Online Appendix: A Stackelberg Game Model for Botnet Data Exfiltration

June 29, 2017

We first provide Lemma 1 showing that the urban security problem presented in (Jain et al. 2011) (see the Related Work section of the main paper) is a special case of the botnet defense problem with uni-exfiltration under some specific conditions. Based on Lemma 1, we then prove the NP-hardness of the attacker and defender oracle problems.

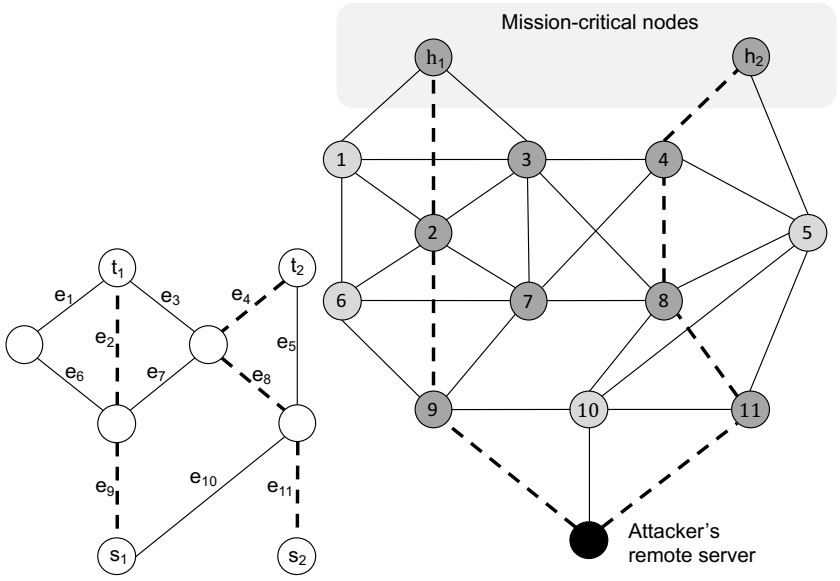
1 Appendix A

Lemma 1. *The urban security problem presented in (Jain et al. 2011) is reducible to the botnet defense problem with respect to data uni-exfiltration conditioned on: (i) the attacker only exfiltrates data from one single mission-critical node; (ii) there is no resource limit for the attacker; and (iii) the defender does not deploy detectors on mission-critical nodes.*

Proof. Let's consider an arbitrary instance of the urban network security problem. There is an urban road network which is represented as a graph $\mathbf{G^u} = (\mathbf{V^u}, \mathbf{E^u})$. The attacker starts at one of the source nodes $s \in \mathbf{S} \subset \mathbf{V^u}$ and travels along a path to attack one of the targets $t \in \mathbf{T^u} \subset \mathbf{V^u}$. The attacker's pure strategies are all possible paths in the graph, each starts from a source $s \in \mathbf{S^u}$ and ends at a target $t \in \mathbf{T^u}$. On the other hand, the defender attempts to protect the targets by placing limited security resources on edges of the graph. The defender's pure strategies are thus all possible allocations of these resources to the edges.

We now construct the corresponding instance $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ of the botnet defense problem as follows. Essentially, the computer network graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is an intersection graph of the edge set $\mathbf{E^u}$. In particular, for each edge $e \in \mathbf{E^u}$ of the urban graph, we create a new node $v^e \in \mathbf{V}$ in the computer graph. Furthermore, for each pair of nodes $v^e, v^{e'} \in \mathbf{V}$ in the computer graph, if the corresponding edges $e, e' \in \mathbf{E^u}$ share a same vertex, we create a new edge $(v^e, v^{e'}) \in \mathbf{E}$ in the computer graph.

For each target $t \in \mathbf{T^u}$, we add a new mission-critical node h_t to the vertex set $\mathbf{V}$. The data value associated with this mission-critical node is equal to the value of target t in the urban graph. For each target $t \in \mathbf{T^u}$ and for each edge $e \in \mathbf{E^u}$ that contains t,



(a) An instance of urban security (b) The corresponding instance of botnet defense

Figure 1: In the instance of urban security, $\{t_1, t_2\}$ are the two targets while $\{s_1, s_2\}$ are the source nodes. In addition, there are other four intermediate nodes and 11 edges denoted by $\{e_1, e_2, \ldots, e_{11}\}$ in the urban graph. In the corresponding instance of botnet defense, there are 11 nodes with respect to to the 11 edges in the urban graph. The connectivity of the nodes in the computer graph is determined based on the connectivity of the edges in urban graph. For example, there is an edge between nodes 1 and 2 since the edges e_1 and e_2 connect via t_1. The two mission-critical nodes h_1 and h_2 correspond to the two targets t_1 and t_2. In addition, the mission-critical node h_1 connects to the nodes $\{1, 2, 3\}$ since target t_1 belongs to edges $\{e_1, e_2, e_3\}$. The attacker path (e_9, e_2) from the source s_1 to the target t_1 in the urban graph is equivalent to the exfiltration path $(h_1, 2, 9, S^a)$ in the computer graph. Conversely, the exfiltration path $(h_2, 4, 8, 11, S^a)$ leads to the path (e_4, e_8, e_{11}).

we create a corresponding new edge that connect the mission-critical node h_t to node $v^e \in \mathbf{V}$. In total, there are $|\mathbf{T^u}|$ corresponding mission-critical nodes in $\mathbf{V^c}$. Finally, we create a remote server of the attacker S^a. For each source $s \in \mathbf{S^u}$ and for each edge $e \in \mathbf{E^u}$ that contains s, we create a new edge which connects S^a with $v^e \in \mathbf{V}$. An example of our computer graph construction is in Figure 1.

We are going to show that finding an SSE of the urban security game is equivalent to finding an SSE in the corresponding instance of the botnet defense game. According to the graph conversion, an edge in the urban graph is a vertex in the computer graph. Therefore, an allocation of the defender's resources on edges in the urban graph is equivalent to an allocation of the defender's resources on nodes in the computer graph. In other words, each pure strategy of the defender in the urban graph is equivalent to a

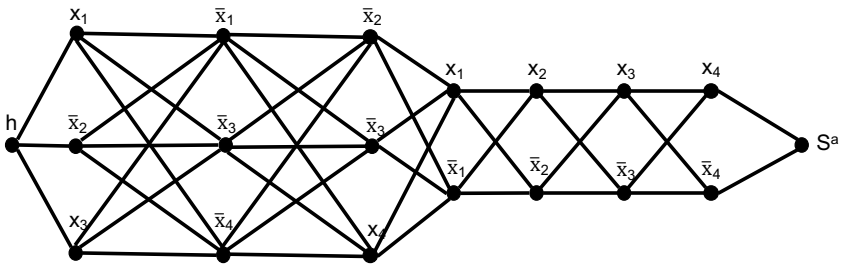


Figure 2: The 3-SAT instance: $(x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (\bar{x}_2 \vee \bar{x}_3 \vee x_4)$ with three clauses and four variables (x_1, x_2, x_3, x_4). In the corresponding computer network, there is a mission-critical node h. An attack path for the attacker can start from the mission-critical node h going through edges on the graph to the server S^a, with the maximum number of compromised nodes is $4 + 3 + 1 = 8$. The defender has eight pure strategies; each includes all nodes corresponding to a literal (e.g., protecting two nodes w.r.t. x_1 or three nodes w.r.t. $\bar{x}_3$) and is played with probability $\frac{1}{8}$.

pure strategy of the defender in the computer graph.

In the instance of the botnet defense game, since the attacker aims at exfiltrating data from one single mission-critical node only without resource-limited constraint, each pure strategy of the attacker can be represented as a path from a mission-critical node to the attacker server S^a to exfiltrate data. We are going to show that each exfiltration path in the computer graph corresponds to an attack path in the urban graph and vice versa. First, any path from a source s to a target t of the attacker in the urban graph is an ordered set of edges $\{e_1, e_2, \ldots, e_K\}$ where $s \in e_1$ and $t \in e_K$ such that each pair of consecutive edges $\{e_k, e_{k+1}\}$ shares a vertex. Thus, this path corresponds to an exfiltration path from the mission-critical node h_1^t to the attacker's server, which is a set of ordered vertices $\{h_1^t, v^{e_K}, \ldots, v^{e_2}, v^{e_1}, S^a\}$. Conversely, considering an exfiltration path from a mission-critical node h_t to S^a, which consists of ordered vertices $\{h_t, v^{e_K}, \ldots, v^{e_2}, v^{e_1}, S^a\}$, we obtain a set of edges $\{e_1, e_2, \ldots, e_K\}$ where there are a source $s \in e_1$ and a target $t \in e_K$ in the urban graph. Since these edges are connected (each pair of consecutive edges shares a vertex), there exists a path from the source s to the target t over these edges, which is a pure strategy of the attacker in the urban security game.

Therefore, the payoff matrix of these two security games is equivalent. Or, finding an SSE of the urban security game is equivalent to finding an SSE in the corresponding instance of the botnet defense game. □

2 Appendix B: Proposition 1

Proposition 1. *The attacker oracle problem corresponding to data uni-exfiltration is NP-hard.*

Proof. Based on Lemma 1, we adapt the NP-hardness proof in (Jain et al. 2011) for

proving the NP-hardness of the attacker oracle problem with data uni-exfiltration. In particular, we show that any instance of the 3-SAT problem can be reducible to an instance of the attacker oracle problem. Let's consider an arbitrary instance of the 3-SAT problem having n variables $\{x_i\}$ for $i = 1, \ldots, n$ and k clauses. Each clause is a disjunction of three literals in which each literal is either a variable or the negation of that variable.

(Jain et al. 2011) construct a corresponding instance of the attacker oracle in the urban security problem for each particular instance of the 3-SAT problem. Based on Lemma 1, we can also construct a corresponding instance of the attacker oracle problem in the botnet defense problem in a similar way. In particular, we construct a computer network as follows: there are $n + k$ layers in the network. The first k layers correspond to k clauses of the 3-SAT problem; each layer has three nodes — each node refers to a literal of the corresponding layer. The next n layers correspond to n variables of the 3-SAT problem; each layer has two nodes referring to a variable and its negation. For the connectivity between consecutive layers, all the nodes in the consecutive layers will connect with each others. In addition to the $n + k$ layers, there is a mission-critical node which contains sensitive data. This mission-critical node connects with all the nodes in the first layer. Finally, there is an edge from all the nodes of the last layers to the attacker's server S^a. An example of the constructed computer network is in Figure 2.

The defender's mixed strategy $(\mathbf{D}, \mathbf{x}^*)$ is defined as follows: there are $2n$ pure strategies in $\mathbf{D}$; each pure strategy corresponds to a literal and the nodes that corresponding to that literal in the network, i.e., these nodes are monitored by the defender's pure strategy. In addition, each pure strategy is played with equal probability of $x_i = \frac{1}{2n}$. In addition, the attacker can compromised $n + k + 1$ nodes at most. We can show that there is an assignment of values to variables in the 3-SAT instance so that the formula is true iff there is an exfiltration path from the mission-critical node to S^a in the corresponding attacker oracle instance which is blocked with probability at most $\frac{1}{2}$, which is similar to (Jain et al. 2011). □

3 Appendix C: Proposition 2

Proposition 2. *The defender oracle problem corresponding to data uni-exfiltration is NP-hard.*

Proof. Similar to (Jain et al. 2011), we also show that any instance of the set-cover problem can be reducible to an instance of the defender oracle problem. In particular, in the Set-Cover problem, there is a set $\mathbf{U}$, a collection $\mathbf{S}$ of subsets of $\mathbf{U}$, i.e., $\mathbf{S} \subseteq \mathbf{U}$ such that $\mathbf{S}$ includes all singleton subsets of $\mathbf{U}$, and an integer k. The question is whether there exists a sub-family $\mathbf{C} \subseteq \mathbf{S}$ of size k such that $\bigcup_{\mathbf{c} \in \mathbf{C}} \mathbf{c} = \mathbf{U}$.

We construct the corresponding computer network instance $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ as follows: there are $|\mathbf{U}|$ mission-critical nodes $h_1, h_2, \ldots, h_{|\mathbf{U}|}$ which contain sensitive data. These $|\mathbf{U}|$ mission-critical nodes correspond to elements in $\mathbf{U}$. In addition, there are other $|\mathbf{U}|$ nodes $(u''_1, u''_2, ..., u''_{|\mathbf{U}|})$ corresponding to elements in $\mathbf{U}$ that directly connect to the attacker's server S^a through $|\mathbf{U}|$ edges. For each non-singleton subset

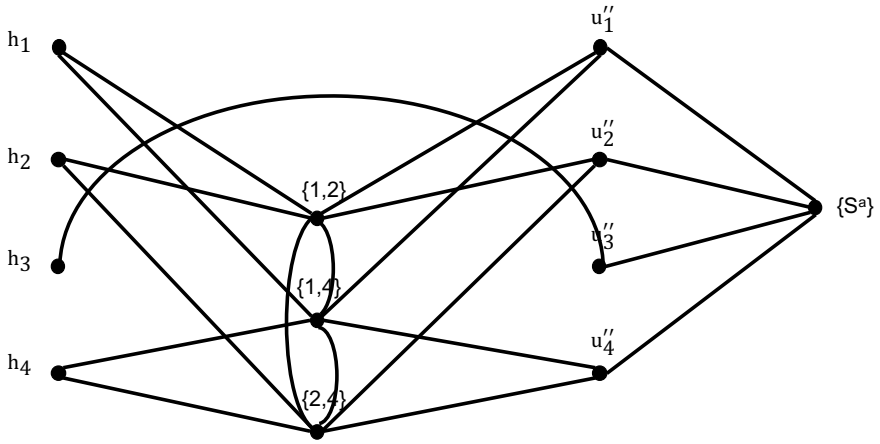


Figure 3: The set $\mathbf{U} = \{1, 2, 3, 4\}$ and the collection of subsets $\mathbf{S} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 4\}, \{2, 4\}\}$. The corresponding computer network is constructed with four mission critical nodes (h_1, h_2, h_3, h_4). There are three nodes corresponding to the three non-singleton subsets $\{1, 2\}, \{1, 4\}, \{2, 4\}$. In addition, four pure strategies of the attacker include $(h_1 \to \{1, 2\} \to \{1, 4\} \to u_1'' \to S^a)$, $(h_2 \to \{1, 2\} \to \{2, 4\} \to u_2'' \to S^a)$, $(h_3 \to u_3'' \to S^a)$, and $(h_4 \to \{1, 4\} \to \{2, 4\} \to u_4'' \to S^a)$. It is clearly that one of the minimum set covering for $\mathbf{U}$ is $\{\{1\}, \{3\}, \{2, 4\}\}$. Equivalently, the defender can also put 3 detectors on nodes $\{h_1\}$, $\{h_3\}$, and $\{2, 4\}$ to block the attacker's exfiltration.

in $\mathbf{S}$, there is a corresponding node in the network. There are edges from nodes of non-singleton subsets of $\mathbf{S}$ to nodes in $(u''_1, u''_2, ..., u''_{|\mathbf{U}|})$ if these subsets contain those nodes. Furthermore, there are also edges from each mission-critical node h_i to nodes of non-singleton subsets in $\mathbf{S}$ containing this mission-critical node 's corresponding element in $\mathbf{U}$. In addition, there are edges between subsets in $\mathbf{S}$ of which intersection is not an empty set. Finally, there is an edge between h_i and u''_i if there is no non-singleton subset in $\mathbf{S}$ containing the corresponding element in $\mathbf{C}$ of these nodes. Figure 3 show an example of the computer network.

The current strategy of the attacker $(\mathbf{A}, \mathbf{a}^*)$ is defined as follows: there are totally $|\mathbf{U}|$ pure strategies of the attacker in $\mathbf{A}$. Each pure strategy consists of an exfiltration path of the attacker and a set of compromised nodes which are the nodes on the exfiltration path. In particular, each exfiltration path starts from each mission-critical node h_i, going through all nodes of non-singleton subsets in $\mathbf{S}$ containing the corresponding element of $\mathbf{U}$, and then ending at S^a. These pure strategies of the attacker are played with equal probability of $\frac{1}{|\mathbf{U}|}$. We show that $\mathbf{U}$ is covered by k subsets in $\mathbf{S}$ if and only if the defender can block all the attacker's exfiltration paths with k resources in the corresponding instance of the defender oracle problem, which is similar to (Jain et al. 2011). □

4 Appendix D: Proposition 3

Proposition 3. *The attacker oracle problem corresponding to data broad-exfiltration is NP-hard.*

Proof. We extend the proof of Proposition 1 to prove the NP-hardness of the attacker oracle problem with broad-exfiltration as follows. In the computer network graph constructed in Proposition 1, we add a new dummy node w and edges connecting w to all other nodes in the network. The network controller uses the following routing algorithm to route data within the network: for any pair of nodes (u, v), if these two nodes do not have a direct connection (i.e., there is no edge connecting them), the routing path from u to v is $\mathbf{P}(u, v) = u \to w \to v$. Otherwise, the routing path is $\mathbf{P}(u, v) = u \to v$. In addition, in the instance of attacker oracle problem with broad-exfiltration, the number of attacker resources is equal to the number of layers in the computer network plus one, i.e., $K^a = n + k + 1$. The defender's mixed strategy $(\mathbf{D}, \mathbf{x}^*)$ is defined similarly as in Proposition 1 but with an addition detector on the dummy node w in each pure strategy of the defender.

Since the attacker can only compromise $n + k + 1$ nodes at most, the attacker has to compromise exactly one node at each layer of the network in addition to the mission-critical node h. Otherwise, all exfiltration paths of the attacker will always go through the dummy node w, which means that the attacker always gets a utility of zero. Thus, even though the attacker can broadcast the stolen data, the only exfiltration path that matters consist of all the compromised nodes ordered according to the layers these nodes belong to. As a result, we obtain an instance of the attacker oracle problem with broad-exfiltration which is equivalent to the instance with uni-exfiltration in Proposition 1. Therefore, the attacker oracle problem with broad-exfiltration is NP-hard. □

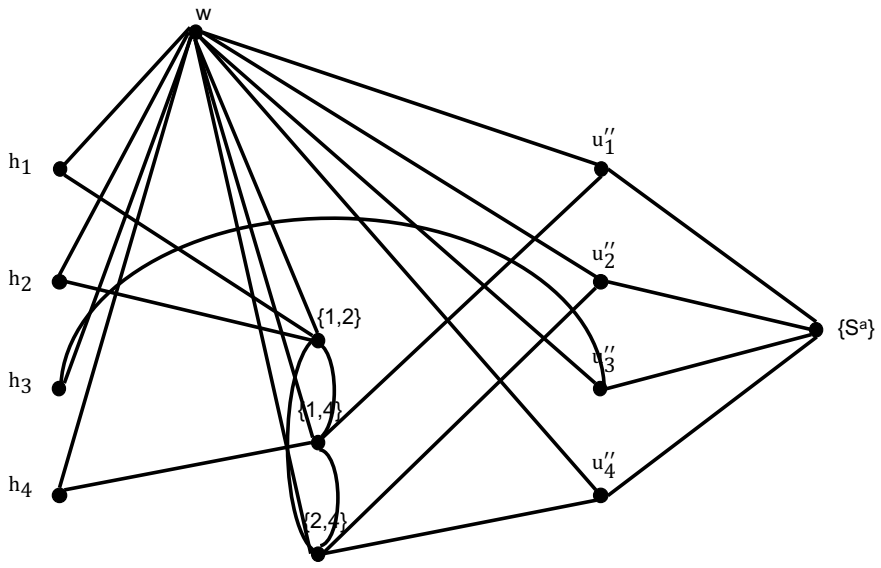


Figure 4: The set $\mathbf{U} = \{1, 2, 3, 4\}$ and the collection of subsets $\mathbf{S} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 4\}, \{2, 4\}\}$. The corresponding computer network is constructed with four mission critical nodes (h_1, h_2, h_3, h_4). There are three nodes corresponding to the three non-singleton subsets $\{1, 2\}, \{1, 4\}, \{2, 4\}$. The lexicographical order of these three nodes is $\{1, 2\} < \{1, 4\} < \{2, 4\}$. In addition, four pure strategies of the attacker are the four sets of compromised nodes: $(h_1, \{1, 2\}, \{1, 4\}, u_1'', S^a)$, $(h_2, \{1, 2\}, \{2, 4\}, u_2'', S^a)$, (h_3, u_3'', S^a), and $(h_4, \{1, 4\}, \{2, 4\}, u_4'', S^a)$. For each pure strategy, for example $(h_1, \{1, 2\}, \{1, 4\}, u_1'', S^a)$, although there are multiple exfiltration paths corresponding to that strategy, the only exfiltration path that matters is $(\{1\} \to \{1, 2\} \to \{1, 4\} \to u_1'' \to S^a)$ since all other exfiltration paths such as $(h_i \to \{1, 2\} \to w \to u_1'' \to S^a)$ need to go through the dummy monitored node w. It is clearly that one of the minimum set covering for $\mathbf{U}$ is $\{\{1\}, \{3\}, \{2, 4\}\}$. Equivalently, the defender can also put 3 detectors on nodes $\{h_1\}$, $\{h_3\}$, and $\{2, 4\}$ to block the attacker's exfiltration.

5 Appendix E: Proposition 4

Proposition 4. *The defender oracle problem corresponding to data broad-exfiltration is NP-hard.*

Proof. We extend the proof of Proposition 2 to prove the NP-hardness of the defender oracle problem with broad-exfiltration. In particular, in constructing a corresponding computer network graph instance $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ for an instance of the Set-Cover problem, the set of nodes in $\mathbf{G}$ is created as the same as in Proposition 2. Nevertheless, unlike Proposition 2, we create a set of edges $\mathbf{E}$ such that for each mission-critical node h_i, there is only a single simple path starting from h_i, going through all non-singleton nodes containing the corresponding element of h_i in $\mathbf{U}$, passing u_i'' and ending at S^a.

In order to do so, we put all the non-singleton nodes in the lexicographical order. We iterate over mission-critical nodes $\{h_i\}$. For each h_i, we create an edge from h_i to the first non-singleton node in the lexicographical order which contains the corresponding element of h_i. We also create an edge from the last non-singleton node in the lexicographical order which contains the corresponding element of u_i'' to u_i''. Furthermore, we create edges connecting consecutive non-singleton nodes in the lexicographical order which has the element of $\mathbf{U}$ corresponding to (h_i, u_i''). Finally, there is an edge between h_i and u_i'' if no non-singleton subset in $\mathbf{S}$ contains the corresponding element of these nodes.

We add a new dummy node w and edges connecting w with all other nodes in the network. The routing algorithm is determined the same as in Proposition 3. Figure 4 shows an example of the constructed computer network.

The current strategy of the attacker $(\mathbf{A}, \mathbf{a}^*)$ is defined as follows: there are totally $|\mathbf{U}|$ pure strategies of the attacker in $|\mathbf{A}|$, each corresponds to an element in $\mathbf{U}$. Each pure strategy consists of all the nodes in the graph corresponding to an element of $\mathbf{U}$ and the non-singleton subset of $\mathbf{S}$ containing that element. Now even though the attacker can broadcast the stolen data, the only exfiltration path in each pure strategy corresponding to the element $i \in \mathbf{U}$ which matters is to start from the mission-critical node h_i, going through all nodes in the lexicographical order which correspond to non-singleton subsets in $\mathbf{S}$ containing i, passing u_i'' and ending at S^a.

As a result, we obtain an instance of the defender oracle problem with broad-exfiltration which is equivalent to the defender oracle instance with uni-exfiltration in Proposition 2. Therefore, the defender oracle problem with broad-exfiltration is NP-hard. □