Detection and Mitigation of Cyber-Attacks Using Game Theory and Learning

João P. Hespanha
Kyriakos G. Vamvoudakis
Cyber Situation Awareness Framework

Observations: Netflow, Probing, Time analysis

Analysis to get up-to-date view of cyber-assets
Analysis to determine dependencies between assets and missions

Mission Model
Cyber-Assets Model

Sensor Alerts
Correlation Engine

Impact Analysis

Create semantically-rich view of cyber-mission status

Simulation/Live Security Exercises

Analyze and Characterize Attackers
Predict Future Actions

Data

COAs

Mission
Cyber-Assets
Outline...

- Large matrix games (summary of results)
- Multi-agent learning under cyber-attack using Q-learning (summary of results)
- Integration of online optimization for real-time attack prediction and visualization (summary of results)
- Observability of dynamical systems under attacks to sensors (summary and new results)
Large matrix games (summary of results)

Multi-agent learning under cyber-attack using Q-learning (summary of results)

Integration of online optimization for real-time attack prediction and visualization (summary of results)

Observability of dynamical systems under attacks to sensors (summary and new results)
Network Security Games

Problem statistics of iCTF 2010
- over 7800 distinct mission states (defender observations)
- over 2500 distinct observations available to the attacker
- defender can choose among about $10^{2527}$ distinct policies
- attacker can choose among $10^{756} - 10^{2616}$ distinct policies, depending on attacker's level of expertise

Even “trivially small” network security games can lead to games with very large decision trees.
Network Security Games

- Developed sample-based approach to solving zero-sum games
- Approach provides probabilistic guarantees on the performance of the policies (in terms of security levels)
- Results applicable to very general classes of games that can include stochastic actions, partial information, etc.

Problem statistics of iCTF 2010
- over 7800 distinct mission states (defender observations)
- over 2500 distinct observations available to the attacker
- defender can choose among about $10^{2527}$ distinct policies
- attacker can choose among $10^{756} - 10^{2616}$ distinct policies, depending on attacker's level of expertise

Even “trivially small” network security games can lead to games with very large decision trees
Application to iCTF 2010

We were able to
- Provide Cyber-security office estimates of mission success
- Provide dynamic rules to control firewall
- Take into account the effect of attacks & counter measures
- Response can be a function of attacker sophistication
- Play what-if scenarios (vulnerabilities, information, etc.)

<table>
<thead>
<tr>
<th>Level of attacker sophistication</th>
<th># units received by Litya for 1 round of missions [Option I, no bribes]</th>
<th># units received by Litya for 1 round of missions [Option I, with bribes]</th>
<th># units received by Litya for 1 round of missions [Option II, with bribes]</th>
</tr>
</thead>
<tbody>
<tr>
<td>no service vulnerable (baseline)</td>
<td>314</td>
<td>314</td>
<td>314</td>
</tr>
<tr>
<td>S2 (vulnerable to 38 teams)</td>
<td>240</td>
<td>240</td>
<td>138</td>
</tr>
<tr>
<td>S2, S6, S9 (vulnerable to at least 6 team)</td>
<td>79</td>
<td>79</td>
<td>43</td>
</tr>
<tr>
<td>S0, S2, S4, S6, S7, S8, S9</td>
<td>11</td>
<td>-738</td>
<td>-1327</td>
</tr>
<tr>
<td>all services vulnerable</td>
<td>11</td>
<td>-848</td>
<td>-1917</td>
</tr>
</tbody>
</table>
Outline...

- Large matrix games (summary of results)
- Multi-agent learning under cyber-attack using Q-learning (summary of results)
- Integration of online optimization for real-time attack prediction and visualization (summary of results)
- Observability of dynamical systems under attacks to sensors (summary and new results)
In complex cyber missions,
• *human operators* define policies and rules
• *computing elements* automate processes of distributed resource allocation, scheduling, inventory management, etc.

What is the impact of attacks on this type of automated/optimization process? Can we devise algorithms with built-in attack prediction/awareness capabilities?
Focus: Distributed Consensus/Agreement

Classical problem in distributed computing:
- A group of computing elements must agree on a common scalar value $x$ (e.g., priority, resources allocated, inventory decision, database value)
- Decision done iteratively & distributed using peer-to-peer communication

2\textsuperscript{nd} order adjustment rule

\begin{align*}
x_i(k + 1) &= x_i(k) + \Delta x_i(k) \\
\Delta x_i(k + 1) &= \Delta x_i(k) + u_i + v_i
\end{align*}

Goal: minimize errors between values of agents and their neighbors

\begin{align*}
e_i := \sum_{j \in N_i} \left[ \frac{x_i - x_j}{\Delta x_i - \Delta x_j} \right]
\end{align*}

Attacker: maximize errors using stealth attacks (small $v_i$)
Focus: Distributed Consensus/Agreement

Classical problem in distributed computing:
- A group of computing elements must agree on a common scalar value \( x \) (e.g., priority, resources allocated, inventory decision, database value)
- Decision done iteratively & distributed using peer-to-peer communication

2\textsuperscript{nd} order adjustment rule

\[ x_i(k + 1) = x_i(k) + \Delta x_i(k) \]
\[ \Delta x_i(k + 1) = \Delta x_i(k) + u_i + v_i \]

Nash equilibrium formulation:

\[ J_i = \sum_{k} \left( \| e_i \|^2 + \sum_{j \in \mathcal{N}} \left( \| u_j \|^2 - \gamma_{ij}^2 \| v_j \|^2 \right) \right) \]

- value at processor \( i \), iteration \( k \)
- correct update on adjustment
- update on adjustment by attacker
- adjustment on \( x_i \) by processor \( i \), at iteration \( k \)

error
- \( min. \) by us
- \( max. \) by attacker

our updates
- (small means smooth)
- \( min. \) by us
- \( max. \) by attacker

attacker updates
- (small means stealth)
- \( max. \) by us
- \( min. \) by attacker
Bellman Equation

\[ \frac{\partial V_i^T}{\partial e_i} \left( \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} e_i - \sum_{j \in \mathcal{N}_i} (u_i + v_i) \right) + \frac{1}{2} \left( \| e_i \|^2 + \sum_{j \in \mathcal{N}_i} (\| u_j \|^2 - \gamma_{ij}^2 \| v_j \|^2) \right) = 0 \]

Optimal Control and Attacker Policies

\[ u_i^* = -d_i \begin{bmatrix} 0 & I \end{bmatrix} \frac{\partial V_i}{\partial e_i} \]

\[ v_i^* = \frac{d_i}{\gamma_{ii}^2} \begin{bmatrix} 0 & I \end{bmatrix} \frac{\partial V_i}{\partial e_i} \]

Under appropriate regularity assumptions (smoothness)

\[ J_i(u_i^*, u_{-i}^*, v^*) \leq J_i(u_i, u_{-i}, v^*) \quad \forall u_i \quad u_i^* \text{ is optimal (minimal) for us} \]

\[ J_i(u^*, v_i^*, v_{-i}^*) \geq J_i(u^*, v_i, v_{-i}^*) \quad \forall v_i \quad v_i^* \text{ is optimal (maximal) for attacker} \]

Moreover,

- Consensus will be reached asymptotically
- All variables will remain bounded through the transient (in fact, Lyapunov stability)
Optimal Solution

Bellman Equation

\[
\frac{\partial V_i}{\partial e_i} \left( \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} e_i - \sum_{j \in N_i} (u_i + v_i) \right) + \frac{1}{2} \left( \|e_i\|^2 + \sum_{j \in N_i} (\|u_j\|^2 - \gamma_{ij}^2 \|v_j\|^2) \right) = 0
\]

But…

Bellman equation difficult to solve (curse of dimensionality)

Approach:

- Machine learning based algorithm to solve this distributed consensus problem

Key contributions

- Applies to second-order updates (and even more complex dynamics)
- Algorithms do not require global knowledge of the communication graph
- Algorithms do not require knowledge of the update rules used by other agents
- Formal guarantees of correctness (convergence)

- All variables will remain bounded through the transient (in fact, Lyapunov stability)
Critic = Model–free (distributed) algorithm to evaluate the current algorithm & estimate attacker actions
Actor = Model–free (distributed) algorithm to enact optimal decisions (based on critic’s findings)
Actor & Critic based on Approximate Dynamic Programming:
    Critic learns Q-function (action dependent) &
    Actor learns optimal control laws
Outline...

- Large matrix games (summary of results)
- Multi-agent learning under cyber-attack using Q-learning (summary of results)
- Integration of online optimization for real-time attack prediction and visualization (summary of results)
- Observability of dynamical systems under attacks to sensors (summary and new results)
**Challenges to real-time cyber-mission protection:**
- cyber assets shared among missions
- cyber asset requirements change over time
- missions can use different configurations of resources
- complex network of cyber-asset dependencies

**Cyber Missions Complexity**

- Attack on service S0 can result in multiple mission failure
- But, damage only realized if missions follow particular paths

**Cyber Awareness Questions:**
- When & where is an attacker most likely to strike?
- When & where is an attacker most damaging to mission completion?
- How will the answer depend on attacker resources? attacker skills? attacker knowledge?

(Real-time what-if analysis)
Developed an optimization formalism to predict (most likely/damaging) attacks…

**Damage equation:**
(For service $s$ at time $t$)

$$PD_s^t \approx \alpha_t^s + b_t^s \ AR_t^s$$

- Potential damage
- Equation parameters vary with time as mission progresses (learned from data in iCTF exercises)

**Uncertainty equation:**
(For service $s$ at time $t$)

$$p_t^s \approx \Pi_{[0,1]}(c_t^s - d_t^s \ AR_t^s)$$

- Probability of realizing damage
- Attack resources

**Optimal attacks:**

- Maximize

$$TD = \sum_t \sum_s PD_t^s(AR_t^s)p_t^s(AR_t^s)$$

- Total damage to mission

- Constrained by

$$\sum_s AR_t^s \leq TR_t, \ \forall t$$

- Total attack resources at time $t$

Formalism and predictions validated in ICTF 2011 exercise (89 teams, 1000+ participants)
Enabling a multi-resolution attack analysis...

1. High-level attack predictions based on online optimization

$$\arg \max_{\text{attacks}} TD = \sum_{t} \sum_{s} PD_t^s(AR_t^s)p_t^s(AR_t^s)$$

2. Potential damage & uncertainty associated with attacks to different services

$$PD_t^s \approx a_t^s + b_t^s \cdot AR_t^s$$
$$p_t^s \approx \Pi_{[0,1]}(c_t^s - d_t^s \cdot AR_t^s)$$

3. Parameters that determine damage and uncertainty

$$PD_t^s \approx a_t^s + b_t^s \cdot AR_t^s$$
$$p_t^s \approx \Pi_{[0,1]}(c_t^s - d_t^s \cdot AR_t^s)$$

<table>
<thead>
<tr>
<th>Service ID</th>
<th>Potential</th>
<th>Probability</th>
<th>Plausability</th>
<th>Res. Spend</th>
<th>Pts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>t+1</td>
<td>t+2</td>
<td>t+3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6.54</td>
<td>2.1</td>
<td>0.0</td>
<td>45.9%</td>
<td>53.6%</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
<td>0.26</td>
<td>0.0</td>
<td>60.0%</td>
<td>58.9%</td>
</tr>
<tr>
<td>3</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
<td>59.9%</td>
<td>60.0%</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
<td>59.9%</td>
<td>60.0%</td>
</tr>
<tr>
<td>5</td>
<td>6.54</td>
<td>0.83</td>
<td>6.54</td>
<td>45.9%</td>
<td>58.1%</td>
</tr>
<tr>
<td>6</td>
<td>4.0</td>
<td>0.50</td>
<td>4.0</td>
<td>45.9%</td>
<td>58.1%</td>
</tr>
<tr>
<td>7</td>
<td>0.0</td>
<td>4.0</td>
<td>0.0</td>
<td>60.0%</td>
<td>45.9%</td>
</tr>
<tr>
<td>8</td>
<td>4.0</td>
<td>4.0</td>
<td>4.0</td>
<td>45.9%</td>
<td>45.9%</td>
</tr>
<tr>
<td>9</td>
<td>1.3</td>
<td>6.54</td>
<td>4.0</td>
<td>56.4%</td>
<td>45.9%</td>
</tr>
<tr>
<td>10</td>
<td>1.3</td>
<td>0.0</td>
<td>4.0</td>
<td>56.4%</td>
<td>60.0%</td>
</tr>
</tbody>
</table>

- High-level predictions permit fast action
- Low-level parameters permits investigating rationale for predictions
Large matrix games (summary of results)

Multi-agent learning under cyber-attack using Q-learning (summary of results)

Integration of online optimization for real-time attack prediction and visualization (summary of results)

Observability of dynamical systems under attacks to sensors (summary and new results)
Detection in Adversarial Environments

How to interpret & access the reliability of “sensors” that have been manipulated?

“Sensors” relevant to cyber missions?
• Measurement sensors (e.g., SCADA systems)
• Computational sensors (e.g., weather forecasting simulation engines)
• Data retrieval sensors (e.g., database queries)
• Cyber-security sensors (e.g., IDSs)

Sensor-error Domains
• Deterministic sensors: with \( n \) sensors, one can get correct answer as long as \( m < n/2 \) sensors have been manipulated
• **Stochastic sensors** without manipulation: solution given by hypothesis testing/estimation
  \[
  P(\text{sensor error}) = p_{\text{err}} \quad \Rightarrow \quad P(\text{n sensor error}) \approx \binom{n}{n/2} p_{\text{err}}^{n/2}
  \]
• **Stochastic sensors** with potential manipulation: open problem?
Problem formulation

$X$ – binary random variable to be estimated

$$P(X = 0) = P(X = 1) = \frac{1}{2} \quad \text{for simplicity}$$

$$P(Y_i \neq X) = p_{\text{err}} \quad P(Y_i = X) = 1 - p_{\text{err}} \quad \forall i$$

per-sensor error probability

(not necessarily very small)

$Y_1, Y_2, \ldots, Y_n$ – “noisy” measurements of $X$ produced by $n$ sensors

$Z_1, Z_2, \ldots, Z_n$ – measurements actually reported by the $n$ sensors

$$Z_i = \begin{cases} Y_i & \text{sensor } i \text{ not attacked} \\ ? & \text{sensor } i \text{ attacked} \end{cases}$$

at most $m$ sensors attacked

$p_{\text{attack}}$ – probability that we are under attack (very hard to know!)

interpretation of sensor data should be mostly independent of $p_{\text{attack}}$
Result for “small” # of sensors \((n<2/p_{err})\)

\(X\) – binary random variable to be estimated

\(Y_1, Y_2, \ldots, Y_n\) – “noisy” measurements of \(X\) produced by \(n\) sensors

\(Z_1, Z_2, \ldots, Z_n\) – measurements actually reported by the \(n\) sensors

\[ Z_i = \begin{cases} Y_i & \text{sensor } i \text{ not attacked} \\ ? & \text{sensor } i \text{ attacked} \end{cases} \]

at most \(m\) sensors attacked

\(p_{\text{attack}}\) – probability that we are under attack (very hard to know!)

\[ \rho := \sum_{k=m}^{n-1} \binom{n-m}{k-m} p_{err}^{n-k} (1-p_{err})^{k-m} \]

\[ \gamma := \sum_{k=0}^{n-1} \binom{n-m}{k} p_{err}^{n-m-k} (1-p_{err})^{k} \]

\[ \beta := \frac{1-p_{\text{attack}}}{p_{\text{attack}}} \frac{1}{(1-p_{err})^n - p_{err}^n} \]

Theorem:

The optimal estimator is

\[ y_2 = \begin{cases} \Pi_{[0,1]} \left( \frac{\gamma - \rho}{(1-p_{err})^{n-m} + p_{err}^{n-m}} \right) & \beta \leq p_{err}^{n-m} \\ 0 & \beta > p_{err}^{n-m} \end{cases} \]

\(Y_2\) go with the majority of the (potentially manipulated) sensor readings

\(Y_2\) go with the majority, EXCEPT if there is consensus

The optimal estimator is largely independent of \(p_{\text{attack}}\) (hard to know)
Theorem:
The optimal estimator is 

\[ \mu \text{ majority w.p. 1} \]

\[ \mu \text{ no consensus w.p. } \]

go with the majority of the (potentially manipulated) sensor readings

EXCEPT if there is consensus

\[ J_0 - n \]

\[ \hat{\mu} \]

\[ \hat{\mu} \]

\[ \hat{\mu} \]

\[ \hat{\mu} \]

\[ \hat{\mu} \]

\[ \hat{\mu} \]

The optimal estimator is largely independent of \( p_{\text{attack}} \) (hard to know)
Continuous Linear Systems

dynamical evolution of systems's state

\[ \dot{x} = Ax + Bu \]

control signals

\[ y_i = C_i x + D_i u, \quad i \in \{1, \ldots, N\} \]

measurements produced by sensor

\[ z_i = \begin{cases} 
  y_i & \text{sensor } i \text{ not attacked} \\
  ? & \text{sensor } i \text{ attacked} 
\end{cases} \]

measurements reported by sensor

at most \( M \) sensors can be manipulated by the attackers

Under what conditions can one reconstruct the state from (potentially corrupted) sensor measurements?
Continuous Linear Systems

dynamical evolution of systems's state

\[ \dot{x} = Ax + Bu \]

\[ y_i = C_i x + D_i u, \quad i \in \{1, \ldots, N\} \]

\[ z_i = \begin{cases} 
  y_i & \text{sensor } i \text{ not attacked} \\
  ? & \text{sensor } i \text{ attacked}
\end{cases} \]

at most \( M \) sensors can be manipulated by the attackers

**Theorem:**

Exact state reconstruction is possible if and only if system is observable through every subset of \( N - 2M \) measurements

\[ \downarrow \]

state could be reconstructed through only \( N - 2M \) measurements in the absence of attacks

\[ \downarrow \]

potential attack at \( M \) sensors, effectively “disables” \( 2M \) sensors
Estimation algorithms

Gramian-based estimator:
- batch, finite-time estimation
- inversion of the observability matrix at each time step

Observer-based estimator:
- asymptotic estimation
- recursive low-computation algorithm
- provably robust with respect to noise on all sensors (including non attacked ones)

Algorithm outline:

1. Build an estimate removing by ignoring a set $S$ of $M$ sensors
2. Build additional estimates by removing, in addition, all combinations of $M$ additional sensors
3. If all attacked sensors were in set $S$, then the estimates in steps 1. and 2. will be consistent (modulo noise)

(all estimates can be constructed without combinatorial complexity, by using finite dimensionality)
Discrete Event Systems - Background

alphabet: $\Sigma = \Sigma_c \cup \Sigma_u$
language: $L \subset \Sigma^*$

supervisor: $f : L \rightarrow \Gamma$

language controller by supervisor:

$$w\sigma \in L_f \iff w \in L_f, w\sigma \in L, \sigma \in f(w)$$

**Theorem.** There exists a supervisor $f$ such that $L_f = K$ iff $K$ is controllable

$$K\Sigma_u \cap L \subset K$$

observation map: $P : \Sigma \rightarrow (\Sigma_o \cup \{\epsilon\})$
P-supervisor: $g : P(L) \rightarrow \Gamma$

language controller by $P$-supervisor:

$$w\sigma \in L_g \iff w \in L_g, w\sigma \in L, \sigma \in g\left(P^*(w)\right)$$

**Theorem.** There exists a $P$-supervisor $g$ such that $L_g = K$ iff $K$ is controllable and $P$-observable

$$P^*(w) = P^*(w') \Rightarrow \exists \sigma \in \Sigma : w\sigma \in K, w'\sigma \in L\setminus K \text{ or } w\sigma \in L\setminus K, w'\sigma \in K$$
Supervised DES under attacks

Attack model:
- $m$ out of $n$ attacks active (which?)
- each attack
  - symbol distortion, erasure, insertion
  - nondeterminism

$A_i : \Sigma_o^* \rightarrow 2^{\Sigma_o^*}$
Supervised DES under attacks

Attack model:
- $m$ out of $n$ attacks active (which?)
- each attack
- symbol distortion, erasure, insertion
- nondeterminism

$A_i : \Sigma_o^* \rightarrow 2^{\Sigma_o^*}$

maximal language $L_{g,A}^{\max}$ controlled by $P$-supervisor $g$ under the attack $A$:

$$w\sigma \in L_{g,A}^{\max} \iff w \in L_{g,A}^{\max}, \ w\sigma \in L, \ \exists y \in AP^*(w), \ \sigma \in g(y),$$

set of observed symbols

minimal language $L_{g,A}^{\max}$ controlled by $P$-supervisor $g$ under the attack $A$:

$$w\sigma \in L_{g,A}^{\max} \iff w \in L_{g,A}^{\max}, \ w\sigma \in L, \ \forall y \in AP^*(w), \ \sigma \in g(y),$$

languages with largest/smallest number of word that attacker could enforce
Supervised DES under attacks

Attack model:
- $m$ out of $n$ attacks active (which?)
- each attack
- symbol distortion, erasure, insertion
- nondeterminism

$$A_i : \Sigma_o^* \rightarrow 2\Sigma_o^*$$

**Theorem.** There exists a $P$-supervisor $g$ such that $L_{g,A}^{\text{min}} = L_{g,A}^{\text{max}} = K$ iff $K$ is controllable and $P$-observable for set of attacks $A$

$$\exists A, A' \in A \quad AP^*(w) \cap A'P^*(w') \neq \emptyset$$

$$\Rightarrow \quad \exists \sigma \in \Sigma : w\sigma \in K, w'\sigma \in L \setminus K \text{ or } w\sigma \in L \setminus K, w'\sigma \in K.$$
Output-symbol attacks

Attack model:
- Each output symbol produced by one sensor
- One sensor could be manipulated (which?)
- Sensor manipulation: symbol erasure, insertion
- nondeterminism

\[ A_i = \text{arbitrary insertions/deletions of } i\text{-th output symbol} \]

**Theorem.** There exists a \( P \)-supervisor \( g \) such that \( L_{g,A}^{\text{min}} = L_{g,A}^{\text{max}} = K \) iff 

- \( K \) is controllable and \( P^{-i,j}_{-i,j} \)-observable, \( \forall i, j \in \Sigma_o \)

remove from output
string the symbols \( i \& j \)

potential attack at 1 sensor, effectively “disables” 2 sensor
Large matrix games (summary of results)

Multi-agent learning under cyber-attack using Q-learning (summary of results)

Integration of online optimization for real-time attack prediction and visualization (summary of results)

Observability of dynamical systems under attacks to sensors (summary and new results)