Optimal Attacks for the iCTF game

Kyriakos G. Vamvoudakis, João P. Hespanha *

July 15, 2012

Abstract

Security competitions are strong motivations for students to find novel security solutions. In this technical report we propose a mechanism for optimally allocating resources through the ten services of the competition 2011 international Capture The Flag (iCTF) in order to maximize the total number of points at the end of the competition. Our proposed mechanism is interpreted as a convex optimization problem. Two different optimization approaches are being considered, the first one considers that the data of the competition is known a priori and the second one uses a moving horizon scheme to make predictions of when to attack by using the history of existing data. We simulate the behavior of our proposed optimization schemes and compare them with what the teams actually did during the actual competition.

1 Background and Overview

The iCTF [2] is a distributed wide-area security exercise whose goal is to test the security skills of the participants. The iCTF contest is organized by Prof. Giovanni Vigna of the Department of Computer Science at UCSB and is held once a year. The Capture the Flag contest is a multi-site, multi-team hacking contest in which a number of teams compete independently against each other.

In traditional editions of the iCTF (2003-2007), the goal of each team was to maintain a set of services such that they remain available and uncompromised throughout the contest phase. Each team also has to attempt to compromise the other teams' services. Since all the teams received an identical copy of the virtual host containing the vulnerable services, each team has to find the vulnerabilities in their copy of the hosts and possible fix the vulnerabilities without disrupting the services. At the same time, the teams have to leverage their knowledge about the vulnerabilities they found to compromise the servers run by other teams. Compromising a service allows a team to bypass the service's security mechanisms and to "capture the flag" associated with the service. During the 2008-2010 iCTFs, new competition designs have been introduced. More precisely, in 2008 they created a separate virtual network for each team. The goal was to attack a terrorist network and defuse a bomb after compromising a number of hosts. In 2009, the participants had to compromise the browsers of a large group of simulated users, steal their money, and create a botnet. In 2010, the participants had to attack the rogue nation Litya, ruled by the evil Lisvoy Bironulesk. A new design forced the team to attack the services supporting Litya's infrastructure only at specific times, when certain activities were in progress. In addition, an intrusion detection system would temporarily firewall out the teams whose attacks were detected. In the last iCTF the teams were responsible for defending themselves against a digital onslaught, more description of this competition is provided below.

Our goal is to describe the 2011 iCTF competition from the perspective of one team playing against the rest of the world.

*K. G. Vamvoudakis, and J. P. Hespanha are with the Center for Control, Dynamical-systems and Computation (CCDC), University of California, Santa Barbara, CA 93106-9560 USA e-mail: kyriakos@ece.ucsb.edu, hespanha@ece.ucsb.edu. This material is based upon work supported in part by ARO MURI Grant number W911NF0910553.

1.1 Basic setup

The game is played in rounds (each takes about 2min). At each round teams take offensive and defensive actions. These actions lead to the team collecting "points'.' At the end of a fixed amount of time, the team that collected more points wins.

One challenge with the formulation "one-against-world" is that in the 2011 iCTF game, winning was not just about maximizing points. Winning was about getting more points than each of the opponents (individually). In the "one-against-world" the opponents lose their individuality and so it is not easy to keep track if one opponent is collecting points faster than we are. One may need to resort to some "mean-field games" setup.

Each team hosts a server that runs 10 services each with its own (unknown) vulnerabilities. Each service i, $i \in \{1, 2, \ldots, 10\}$ of each hosting team j (this index is not really used in our formulation since we model what the "optimal" team will do) is characterized by three time-varying quantities for $\forall k \in \{1, 2, \ldots, 248\}$ (the actual data was $\forall k \in \{3, 4, \ldots, 44, 50, \ldots, 255\}$, and has 248 ticks because the server was down from $45 - 49$, we mapped everything to write $k \in \{1, 2, \ldots, 248\}$):

- the *cut* C_k^i, which is the percentage of money that goes to team j when money is laundered through service i (same values for every team),

- the *payoff* P_k^i, which is the percentage of money that will be transformed into points for the team that launders the money (same value for every team);

$$P_k^i = 0.9e^{-\frac{TicksActive}{10}}$$

- the *risk* R_k^i, which is the probability of losing all the money

$$R_k^i = 0.9e^{-\frac{No.OfActiveStates}{2}}$$

At the beginning of each round k, the team is informed of the values of C_k^i, P_k^i, R_k^i for every i, and k.

1.2 Actions available to every team

A team (we) has the following actions in the actual competition:

1. *Defensive actions*: Activate/deactivate own services.

   In the iCTF game a team could also fix the vulnerability of a service. We assume here that we have fixed all vulnerabilities that we knew how to fix.

   We also assume that there is no tipping since that is an action that is specific against a particular opponent.

2. *Money laundering*: Select
   (a) team to attack j_k (mute decision within the "one-against-world" formulation);
   (b) service i to compromise, which implicitly determines the payoff P_k^i, the risk R_k^i, and the cut C_k^i (these quantities are specified at each round and take values $0 - 100\%$);
   (c) amount on money to launder u_k.

   This action results in a number of points given by

$$X_k = \begin{cases} P_k^i(1 - C_k^i)D_k u_k & \text{w.p. } 1 - \min\{O_k, 1\} \\ 0 & \text{w.p. } \min\{O_k, 1\} \end{cases}$$

where D_k is the team's defense level and

$$O_k \coloneqq \frac{R_k^i u_k}{30} + \frac{1}{6}\Big(\frac{N_k^j - 700}{300 + |N_k^j - 700|} + 1\Big) + \frac{1}{6}\Big(\frac{Q_k^i - 1500}{300 + |Q_k^i - 1500|} + 1\Big)$$

where N_k^j is the overall amount of money that has been laundered by the team through the particular team being exploited and Q_k^i is the overall amount of money that has been laundered by the team through the particular service being exploited. Because we do not model each team individually we will consider the "worst" case scenario for those two quantities, $N = 492$ and $Q = 2257$.

3. *Snitching services.* If team A submits a flag for a service for team B as "snitching", the conversion of team B will be penalized, because the service will appear as compromised.

In the following optimization schemes we limit the actions of the "optimal" team to when and how much money to launder and through which services. Also we assume that the team starts with \$100 and at every tick the team makes from challenges \$20. This scenario is equivalent with a team starting with a budget of \$5060, since the teams that are being simulated are not being attacked by any team. Moreover since we are considering one team against the world, the index j is removed.

2 Optimization using all the data a-priori

We are seeking to optimally allocate our available resources in the competition such that the total number of points is maximized while meeting the specified constraints. The optimization problem can be formulated as follows

$$\begin{aligned}
\text{maximize} \quad & \sum_{k=1}^{248}\sum_{i=1}^{10} \mathbb{E}\big[\rho_k^i P_k^i (1 - C_k^i) D_k^i u_k^i\big] \\
\text{subject to} \quad & \sum_{k=1}^{248}\sum_{i=1}^{10} u_k^i - \sum_{k=1}^{248} d_k \leqslant U \\
\text{w.r.t.} \quad & u_k^i \in [0, \infty),\ \forall i \in \{1, 2, \ldots, 10\},\ k \in \{1, 2, \ldots, 248\},
\end{aligned}$$

where d_k is the cut the team receives because of the owned services (in the optimization we are assuming that the team either started with a budget of \$5060 and did not make any cuts $d_k = 0, \forall k$, or the team started with \$100 and made $d_k = 20, \forall k$), U is the total amount of money available since the beginning of the competition and also for each $i \in \{1, 2, \ldots, 10\}$, the $\rho_1^i, \rho_2^i, \ldots, \rho_{248}^i$ are i.i.d. Bernoulli random variables with

$$\mathrm{P}(\rho_k^i = 0) = \min\{\frac{R_k^i}{30} u_k + \beta_k^i, 1\}$$

and the parameters $P_k^i, C_k^i, D_k^i, \in [0, 1]$, and $\beta_k^i \equiv \frac{1}{6}\left(\frac{N_k(j_k) - 700}{300 + |N_k(j_k) - 700|} + 1\right) + \frac{1}{6}\left(\frac{Q_k(i_k) - 1500}{300 + |Q_k(i_k) - 1500|} + 1\right) = 0.4$ $\forall i \in \{1, 2, \ldots, 10\}$, $k \in \{1, 2, \ldots, 248\}$ are all given.

Because of the i.i.d. assumption, and since it never makes sense to choose a value for u_k^i for which $\rho_k^i = 0$ with probability one, we can expand the expected value in the optimization criterion and obtain

$$\begin{aligned}
\text{maximize} \quad & \sum_{k=1}^{248}\sum_{i=1}^{10} P_k^i (1 - C_k^i) D_k^i (0.6 - \frac{R_k^i}{30} u_k) u_k^i \\
\text{subject to} \quad & \sum_{k=1}^{248}\sum_{i=1}^{10} u_k^i - \sum_{k=1}^{248} d_k \leqslant U
\end{aligned}$$

$$\text{w.r.t.} \qquad u_k^i \in \Big[0, \frac{0.6}{\frac{R_k^i}{30}}\Big], \ \forall i \in \{1,2,\dots,10\}, \ k \in \{1,2,\dots,248\},$$

which is a concave maximization problem (convex minimization) with linear constrains, which is easy to solve numerically and for which the duality gap is zero. The dual problem is given by

$$\begin{aligned}
J^\perp \coloneqq & \max_{\lambda\geqslant 0,\eta_k^i\geqslant 0,\zeta_k^i\geqslant 0} \max_{u_k^i\in\mathbb{R}} \sum_{k=1}^{248}\sum_{i=1}^{10} P_k^i(1-C_k^i)D_k^i(0.6-\frac{R_k^i}{30}u_k^i)u_k^i \\
& - \lambda\Big(\sum_{k=1}^{248}\sum_{i=1}^{10} u_k^i - \sum_{k=1}^{248} d_k - U\Big) - \sum_{k=1}^{248}\sum_{i=1}^{10}\eta_k^i\Big(u_k^i - \frac{0.6}{\frac{R_k^i}{30}}\Big) + \sum_{k=1}^{248}\sum_{i=1}^{10}\zeta_k^i u_k^i \\
= & \max_{\lambda\geqslant 0,\eta_k^i\geqslant 0,\zeta_k^i\geqslant 0} \max_{u_k^i\in\mathbb{R}} \sum_{k=1}^{248}\sum_{i=1}^{10}\Big(P_k^i(1-C_k^i)D_k^i(0.6-\frac{R_k^i}{30}u_k^i)u_k^i - \lambda u_k^i - \eta_k^i\Big(u_k^i - \frac{0.6}{\frac{R_k^i}{30}}\Big) + \zeta_k^i u_k^i\Big) \\
& + \lambda U + \lambda\sum_{k=1}^{248} d_k \\
= & \max_{\lambda\geqslant 0,\eta_k^i\geqslant 0,\zeta_k^i\geqslant 0} \max_{u_k^i\in\mathbb{R}} \sum_{k=1}^{248}\sum_{i=1}^{10}\Big(-P_k^i(1-C_k^i)D_k^i\frac{R_k^i}{30}{u_k^i}^2 + (P_k^i(1-C_k^i)D_k^i 0.6 + \zeta_k^i - \eta_k^i - \lambda)u_k^i + \eta_k^i\frac{0.6}{\frac{R_k^i}{30}}\Big) \\
& + \lambda U + \lambda\sum_{k=1}^{248} d_k.
\end{aligned}$$

The inner maximization can be solved using standard calculus and is achieved for

$$u_k^i = \frac{P_k^i(1-C_k^i)D_k^i 0.6 + \zeta_k^i - \eta_k^i - \lambda}{2P_k^i(1-C_k^i)D_k^i\frac{R_k^i}{30}},$$

yielding

$$J^\perp \coloneqq \max_{\lambda\geqslant 0,\eta_k^i\geqslant 0,\zeta_k^i\geqslant 0} \sum_{k=1}^{248}\sum_{i=1}^{10}\Big(\frac{(P_k^i(1-C_k^i)D_k^i 0.6 + \zeta_k^i - \eta_k^i - \lambda)^2}{4P_k^i(1-C_k^i)D_k^i\frac{R_k^i}{30}} + \eta_k^i\frac{0.6}{\frac{R_k^i}{30}}\Big) + \lambda U + \lambda\sum_{k=1}^{248} d_k.$$

For this problem the KKT conditions lead to

$$\begin{aligned}
&\frac{\partial(\cdot)}{\partial\lambda} = U + \sum_{k=1}^{248} d_k - \sum_{k=1}^{248}\sum_{i=1}^{10}\frac{P_k^i(1-C_k^i)D_k^i 0.6 + \zeta_k^i - \eta_k^i - \lambda}{2P_k^i(1-C_k^i)D_k^i\frac{R_k^i}{30}} = 0 \qquad \text{or} \qquad \lambda = 0 \\
&\frac{\partial(\cdot)}{\partial\eta_k^i} = -\frac{P_k^i(1-C_k^i)D_k^i 0.6 + \zeta_k^i - \eta_k^i - \lambda}{2P_k^i(1-C_k^i)D_k^i\frac{R_k^i}{30}} + \frac{0.6}{\frac{R_k^i}{30}} = 0 \qquad \text{or} \qquad \eta_k^i = 0 \\
&\frac{\partial(\cdot)}{\partial\zeta_k^i} = \frac{P_k^i(1-C_k^i)D_k^i 0.6 + \zeta_k^i - \eta_k^i - \lambda}{2P_k^i(1-C_k^i)D_k^i\frac{R_k^i}{30}} = 0 \qquad \text{or} \qquad \zeta_k^i = 0.
\end{aligned}$$

Assuming that all the u_k^i constraints are inactive, this would lead to all the η_k^i and ζ_k^i equal to zero an therefore we would need

$$\sum_{k=1}^{248}\sum_{i=1}^{10}\frac{P_k^i(1-C_k^i)D_k^i 0.6 - \lambda}{2P_k^i(1-C_k^i)D_k^i\frac{R_k^i}{30}} = U + \sum_{k=1}^{248} d_k \quad \Leftrightarrow \quad \lambda = \frac{\sum_{k=1}^{248}\sum_{i=1}^{10}\frac{0.6}{2\frac{R_k^i}{30}} - U - \sum_{k=1}^{248} d_k}{\sum_{k=1}^{248}\sum_{i=1}^{10}\frac{1}{2P_k^i(1-C_k^i)D_k^i\frac{R_k^i}{30}}} \geqslant 0 \quad \text{or} \quad \lambda = 0$$

and

$$u_k^i = \bar{u}_k^i - \bar{\mu}_k^i \max\left\{0, \sum_{\bar{k}=1}^{248}\sum_{i=1}^{10} \bar{u}_{\bar{k}}^i - U - \sum_{\bar{k}=1}^{248} d_{\bar{k}}\right\}, \quad \bar{u}_k^i \coloneqq \frac{0.6}{2\frac{R_k^i}{30}}, \quad \bar{\mu}_k^i \coloneqq \frac{\frac{1}{2P_k^i(1-C_k^i)D_k^i\frac{R_k^i}{30}}}{\sum_{\bar{k}=1}^{248}\sum_{i=1}^{10}\frac{1}{2P_{\bar{k}}^i(1-C_{\bar{k}}^i)D_{\bar{k}}^i\frac{R_{\bar{k}}^i}{30}}}$$

We can view the term being subtracted as a normalizing term that makes sure that the u_k^i add up to the constraint U (in case controls equal to the first term would exceed it). Note that if the formula above for u_k^i ever becomes negative, then the corresponding ζ_k^i cannot be zero and we must have

$$\frac{\partial(\cdot)}{\partial\zeta_k^i} = \frac{P_k^i(1-C_k^i)D_k^i 0.6 + \zeta_k^i - \lambda}{2P_k^i(1-C_k^i)D_k^i\frac{R_k^i}{30}} = 0 \quad \Rightarrow \quad \zeta_k^i = \lambda - P_k^i(1-C_k^i)D_k^i 0.6 \quad \Rightarrow \quad u_k^i = 0,$$

which would have to be taken into account in the formula for λ.

The following section proposes a moving horizon optimization scheme that allows the attacker to use past data and based on that make predictions on when to allocate her resources in order to maximize the points at the end of the game.

3 Optimization using unknown data

A moving horizon optimization scheme is an ideal approach since the attacker continuously extend her horizon as time evolves, which allows her to incorporate new data information at any point of time. If the horizon is small (e.g. $N \leqslant 5$) one can make predictions of when to launder the money through the services to obtain the maximum number of points at the end of the competition. At time instant t, we will take into account only the last measurements available whereas the older measurements will be dealt by summarizing the history of the system before time instant $t-N$. The results show that this optimization scheme yields result really close to the scheme in the previous section with known data.

We want to solve the following optimization problem repeatedly,

$$\begin{aligned}
\text{maximize} \quad & \sum_{k=t}^{t+N}\sum_{i=1}^{10}\mathbb{E}\left[\rho_k^i p_k^i u_k^i\right] \\
\text{subject to} \quad & \sum_{k=t}^{t+N}\sum_{i=1}^{10} u_k^i - \sum_{k=t}^{t+N} d_k \leqslant U \\
\text{w.r.t.} \quad & u_k^i \in [0,\infty),\ \forall i \in \{1,2,\ldots,10\},\ t \in \{1,2,\ldots,243\},
\end{aligned}$$

where d_k is the cut the team receives because of the owned services, also for each $i \in \{1,2,\ldots,10\}$, the $\rho_1^i, \rho_2^i, \ldots, \rho_{248}^i$ are independent and identically distributed (i.i.d.) Bernoulli random variables with

$$\mathrm{P}(\rho_k^i = 0) = \min\{\alpha_k^i u_k + 0.4, 1\}$$

where $p_k^i \equiv P_k^i(1-C_k^i)D_k^i \in [0,1]$ and $\alpha_k^i \equiv \frac{R_k^i}{30} \in [0, 0.03]$.

Now we will define $\hat{y}_k^i = \left[\bar{\beta}_k^i \ \bar{\alpha}_k^i\right]$, where $\bar{\beta}_k^i = 0.6p_k^i$, $\bar{\alpha}_k^i = \alpha_k^i p_k^i$ are given by the following Auto Regressive (AR) models with ϵ_k^i a white noise process with zero-mean and variance σ_ϵ^2

$$\bar{\beta}_k^i = h_0^i\bar{\beta}_{k-1}^i + h_1^i\bar{\beta}_{k-2}^i + \epsilon_k^i$$

$$\bar{\alpha}_k^i = c_0^i\bar{\alpha}_{k-1}^i + c_1^i\bar{\alpha}_{k-2}^i + \epsilon_k^i.$$

The AR models can be described by the following stochastic dynamic systems where the matrices A^i, B^i, H^i are going to be defined in a subsequent subsection after employing system identification in the available data and w_k^i is a sequence of a random noise process with zero-mean and variance σ_w^2 ,

$$\begin{aligned}\hat{x}_{k+1}^i &= A^i\hat{x}_k^i + B^i w_k^i\\ \hat{y}_k^i &= H^i\hat{x}_k^i.\end{aligned}$$

Now we can write the optimization problem as

$$\begin{aligned}
\text{maximize} \quad & \sum_{k=t}^{t+N}\sum_{i=1}^{10}\mathbb{E}[y_k^i]\left[u_k^i \;\; -{u_k^i}^2\right]^T\\
\text{subject to} \quad & \sum_{k=t}^{t+N}\sum_{i=1}^{10}u_k^i - \sum_{k=t}^{t+N}d_k \leqslant U\\
& \hat{x}_{k+1}^i = A^i\hat{x}_k^i + B^i w_k^i\\
& \hat{y}_k^i = H^i\hat{x}_k^i\\
& x_0^i = x(0)^i\\
\text{w.r.t.} \quad & u_k^i \in [0, \frac{0.6}{\mathbb{E}[a_k^i]}],\ \forall i \in \{1,2,\dots,10\},\ t \in \{1,2,\dots,243\}
\end{aligned}$$

which is a concave maximization problem (convex minimization) with linear constrains, which is easy to solve numerically and for which the duality gap is zero. The dual problem is given by

$$\begin{aligned}
J^\perp \coloneqq & \max_{\lambda\geqslant 0,\lambda_{k1}^i\geqslant 0,\lambda_{k2}^i\geqslant 0,\eta_k^i\geqslant 0,\zeta_k^i\geqslant 0}\ \max_{u_k^i\in\mathbb{R}}\ \sum_{k=t}^{t+N}\sum_{i=1}^{10}\mathbb{E}[\hat{y}_k^i]\left[u_k^i \;\; -{u_k^i}^2\right]^T + {\lambda_{k1}^i}^T\mathbb{E}(\hat{x}_{k+1}^i - A^i\hat{x}_k^i - B^i w_k^i) + \lambda_{k2}^i\mathbb{E}(\hat{y}_k^i - C^i x_k^i)\\
& - \lambda\Big(\sum_{k=t}^{t+N}\sum_{i=1}^{10}u_k^i - \sum_{k=1}^{248}d_k - U\Big) - \sum_{k=t}^{t+N}\sum_{i=1}^{10}\eta_k^i\Big(u_k^i - \frac{0.6}{\mathbb{E}[a_k^i]}\Big) + \sum_{k=t}^{t+N}\sum_{i=1}^{10}\zeta_k^i u_k^i\\
= & \max_{\lambda\geqslant 0,\lambda_{k1}^i\geqslant 0,\lambda_{k2}^i\geqslant 0,\eta_k^i\geqslant 0,\zeta_k^i\geqslant 0}\ \max_{u_k^i\in\mathbb{R}}\ \sum_{k=t}^{t+N}\sum_{i=1}^{10}\mathbb{E}\Big(\bar{\beta}_k^i u_k^i - \bar{\alpha}_k^i {u_k^i}^2 - \lambda u_k^i - \eta_k^i\Big(u_k^i - \frac{0.6}{\alpha_k^i}\Big) + \zeta_k^i u_k^i\Big)\\
& + \lambda U + \lambda\sum_{k=t}^{t+N}d_k + {\lambda_{k1}^i}^T\mathbb{E}(\hat{x}_{k+1}^i - A^i\hat{x}_k^i - B^i w_k^i) + \lambda_2\mathbb{E}(y_k^i - H^i\hat{x}_k^i)\\
= & \max_{\lambda\geqslant 0,\lambda_{k1}^i\geqslant 0,\lambda_{k2}^i\geqslant 0,\eta_k^i\geqslant 0,\zeta_k^i\geqslant 0}\ \max_{u_k^i\in\mathbb{R}}\ \sum_{k=t}^{t+N}\sum_{i=1}^{N}\mathbb{E}\Big((\bar{\beta}_k^i - \lambda - \eta_k^i + \zeta_k^i)u_k^i - \bar{\alpha}_k^i {u_k^i}^2 + \eta_k^i\frac{0.6}{\alpha_k^i}\Big)\\
& + \lambda U + \lambda\sum_{k=t}^{t+N}d_k + {\lambda_{k1}^i}^T\mathbb{E}(\hat{x}_{k+1}^i - A^i\hat{x}_k^i - B^i w_k^i) + \lambda_{k2}^i\mathbb{E}(\hat{y}_k^i - H^i\hat{x}_k^i).
\end{aligned}$$

The inner maximization can be solved using standard calculus and is achieved for

$$u_k^i = \frac{\mathbb{E}[\bar{\beta}_k^i] + \zeta_k^i - \eta_k^i - \lambda}{2\mathbb{E}[\bar{\alpha}_k^i]},$$

yielding

$$J^\perp \coloneqq \max_{\lambda\geqslant 0,\lambda_{k1}^i\geqslant 0,\lambda_{k2}^i\geqslant 0,\eta_k^i\geqslant 0,\zeta_k^i\geqslant 0}\ \sum_{k=t}^{t+N}\sum_{i=1}^{10}\mathbb{E}\Big(\frac{(\bar{\beta}_k^i + \zeta_k^i - \eta_k^i - \lambda)^2}{4\bar{\alpha}_k^i} + \eta_k^i\frac{0.6}{\alpha_k^i}\Big) + \lambda U + \lambda\sum_{k=t}^{t+N}d_k$$

$$+ {\lambda^i_{k1}}^T \mathbb{E}(\hat{x}^i_{k+1} - A^i \hat{x}^i_k - B^i w^i_k) + \lambda^i_{k2} \mathbb{E}(\hat{y}^i_k - H^i \hat{x}^i_k).$$

For this problem the KKT conditions lead to

$$\begin{aligned}
&\frac{\partial(\cdot)}{\partial \lambda} = U + \sum_{k=t}^{t+N} d_k - \sum_{k=t}^{t+N} \sum_{i=1}^{10} \frac{\mathbb{E}[\bar{\beta}^i_k] + \zeta^i_k - \eta^i_k - \lambda}{2\mathbb{E}[\bar{\alpha}^i_k]} = 0 && \text{or} && \lambda = 0 \\
&\frac{\partial(\cdot)}{\partial \eta^i_k} = \frac{\mathbb{E}[\bar{\beta}^i_k] + \zeta^i_k - \eta^i_k - \lambda}{2\mathbb{E}[\bar{\alpha}^i_k]} + \frac{0.6}{\mathbb{E}[\alpha^i_k]} = 0 && \text{or} && \eta^i_k = 0 \\
&\frac{\partial(\cdot)}{\partial \zeta^i_k} = \frac{\mathbb{E}[\bar{\beta}^i_k] + \zeta^i_k - \eta^i_k - \lambda}{2\mathbb{E}[\bar{\alpha}^i_k]} = 0 && \text{or} && \zeta^i_k = 0 \\
&\frac{\partial(\cdot)}{\partial \lambda^i_{k1}} = \mathbb{E}[\hat{x}^i_{k+1}] - A^i \mathbb{E}[\hat{x}^i_k] = 0 && \text{or} && \lambda^i_{k1} = 0 \\
&\frac{\partial(\cdot)}{\partial \lambda^i_{k2}} = \mathbb{E}[\hat{y}^i_k] - H^i \mathbb{E}[\hat{x}^i_k] = 0 && \text{or} && \lambda_2 = 0.
\end{aligned}$$

Assuming that all the u^i_k constraints are inactive, this would lead to all the η^i_k and ζ^i_k equal to zero and therefore we would need

$$\sum_{k=t}^{t+N} \sum_{i=1}^{10} \frac{\mathbb{E}[\bar{\beta}^i_k] - \lambda}{2\mathbb{E}[\bar{\alpha}^i_k]} = U + \sum_{k=t}^{t+N} d_k \quad \Leftrightarrow \quad \lambda = \frac{\sum_{k=t}^{t+N} \sum_{i=1}^{10} \frac{0.6}{2\mathbb{E}[\alpha^i_k]} - U - \sum_{k=t}^{t+N} d_k}{\sum_{k=t}^{t+N} \sum_{i=1}^{10} \frac{1}{2\mathbb{E}[\bar{\alpha}^i_k]}} \geqslant 0 \quad \text{or} \quad \lambda = 0$$

and

$$u^i_k = \bar{u}^i_k - \bar{\mu}^i_k \max\left\{0, \sum_{\bar{k}=t}^{t+N} \sum_{i=1}^{10} \bar{u}^i_{\bar{k}} - U - \sum_{\bar{k}=t}^{t+N} d_{\bar{k}}\right\}, \qquad \bar{u}^i_k \coloneqq \frac{0.6}{2\mathbb{E}[\alpha^i_k]}, \qquad \bar{\mu}^i_k \coloneqq \frac{\frac{1}{2\mathbb{E}[\bar{\alpha}^i_k]}}{\sum_{\bar{k}=t}^{t+N} \sum_{i=1}^{N} \frac{1}{2\mathbb{E}[\bar{\alpha}^i_k]}}.$$

We can view the term being subtracted as a normalizing term that makes sure that the u^i_k add up to the constraint U (in case controls equal to the first term would exceed it). Note that if the formula above for u^i_k ever becomes negative, then the corresponding ζ^i_k cannot be zero and we must have

$$\frac{\partial(\cdot)}{\partial \zeta^i_k} = \frac{\mathbb{E}[\bar{\beta}^i_k] + \zeta^i_k - \lambda}{2\mathbb{E}[\bar{\alpha}^i_k]} = 0 \quad \Rightarrow \quad \zeta^i_k = \lambda - \mathbb{E}[\bar{\beta}^i_k] \quad \Rightarrow \quad u^i_k = 0,$$

which would have to be taken into account in the formula for λ.

3.1 Identification of the data coming into the system from the 10 services

In this subsection we will provide the state space description (Auto Regression Models) for every of the 10 services. The model $\hat{y}^i_k = [\hat{y}^i_{\bar{\beta}k} \ \hat{y}^i_{\bar{\alpha}k}]$ for every service is given by

$$\hat{y}^i_{(.)k+1} - \hat{y}^i_{(.)k} = w_k$$

where $(.)$ can be $\bar{\alpha}$ or $\bar{\beta}$ and w_k is an independent sequence of random noise with zero-mean and variance σ^2_w. By taking the z-transform one has,

$$\hat{Y}_{(.)}(z)^i = \frac{W(z)}{z-1}.$$

This state space representation found, will be used in the previous optimization framework. The state

models are, for the first service

$$\hat{xb}_{k+1}^{1} = \begin{bmatrix} 0.4578 & 0.5422 \\ 1 & 0 \end{bmatrix} \hat{xb}_{k}^{1} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} w_k$$

$$\hat{yb}_{k}^{1} = \begin{bmatrix} 0.3701 & 0 \end{bmatrix} \hat{xb}_{k}^{1}$$

$$\hat{xa}_{k+1}^{1} = \begin{bmatrix} 0.5428 & 0.4572 \\ 1 & 0 \end{bmatrix} \hat{xa}_{k}^{1} + \begin{bmatrix} 0.0625 \\ 0 \end{bmatrix} w_k$$

$$\hat{ya}_{k}^{1} = \begin{bmatrix} 0.1044 & 0 \end{bmatrix} \hat{xa}_{k}^{1}.$$

For the second service,

$$\hat{xb}_{k+1}^{2} = \begin{bmatrix} 0.5177 & 0.4823 \\ 1 & 0 \end{bmatrix} \hat{xb}_{k}^{2} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} w_k$$

$$\hat{yb}_{k}^{2} = \begin{bmatrix} 0.3256 & 0 \end{bmatrix} \hat{xb}_{k}^{2}$$

$$\hat{xa}_{k+1}^{2} = \begin{bmatrix} 0.53 & 0.47 \\ 1 & 0 \end{bmatrix} \hat{xa}_{k}^{2} + \begin{bmatrix} 0.0625 \\ 0 \end{bmatrix} w_k$$

$$\hat{ya}_{k}^{2} = \begin{bmatrix} 0.1072 & 0 \end{bmatrix} \hat{xa}_{k}^{2}.$$

For the third one,

$$\hat{xb}_{k+1}^{3} = \begin{bmatrix} 0.4512 & 0.5488 \\ 1 & 0 \end{bmatrix} \hat{xb}_{k}^{3} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} w_k$$

$$\hat{yb}_{k}^{3} = \begin{bmatrix} 0.3640 & 0 \end{bmatrix} \hat{xb}_{k}^{3}$$

$$\hat{xa}_{k+1}^{3} = \begin{bmatrix} 0.4731 & 0.5269 \\ 1 & 0 \end{bmatrix} \hat{xa}_{k}^{3} + \begin{bmatrix} 0.0625 \\ 0 \end{bmatrix} w_k$$

$$\hat{ya}_{k}^{3} = \begin{bmatrix} 0.1080 & 0 \end{bmatrix} \hat{xa}_{k}^{3}.$$

For the fourth one,

$$\hat{xb}_{k+1}^{4} = \begin{bmatrix} 0.5095 & 0.4905 \\ 1 & 0 \end{bmatrix} \hat{xb}_{k}^{4} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} w_k$$

$$\hat{yb}_{k}^{4} = \begin{bmatrix} 0.3219 & 0 \end{bmatrix} \hat{xb}_{k}^{4}$$

$$\hat{xa}_{k+1}^{4} = \begin{bmatrix} 0.5672 & 0.4328 \\ 1 & 0 \end{bmatrix} \hat{xa}_{k}^{4} + \begin{bmatrix} 0.0625 \\ 0 \end{bmatrix} w_k$$

$$\hat{ya}_k^4 = \begin{bmatrix} 0.0837 & 0 \end{bmatrix} \hat{xa}_k^1.$$

For the fifth one,

$$\hat{xb}_{k+1}^5 = \begin{bmatrix} 0.4694 & 0.5306 \\ 1 & 0 \end{bmatrix} \hat{xb}_k^5 + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} w_k$$

$$\hat{yb}_k^5 = \begin{bmatrix} 0.3231 & 0 \end{bmatrix} \hat{xb}_k^5$$

$$\hat{xa}_{k+1}^5 = \begin{bmatrix} 0.5313 & 0.4649 \\ 1 & 0 \end{bmatrix} \hat{xa}_k^5 + \begin{bmatrix} 0.0625 \\ 0 \end{bmatrix} w_k$$

$$\hat{ya}_k^5 = \begin{bmatrix} 0.1080 & 0 \end{bmatrix} \hat{xa}_k^1.$$

For the sixth one,

$$\hat{xb}_{k+1}^6 = \begin{bmatrix} 0.5164 & 0.4836 \\ 1 & 0 \end{bmatrix} \hat{xb}_k^6 + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} w_k$$

$$\hat{yb}_k^6 = \begin{bmatrix} 0.3315 & 0 \end{bmatrix} \hat{xb}_k^6$$

$$\hat{xa}_{k+1}^6 = \begin{bmatrix} 0.5682 & 0.4318 \\ 1 & 0 \end{bmatrix} \hat{xa}_k^6 + \begin{bmatrix} 0.0625 \\ 0 \end{bmatrix} w_k$$

$$\hat{ya}_k^6 = \begin{bmatrix} 0.1062 & 0 \end{bmatrix} \hat{xa}_k^6.$$

For the seventh one,

$$\hat{xb}_{k+1}^7 = \begin{bmatrix} 0.5133 & 0.4867 \\ 1 & 0 \end{bmatrix} \hat{xb}_k^7 + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} w_k$$

$$\hat{yb}_k^7 = \begin{bmatrix} 0.3477 & 0 \end{bmatrix} \hat{xb}_k^7$$

$$\hat{xa}_{k+1}^7 = \begin{bmatrix} 0.5069 & 0.4931 \\ 1 & 0 \end{bmatrix} \hat{xa}_k^7 + \begin{bmatrix} 0.0625 \\ 0 \end{bmatrix} w_k$$

$$\hat{ya}_k^7 = \begin{bmatrix} 0.0942 & 0 \end{bmatrix} \hat{xa}_k^7.$$

For the eight one,

$$\hat{xb}_{k+1}^8 = \begin{bmatrix} 0.5207 & 0.4793 \\ 1 & 0 \end{bmatrix} \hat{xb}_k^8 + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} w_k$$

$$\hat{yb}_k^8 = \begin{bmatrix} 0.3316 & 0 \end{bmatrix} \hat{xb}_k^8$$

$$\hat{xa}_{k+1}^{8} = \begin{bmatrix} 0.5606 & 0.4394 \\ 1 & 0 \end{bmatrix} \hat{xa}_{k}^{8} + \begin{bmatrix} 0.0625 \\ 0 \end{bmatrix} w_k$$

$$\hat{ya}_{k}^{8} = \begin{bmatrix} 0.0996 & 0 \end{bmatrix} \hat{xa}_{k}^{8}.$$

For the nine one,

$$\hat{xb}_{k+1}^{9} = \begin{bmatrix} 0.4858 & 0.5142 \\ 1 & 0 \end{bmatrix} \hat{xb}_{k}^{9} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} w_k$$

$$\hat{yb}_{k}^{9} = \begin{bmatrix} 0.2924 & 0 \end{bmatrix} \hat{xb}_{k}^{9}$$

$$\hat{xa}_{k+1}^{9} = \begin{bmatrix} 0.5410 & 0.4590 \\ 1 & 0 \end{bmatrix} \hat{xa}_{k}^{9} + \begin{bmatrix} 0.0625 \\ 0 \end{bmatrix} w_k$$

$$\hat{ya}_{k}^{9} = \begin{bmatrix} 0.1083 & 0 \end{bmatrix} \hat{xa}_{k}^{9}.$$

For the last service,

$$\hat{xb}_{k+1}^{10} = \begin{bmatrix} 0.5474 & 0.4526 \\ 1 & 0 \end{bmatrix} \hat{xb}_{k}^{10} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} w_k$$

$$\hat{yb}_{k}^{10} = \begin{bmatrix} 0.3060 & 0 \end{bmatrix} \hat{xb}_{k}^{10}$$

$$\hat{xa}_{k+1}^{10} = \begin{bmatrix} 0.4812 & 0.5188 \\ 1 & 0 \end{bmatrix} \hat{xa}_{k}^{10} + \begin{bmatrix} 0.0625 \\ 0 \end{bmatrix} w_k$$

$$\hat{ya}_{k}^{10} = \begin{bmatrix} 0.1045 & 0 \end{bmatrix} \hat{xa}_{k}^{1}.$$

4 Simulations

This section presents simulation examples of the optimization schemes proposed by using data from the competition. All the simulations have been implemented through a Matlab-based convex optimization solver such as CVX [1]. Figures 1-3 show the values of $1 - C$, P, R for every service. Figures 4-6 show when is good for the "Oracle" team to bet the money such that she maximizes the total number of points at the end of the game. We also assume that we have an upper bound on the money to launder and also that the defense level is $D = 1$. Figures 7-9 show the points collected by the "Oracle" team through the 10 services. Figures 11-12 show the points and the money laundered through the 10 services from the teams in the competition. Finally Figure 13 shows the total money laundered through the 10 services by the "Oracle" team. The "Oracle" team ended up with 1987 points.

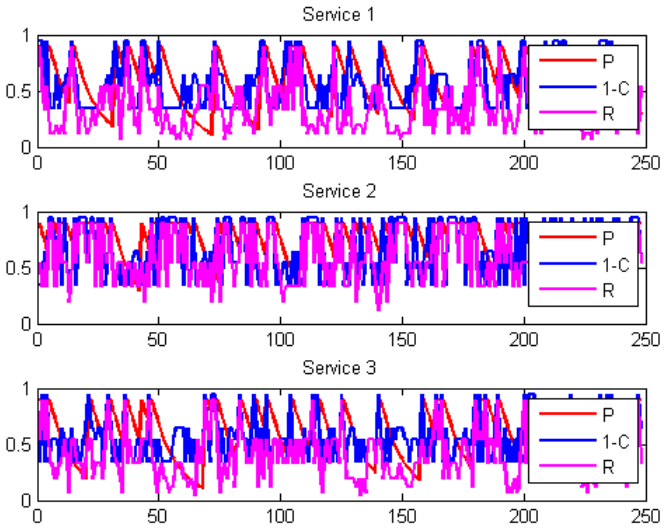

Figure 1: CPR values

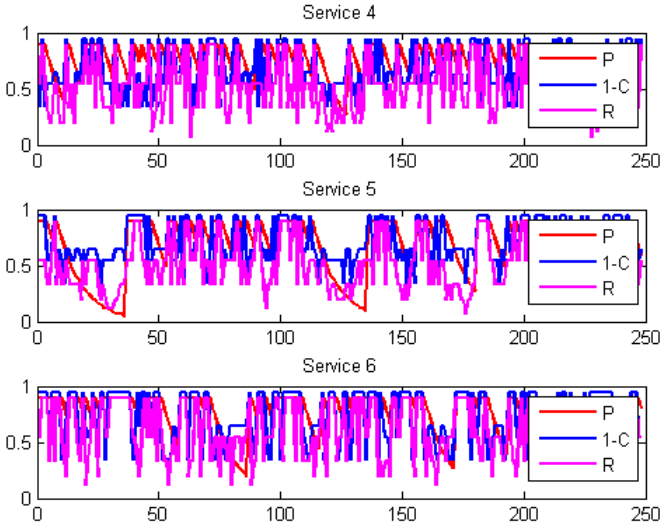

Figure 2: CPR values

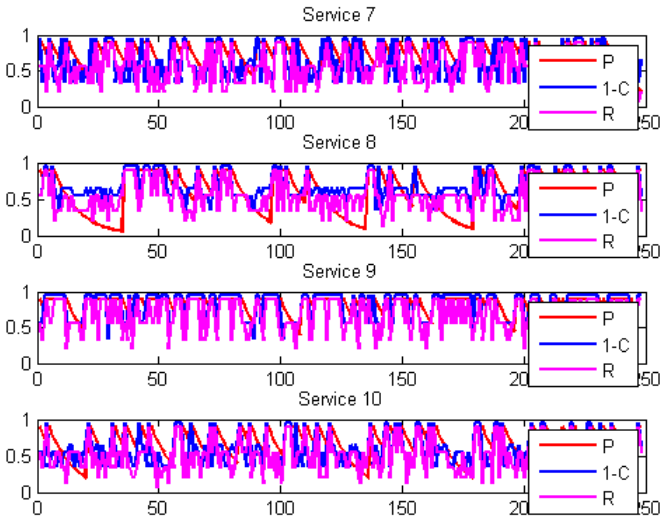

Figure 3: CPR values

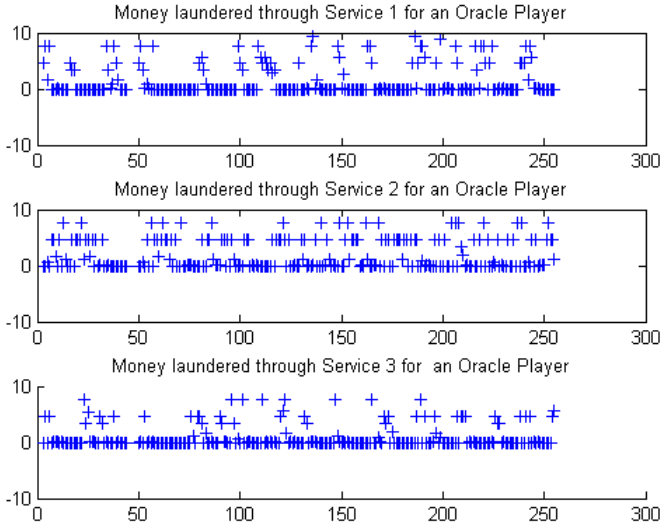

Figure 4: Money laundered from Oracle Team

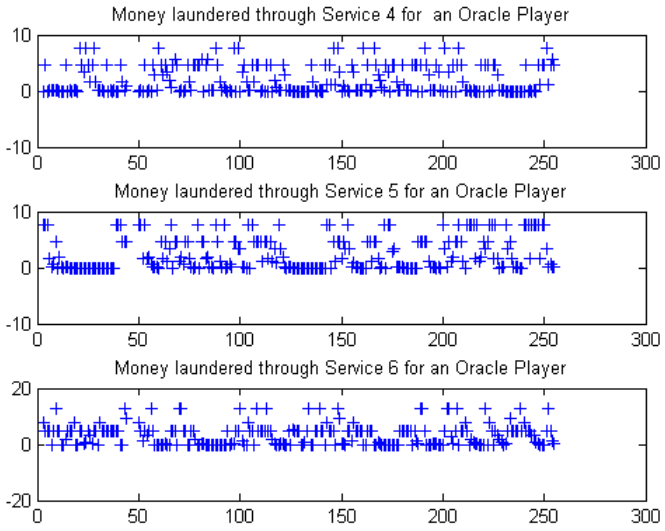

Figure 5: Money laundered from Oracle Team

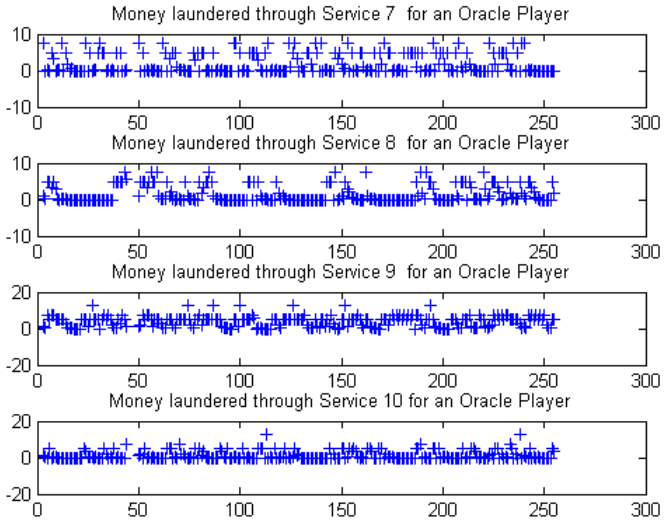

Figure 6: Money laundered from Oracle Team

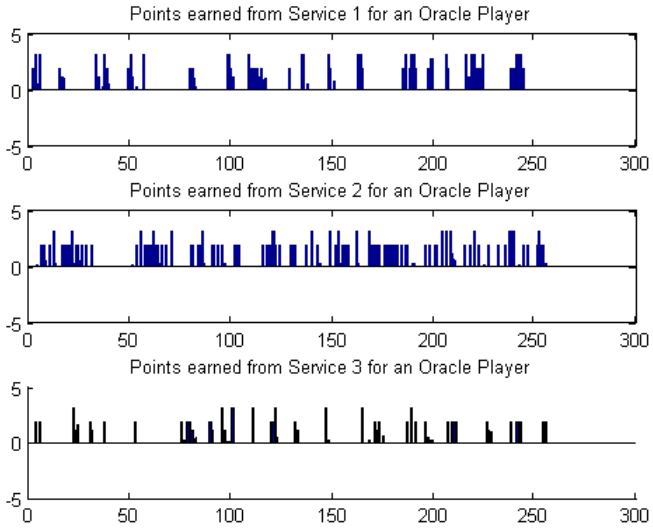

Figure 7: Points earned from Oracle Team

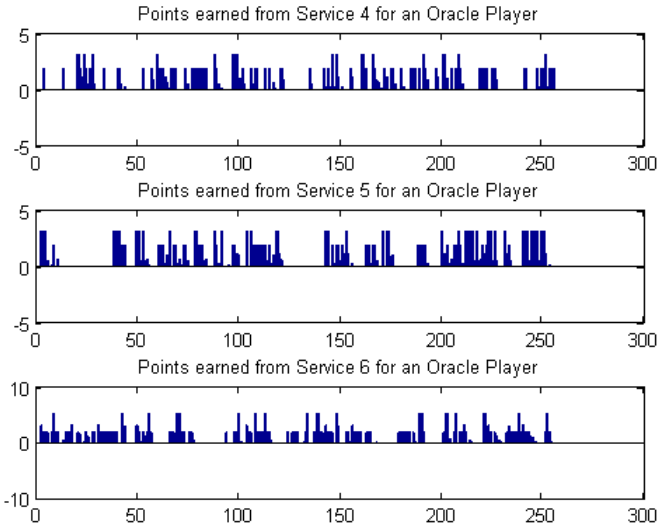

Figure 8: Points earned from Oracle Team

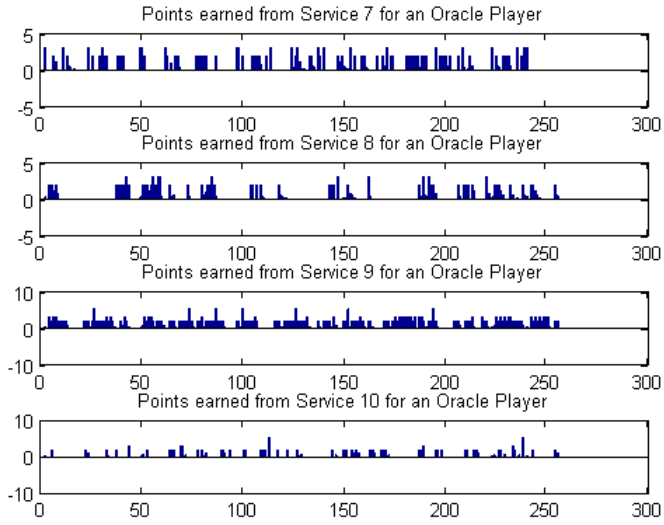

Figure 9: Points earned from Oracle Team

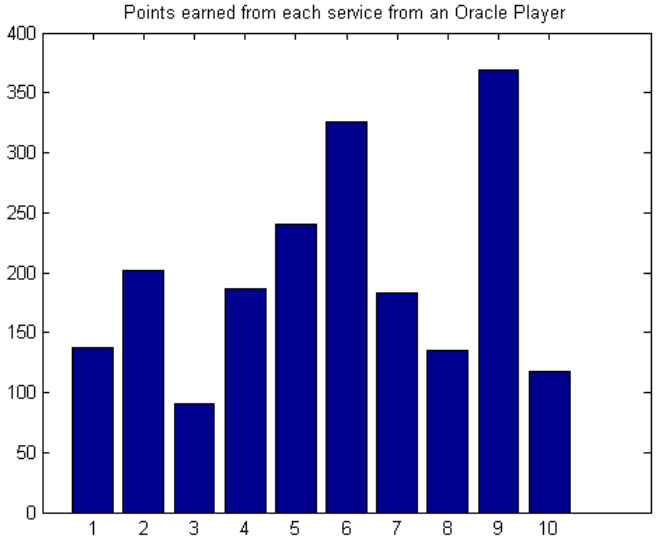

Figure 10: Points earned from Oracle Team

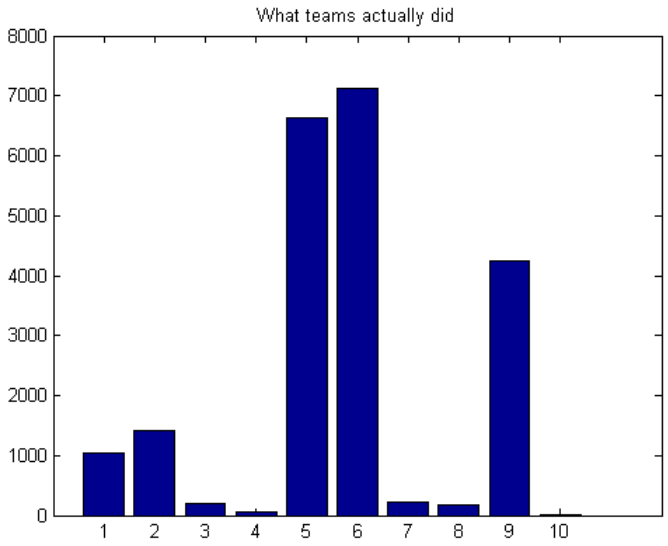

Figure 11: Points earned through the services from all the teams participated in the competition

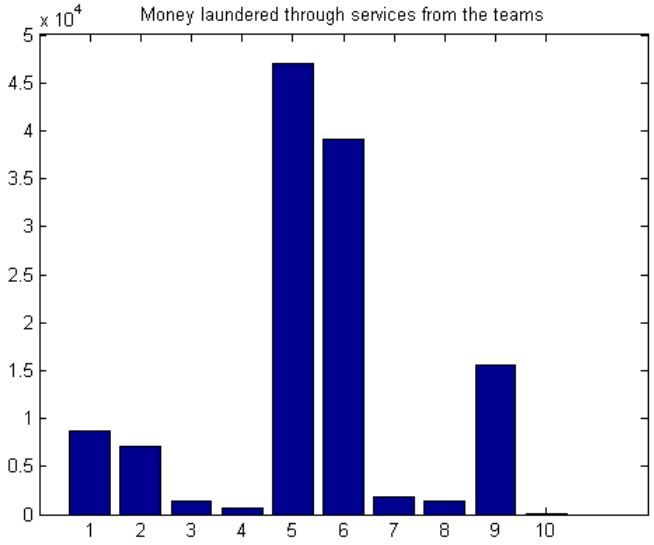

Figure 12: Money laundered from all the teams participated in the competition through the 10 services

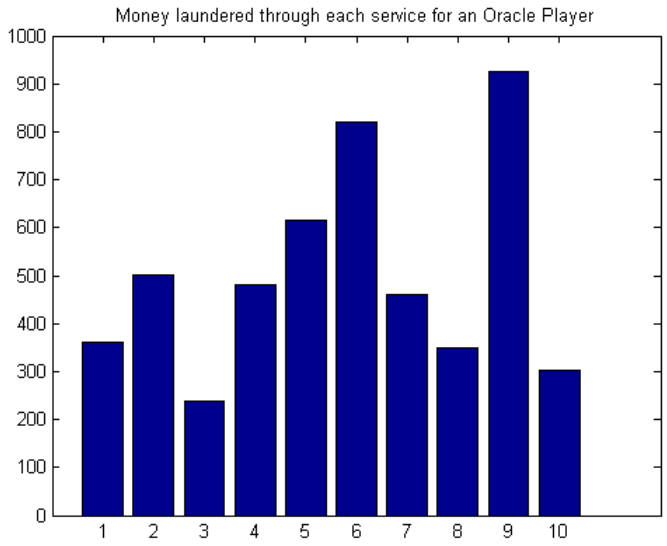

Figure 13: Money laundered from Oracle team through the 10 services

5 Players with different level of sophistication

In the previous section we considered an "Oracle" player (points earned were 1987) who was laundering his money through all of the 10 services in a sophisticated way to achieve the most points at the end of competition. In this section we will consider two other kind of players, namely the "Average" player, and the "Naive" player. As one sees from Figure 10 the services 5, 6 and 9 were the services that can give the most points to the teams. For that reason an average player will attack those three services mostly (by assuming that attacking the same team does not affect his outcome) at all time (see what team 3 did in the competition). Figures 14-19 show the "hard to get into" services 2, 4, 5, 6, 9 attacked by an average team (points earned were 1821). Figures 20-25 show a naive team (points earned were 1721)that can only access "easy to get into" services such as 1, 2, 5, 6, 7, 8.

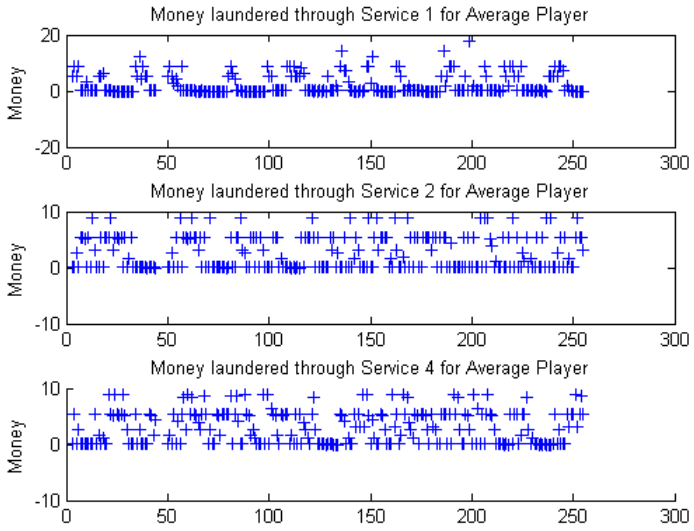

Figure 14: Money laundered from Average Team

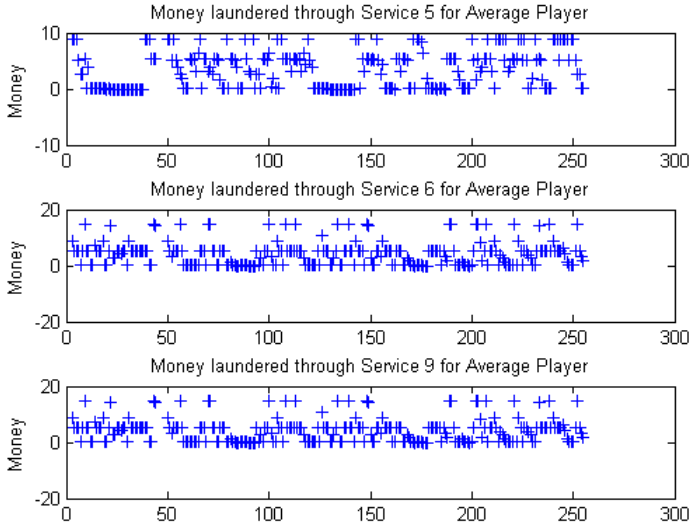

Figure 15: Money laundered from Average Team

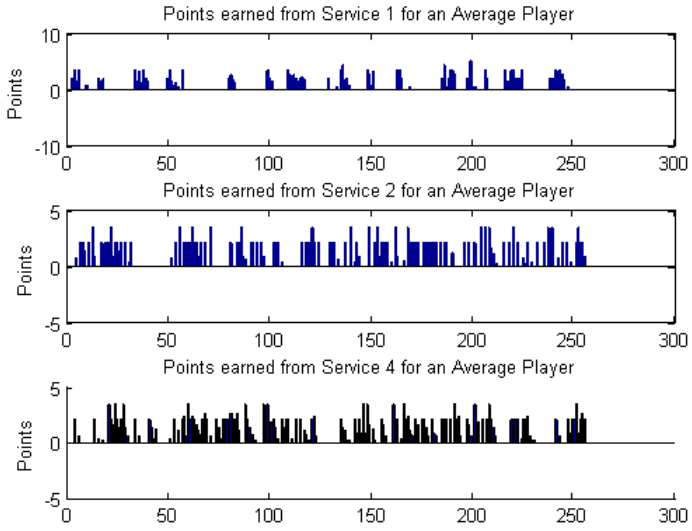

Figure 16: Points earned from Average Team

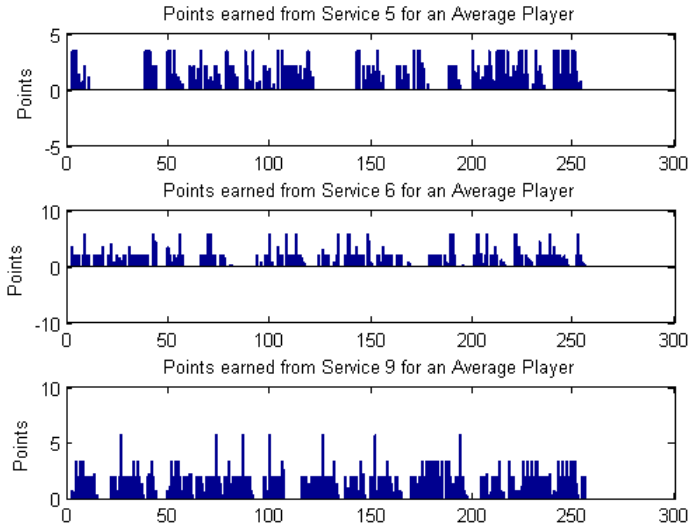

Figure 17: Points earned from Average Team

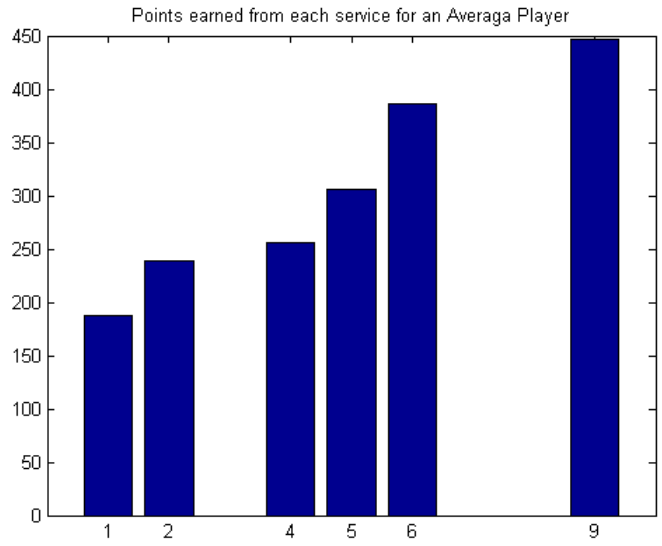

Figure 18: Points earned from Average Team

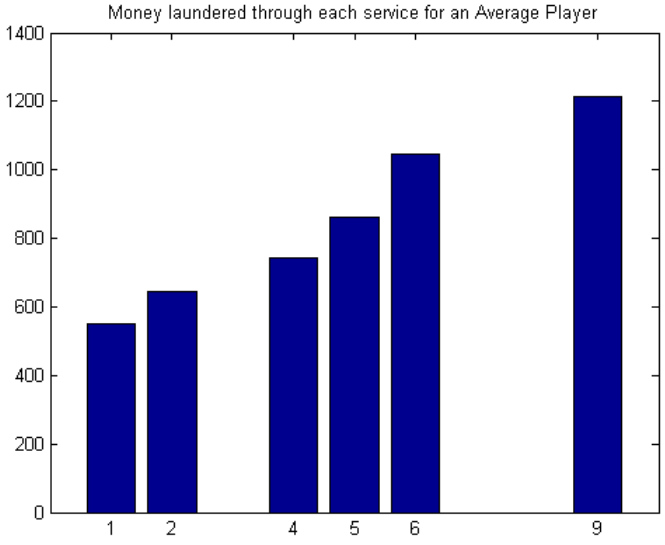

Figure 19: Money laundered from Average Team

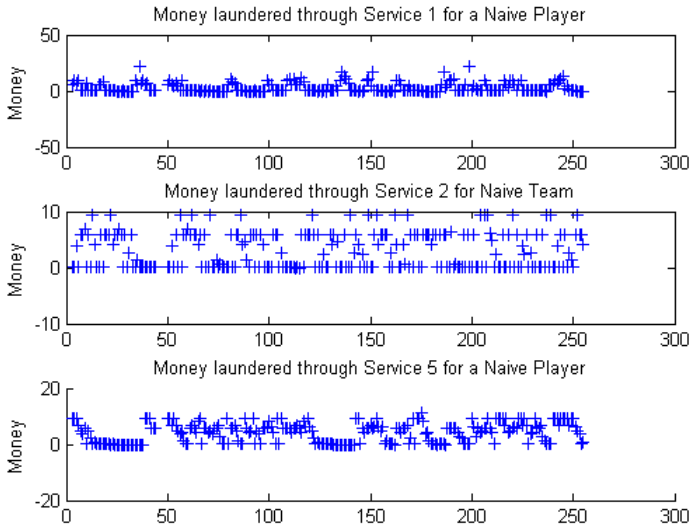

Figure 20: Money laundered from Naive Team

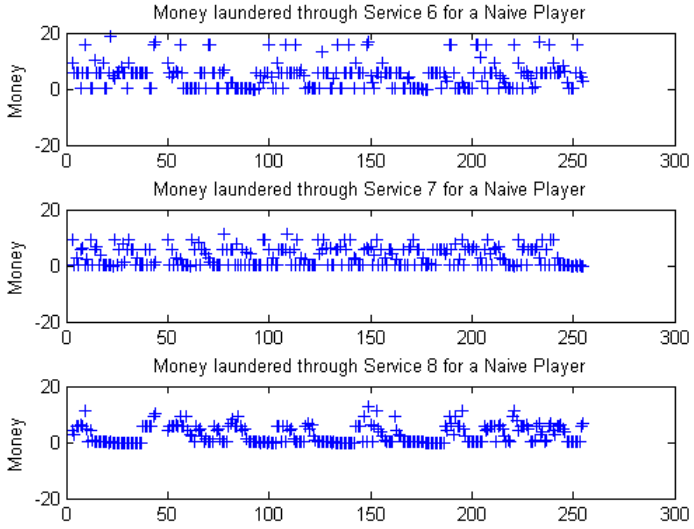

Figure 21: Money laundered from Naive Team

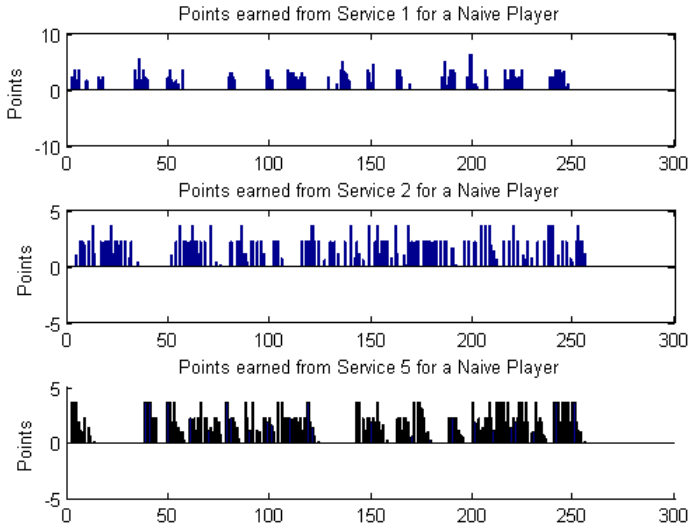

Figure 22: Points earned from Naive Team

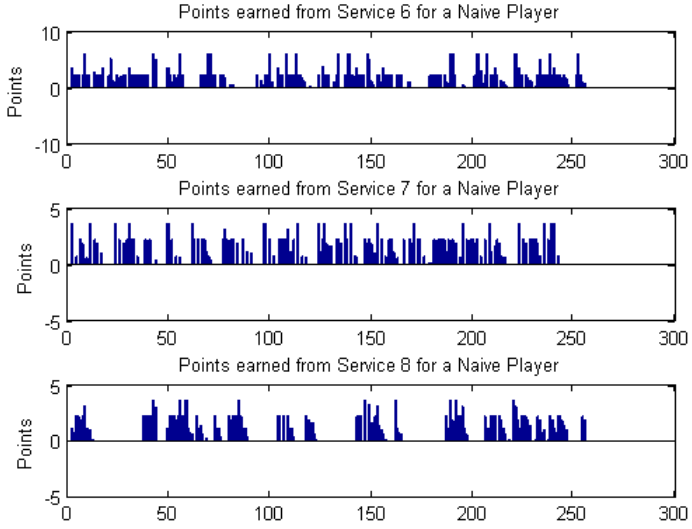

Figure 23: Points earned from Naive Team

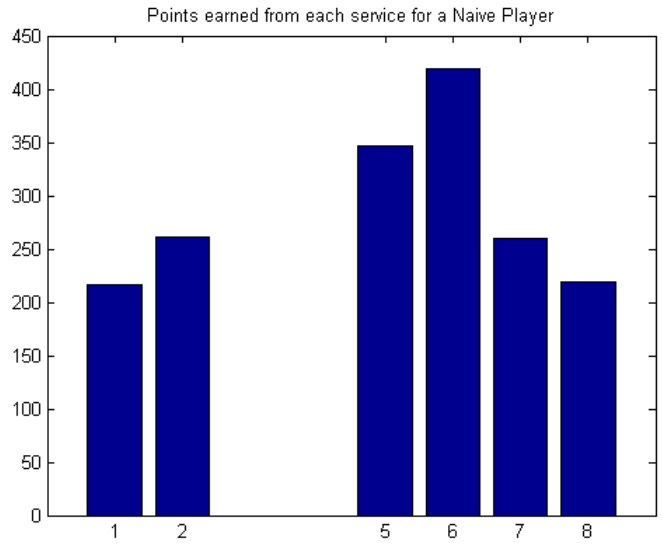

Figure 24: Points earned from Naive Team

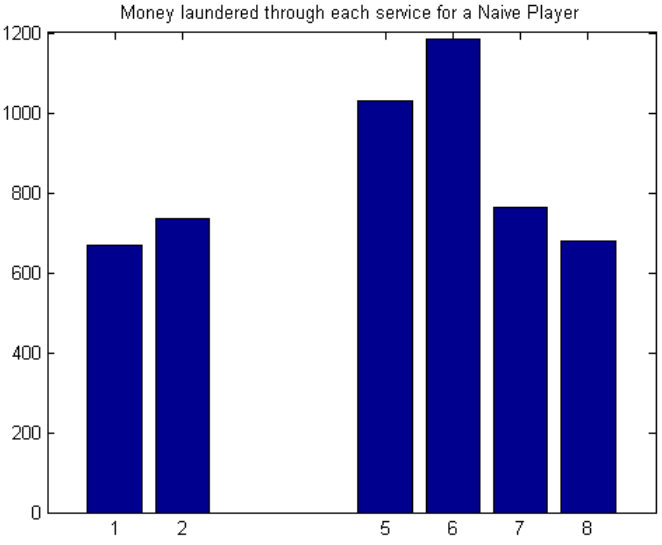

Figure 25: Money laundered from Naive Team

6 Discussion

It is clear from the previous simulation examples and especially from Figures 10 and 11 that the top services to attack were 5, 6 and 9. The teams in the competition attacked mostly 5 and 6 because 9 was a hard service to get into. Only the top teams found out how to attack service 9 and only at the end of the game. For all the "optimal" attackers described before, we assumed that they know the vulnerabilities

of every service and can attack any service at any time. If the teams in the competition were able to exploit all services from the beginning of the game, the Figures 10 and 11 would have been much closer.

Also we can conclude that a team that had access to all the services, like the "Oracle" team, would have scored more points than any other team. The top two teams in the competition followed the "Average" team strategies and they got similar points to this team (top team in the actual competition got 1780 points).

7 Future work

Our greatest concentration of future work will be on supporting and providing analysis tools for what if scenarios based on past data and any known future data (such as the structure of the mission and which states follow from the currently active state). To this end we are developing optimization schemes for the defender's possible actions such as taking a service off-line when not needed or extending the duration of a state that would be unable to progress if a certain service were compromised.

References

[1] S. Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004 (cited in p. 10)

[2] G. Vigna, *The 2011 UCSB iCTF: Description of the game, http://ictf.cs.ucsb.edu/*, 2011 (cited in p. 1)