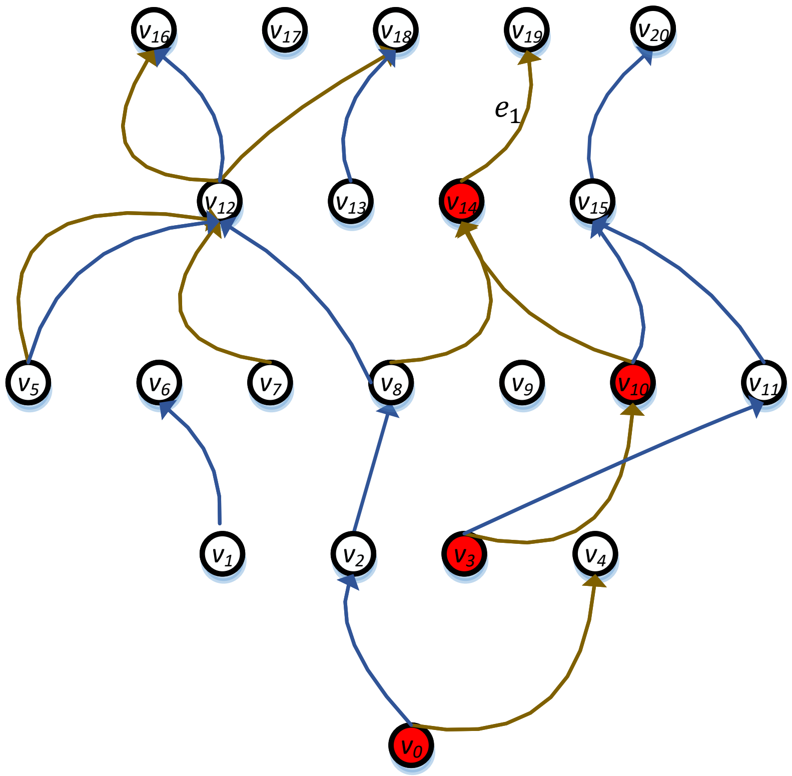
Matrix of actions at v_{10}

Active vulnerabilities					
Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6

- Attacker moves to node v_{14}
- Vulnerable service associated to Φ_4 is disabled

Vulnerability Multi-Graph

Lateral movement



Defender actions

Attacker actions

	Φ_5	Φ_6
e_1	v_{19}	v_{19}

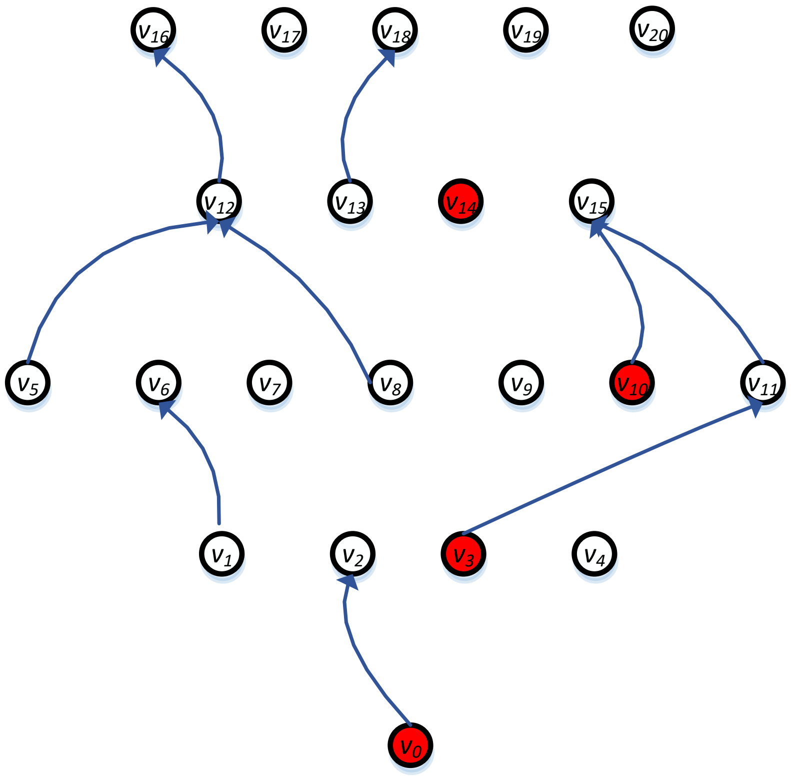
Matrix of actions at v_{14}

Active vulnerabilities					
Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6

- Attacker is isolated at node v_{14}
- Vulnerable service associated to Φ_4 is disabled

Vulnerability Multi-Graph

Lateral movement



Defender actions

Attacker actions

	Φ_5	Φ_6
e_1	v_{19}	v_{19}

- Attacker is isolated at node v_{14}
- Vulnerable service associated to Φ_4 is disabled

Six vulnerabilities

Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6

OUTLINE

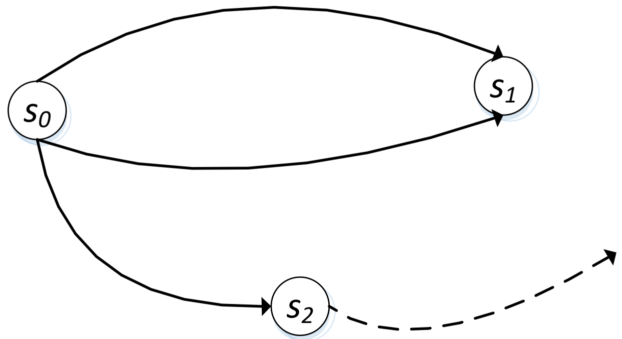
- INTRODUCTION
- POWER GRID NETWORK ARCHITECTURE
- VULNERABILITY MULTI-GRAH
- TWO-PLAYER ZERO SUM-MARKOV GAME
- SIMULATIONS
- CONCLUSION

Two-player zero-sum Markov Game

Definition

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

- $S = \{s_1 .. s_l\}$ is a finite set of game states;
- $A = \{a_1 .. a_n\}$ is the set of actions of the maximizer (row player);
- $O = \{o_1 .. o_m\}$ 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 $\mathcal{R}(s, a, o)$ 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;

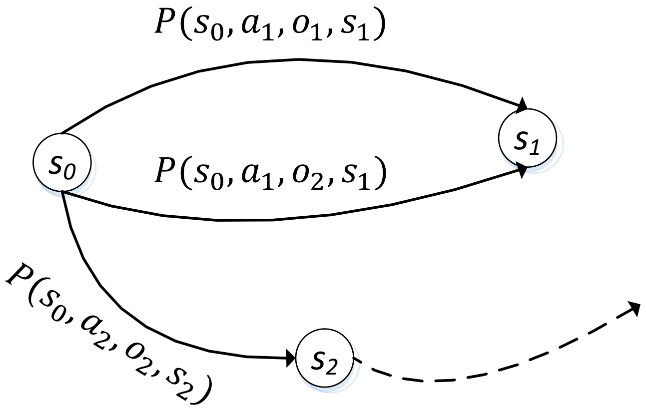


Two-player zero-sum Markov Game

Game matrix

Immediate reward matrix for state $s \in S$

		Column player			
		o_1	o_2	…	o_m
Row player	a_1	$\mathcal{R}(s, a_1, o_1)$			
	a_2				
	…				
	a_n				$\mathcal{R}(s, a_n, o_m)$



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 π_A, $\pi_A(s,a)$ denotes the probability to take action a in state s.

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

$$Q^\pi(s,a,o) = \underbrace{\mathcal{R}(s,a,o)}_{\text{Immediate reward}} + \underbrace{\gamma \sum_{s' \in S} P(s,a,o,s') \min_{o' \in O} \sum_{a' \in A} Q^\pi(s,a,o)\, \pi(s',a')}_{\text{Future rewards}}$$

- Optimal policy π and two Bellman functions:

$$\begin{cases} W(s) = \max_{\pi_A(s) \in \Omega(A)} \min_{o \in O} \sum_{a \in A} Q(s\ a,o)\, \pi'(s,a) \\ Q(s,a,o) = \sum_{s' \in S} P(s' |\, a,o,s)[\mathcal{R}(s,a,o,s') + \gamma W(s')] \end{cases}$$

Two-player zero-sum Markov Game

Value iteration algorithm

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

$W \leftarrow 0$
$l \leftarrow 0$
Repeat

$l++$

For each $s \in S$ **do**

$$W_{l+1}(s) = \max_{\pi_A(s) \in \Omega(A)} \min_{o \in O} \sum_{a \in A} \pi(s, a) \sum_{s' \in S} P(s' | a, o, s)[\mathcal{R}(s, a, o, s') + \gamma W_l(s')]$$

Until $\forall s \in S, |W_{l+1}(s) - W_l(s)| < \epsilon$

For each $s \in S$ **do**

$$\pi(s) \leftarrow \pi(s): \max_{\pi_A(s) \in \Omega(A)} \min_{o \in O} \sum_{a \in A} \pi(s, a) \sum_{s' \in S} P(s' | a, o, s)[\mathcal{R}(s, a, o, s') + \gamma W_l(s')]$$

Return π, W_{l+1}

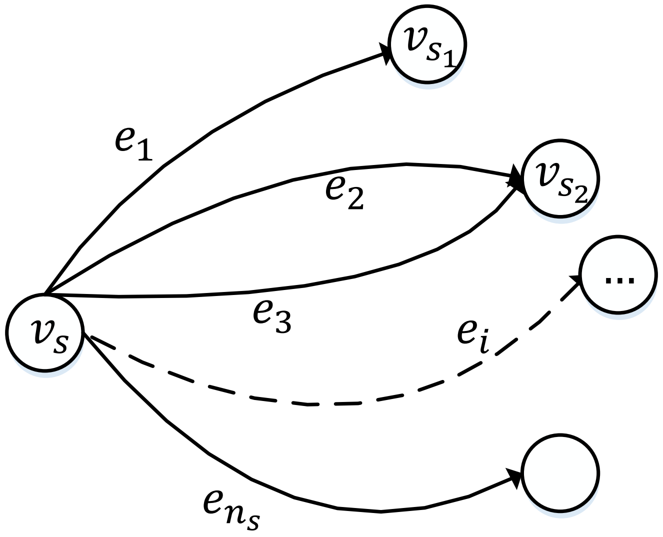
Two-player zero-sum Markov Game

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



		Defender actions			
		φ_1	φ_2	…	φ_{m_s}
Attacker actions	e_1				
	e_2				
	…				
	e_{n_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 $\mathcal{R}_A(s, e_i, \varphi_j) = \underbrace{\zeta(\varphi(e_i))}_{<0}$
  - The attacker stays at the same mode.

Effort required to exploit the vulnerability

- $\varphi_j \neq \varphi(e_i)$:
  - The attacker exploits successfully edge e_i and moves forward to next node
  - The immediate reward of the attacker is $\mathcal{R}_A(s, e_i, \varphi_j) = \underbrace{\zeta(\varphi(e_i))}_{<0} + \underbrace{A_t[\Upsilon_{Head}(e_i)]}_{\geq 0}$
  - The attacker moves to the next node.

Attractivity of next node

Effort required to exploit the vulnerability

OUTLINE

- INTRODUCTION
- POWER GRID NETWORK ARCHITECTURE
- VULNERABILITY MULTI-GRAH
- TWO-PLAYER ZERO SUM-MARKOV GAME
- SIMULATIONS
- CONCLUSION

Simulation

Exploit cost of vulnerabilities

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

Access Vector $A_v(\varphi)$	Describes how close the attacker must be to exploit the vulnerability φ	Local	0.395
		Adjacent network	0.64
		Remote network	1.0
Access Complexity $A_c(\varphi)$	Describes how easy or difficult it is to exploit the vulnerability φ	High	0.395
		Medium	0.61
		Low	0.71
Access Authentication $A_a(\varphi)$	Describes the number of time an attacker must authenticate to exploit the vulnerability φ	Multiple	0.45
		Single	0.56
		None	0.704
CVSS Score $20. A_v(\varphi). A_c(\varphi). A_a(\varphi)$		By construction, $1.4 \leq CVSS(\varphi) \leq 10$	

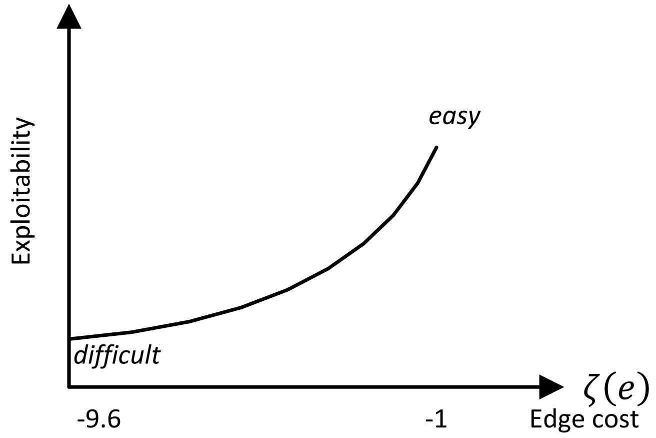
[3] Common Vulnerability Scoring System (CVSS-SIG), https://www.first.org/cvss

Simulation

Exploit cost of vulnerabilities

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

$$\zeta : E \rightarrow [-11, -1]$$
$$e \rightarrow \zeta(e) = CVSS(\varphi(e)) - 11$$



Simulation

Node Attractivity

The attractivity of a node $v \in G$ measures its appeal to cyber attack.

Typical nodes	Layer	Severity of cyber-attacks	Impact on power grid	Features	Attractivity $A_t(v)$
Application servers Database servers Engineering workstations	Control system	Critical	Availability Integrity Confidentiality	▪ Access to substation's controllers in real time ▪ Control algorithms and control commands ▪ Power transmission planning ▪ Power grid sensor's data	100
Historian database Web servers Authentication servers	Control system DMZ	High	Confidentiality Integrity	▪ Copy of control system data	50
Business servers Business workstations Web servers	Corporate network	Medium	Confidentiality	▪ Business data (billing, power consumption, etc…) ▪ Data centers	25
Authentication server Web servers FTP servers	Corporate DMZ	Low	Confidentiality	Copy of corporate data	12.5

Simulation

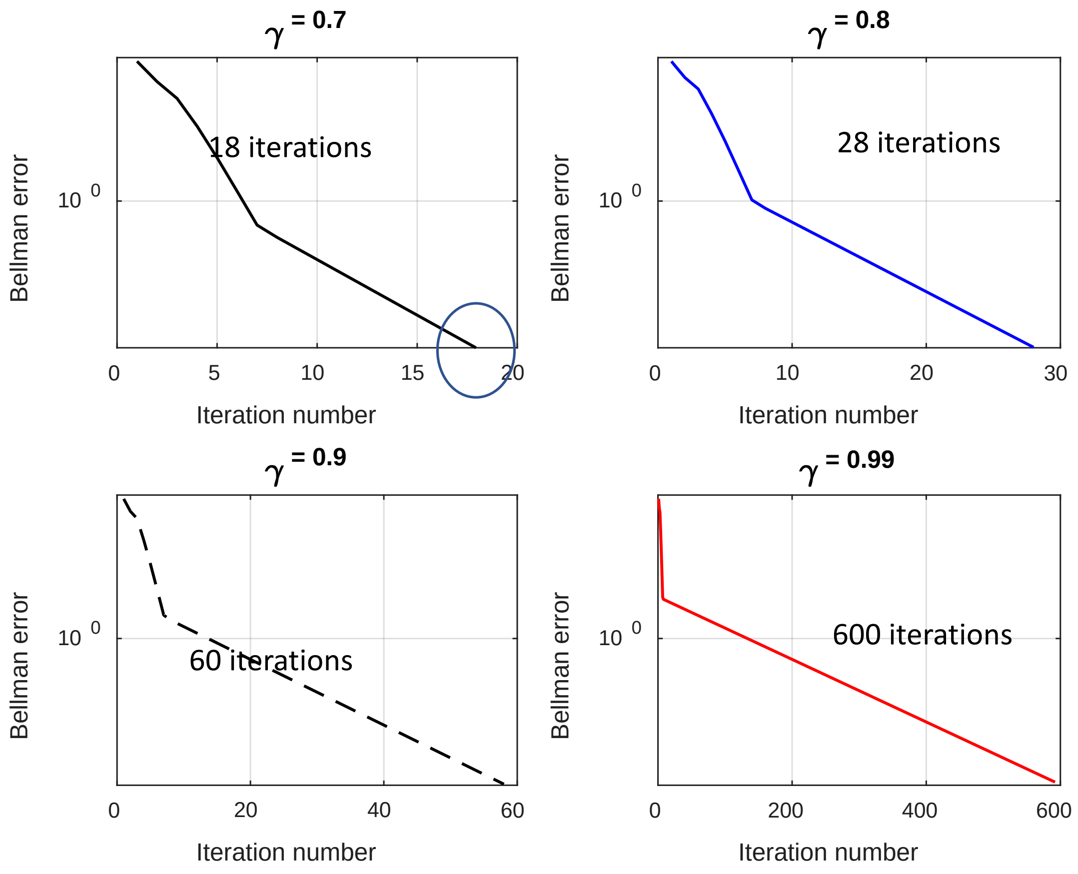
Simulation setup

- 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:

Layer	**Corporate DMZ**	**Corporate**	**Control DMZ**	**Control system**	**Total**
Number of nodes	6	64	4	26	100
Percentage	70%		30%		100%

Simulation

Rapidity



The convergence speed is affected by the discounted factor.

Simulation

Deterministic Strategies

If the attacker uses a deterministic strategy, the optimal defense strategy is also deterministic.

Attacker strategy	**Optimal defense strategy**
Shortest path	Vulnerabilities corresponding to the shortest path
Least cost edges	Vulnerabilities corresponding to least cost edges
Movement toward next most attractive node	Vulnerabilities corresponding to most attractive node

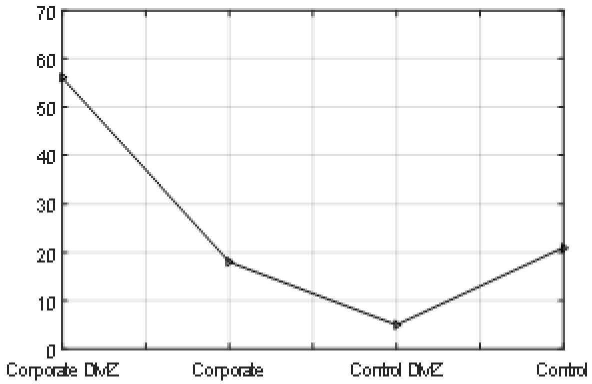
Simulation

Robustness

DEFENDER

Markov

Markov

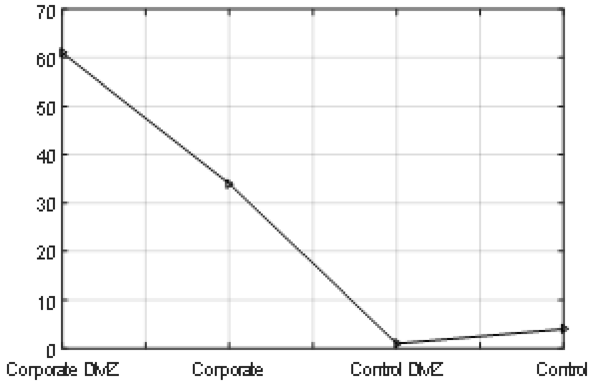


$\overline{ATTRACTIVITY} = 35$

$\overline{LOCATION} = CORPORATE\ DMZ$

ATTACKER

Randomly uniformly



$\overline{ATTRACTIVITY} = 20$

$\overline{LOCATION} = CORPORATE\ DMZ$

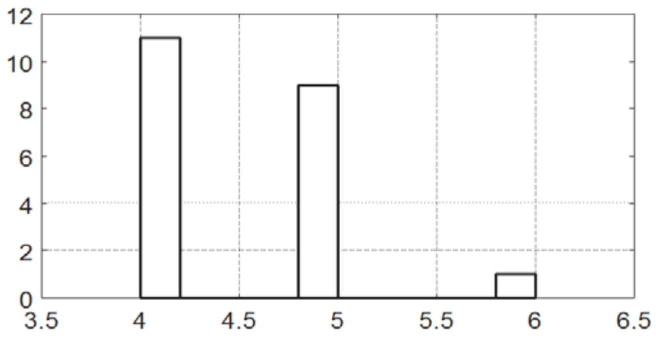
Statistical distribution of the final location of the attacker with 100 Monte Carlo trials

Simulation

Robustness

DEFENDER
Markov

Markov

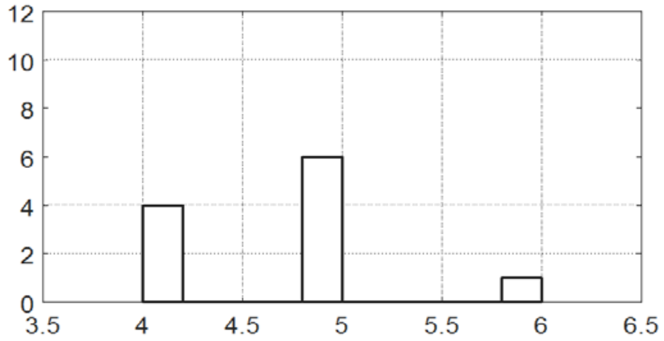


21 % of attacks ended in the control system

$$\overline{TIME} = 4.5$$

ATTACKER

Randomly uniformly



11 % of attacks ended in the control system

$$\overline{TIME} = 4.7$$

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

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