IoT Threat Modeling Framework: Computing Inherent Risk

Introduction

The following formulation is proposed for calculating measures of risk in the IoT threat modeling framework. The formulation uses the IoT attack taxonomy and answers to the questionnaire as inputs and outputs a measure of inherent risk in each one of the risk domains. In order for this framework to output a useful measure of risk, each element in dimensions 1, 2, 3, and 4 of the IoT attack taxonomy has a score assigned to it.

IoT Attack Taxonomy Score Assignments

- In dimension 1, asset likelihood scores $p(s_i|t_j)$ are assigned to each attacker asset s_i proportional to how likely it is for the attacker to possess that asset given that the attacker is a specific threat actor t_j. In short, each asset likelihood score represents the probability of a particular threat actor possessing or having access to a specific asset.
- In dimension 2, action likelihood scores $p(a_i|t_j)$ are assigned to each attacker action a_i proportional to how easy it is for the attacker to carry out that action given that the attacker is a specific threat actor t_j. In essence, each action likelihood score represents the probability of a particular threat actor carrying out a specific action.
- In dimension 3, vulnerability prevalency scores $p(v_i)$ are assigned to each vulnerability v_i representing how prevalent a particular vulnerability is within the IoT environment. This information can be gathered from vulnerability scanning, penetration testing, and client questionnaires. To make vulnerability score assignment easier, one may simply choose all of the device types relevant to the client and use the devices to vulnerabilities mapping as opposed to assigning vulnerability prevalency scores to every possible vulnerability.
- In dimension 4, impact scores w_p^i are assigned to each compromised property p_i in such a way that they are proportional to the impact of the property being compromised within the IoT environment.
- Each score is assigned within a specific range of values between w_{min} and w_{max}, where w_{min} represents the minimum score assignment for the dimension and w_{max} represents the maximum score assignment for the dimension. Before any risk calculations occur, each score from dimensions 1, 2, and 3 is scaled to take on a value between 0 and 1 according to $w_{scaled} = (w - w_{min})/(w_{max} - w_{min})$, where w denotes the unscaled score and w_{scaled} denotes the scaled score.

Variable Definitions

t_i represents threat actor i, and s_i^j represents attacker asset i from dimension 1 and sub-dimension j of the IoT attack taxonomy. An attacker asset vector $\vec{s}_i$ is composed of 6 attacker asset elements s_i^j, one from each sub-dimension of dimension 1. a_i denotes attacker action i from dimension 2, v_i denotes vulnerability i from dimension 3, and p_i represents compromised property i from dimension 4. Note that $\vec{s}_i$ is a vector of elements whereas s_i, a_i, v_i, and p_i each represent one element from the taxonomy.

Calculating a Measure of Likelihood for Each Vulnerability

To calculate the likelihood of a specific vulnerability being exploited, we consider the appropriate combinations of threat actors, attacker assets, and attacker actions that can be used to exploit the particular vulnerability. The likelihood of the union of all possible combinations of threat actors, attacker assets, and attacker actions is used as the measure of likelihood for a particular vulnerability being exploited. In order to simplify the computation, independence is assumed between all threat actors, attacker assets, and attacker actions associated with the particular vulnerability. We start by calculating the likelihood of a specific threat actor possessing at least one of the attacker assets in sub-dimension j of dimension 1 that can be used to carry out action k. Using the inclusion-exclusion principle and the assumed independence between attacker assets, this likelihood is given in the equation below where S_k^j represents the set of assets from sub-dimension j of dimension 1 that can be used to carry out action k, N_k^j represents the number of elements in this set, and $|I|$ represents the cardinality of set I.

$$p\left(\bigcup_{s_i^j \in S_k^j} s_i^j \,\middle|\, t_n\right) = \sum_{\substack{i=1 \\ s_i^j \in S_k^j}}^{N_k^j} \left((-1)^{i-1} \sum_{\substack{I \subset \{1,\ldots,N_k^j\} \\ |I|=i}} \prod_{m \in I} p(s_m^j|t_n) \right) \tag{1}$$

We then calculate the likelihood of a specific threat actor possessing at least one vector of assets that can be used to carry out action k. Using the assumed independence between attacker assets, this likelihood is given in the equation below where S_k represents the set of asset vectors that can be used to carry out action k.

$$p\left(\bigcup_{\vec{s}_q \in S_k} \vec{s}_q \,\middle|\, t_n\right) = \prod_{j=1}^{6} p\left(\bigcup_{s_i^j \in S_k^j} s_i^j \,\middle|\, t_n\right) \tag{2}$$

Next, we calculate the likelihood of a specific vulnerability being exploited by a particular threat actor. Using the inclusion-exclusion principle and the assumed independence between attacker assets and actions, we consider all possible combinations of asset vectors and actions that could be used to exploit the vulnerability. This likelihood is given in the equation below where A_r represents the set of actions that can be used to exploit vulnerability r and N_r represents the number of elements in this set.

$$p(v_r|t_n) = p\left(\bigcup_{a_k \in A_r} \left(a_k, \bigcup_{\vec{s}_q \in S_k} \vec{s}_q\right) \,\middle|\, t_n\right) = \sum_{\substack{k=1 \\ a_k \in A_r}}^{N_r} \left((-1)^{k-1} \sum_{\substack{I \subset \{1,\ldots,N_r\} \\ |I|=k}} \prod_{m \in I} p(a_m|t_n) p\left(\bigcup_{\vec{s}_q \in S_m} \vec{s}_q \,\middle|\, t_n\right)\right) \tag{3}$$

Lastly, we calculate the likelihood L_r of a specific vulnerability being exploited by at least one of the relevant threat actors. Using the inclusion-exclusion principle and the assumed independence between threat actors, this likelihood is given in the equation below where T represents the relevant set of threat actors and N represents the number of elements in this set.

$$L_r = p\left(v_r \,\middle|\, \bigcup_{t_n \in T} t_n\right) = \sum_{\substack{n=1 \\ t_n \in T}}^{N} \left((-1)^{n-1} \sum_{\substack{I \subset \{1,\ldots,N\} \\ |I|=n}} \prod_{m \in I} p(v_r|t_m)\right) \tag{4}$$

Calculating a Measure of Impact for Each Vulnerability

To calculate the impact associated with a particular vulnerability being exploited, we consider all the properties that are compromised when that vulnerability is exploited by adding together the associated impact scores. The impact I_i of exploiting vulnerability i is given in the equation below where P_i represents the set of properties that are compromised when vulnerability i is exploited.

$$I_i = \sum_{p_j \in P_i} w_p^j \tag{5}$$

Calculating Inherent Risk for Each Risk Domain

To calculate the inherent risk associated with a particular vulnerability, we multiply the likelihood of exploiting the vulnerability times the impact associated with exploiting the vulnerability. When calculating the inherent risk R_i for risk domain i, we carry out a weighted sum of the inherent risks associated with each vulnerability in risk domain i where the weights are the vulnerability prevalency scores. This computation is given in the equation below where V_i represents the set of vulnerabilities associated with risk domain i.

$$R_i = \sum_{v_j \in V_i} p(v_j) L_j I_j \tag{6}$$

To normalize the inherent risk so that all measures of inherent risk lie between a minimum value r_{min} and a maximum value r_{max}, we scale the inherent risk by the maximum possible inherent risk. The maximum possible inherent risk occurs when all vulnerabilities are considered, the likelihood of exploiting any vulnerability is 100%, and the impact score for every vulnerability is the maximum possible impact score. The normalized inherent risk R_i^{norm} for risk domain i is given in the equation below where V represents the set of all possible vulnerabilities and I_j^{max} represents the maximum possible impact score for vulnerability j.

$$R_i^{norm} = \frac{R_i}{\sum_{v_j \in V} p(v_j) I_j^{max}} (r_{max} - r_{min}) + r_{min} \tag{7}$$

IoT Maturity Assessment: Computing Residual Risk

Introduction

The IoT maturity assessment gives an understanding of how well a client has implemented risk-mitigating controls, allowing a measure of residual risk to be calculated. Each control c_i has an associated control implementation score $p(c_i)$ and a control effectiveness score $p(e_i)$ that represent how well a client has implemented that particular control and how effective that particular control is in mitigating risk, respectively. These control scores can be assigned using answers to the questionnaire. Each score is assigned within a specific range of values between w_{min} and w_{max}, where w_{min} represents the minimum score assignment for the dimension and w_{max} represents the maximum score assignment for the dimension. Before any maturity assessment calculations occur, each score is scaled to take on a value between 0 and 1 according to $w_{scaled} = (w - w_{min})/(w_{max} - w_{min})$, where w denotes the unscaled score and w_{scaled} denotes the scaled score.

Calculating a Maturity Score for Each Vulnerability

To calculate a maturity score for a specific vulnerability, we first calculate a mitigation percentage for each control where the mitigation percentage equals how well the control is implemented times how effective the control is. The mitigation percentage $p(m_i)$ for control i is given in the equation below and represents how well control i mitigates risk.

$$p(m_i) = p(c_i)p(e_i) \tag{8}$$

Using the inclusion-exclusion principle and assuming independence between controls, the maturity score M_j for vulnerability j can be calculated as shown in the equation below where C_j represents the set of controls associated with vulnerability j and N_j represents the number of elements in this set.

$$M_j = p\left(\bigcup_{m_i \in C_j} m_i\right) = \sum_{\substack{i=1 \\ m_i \in C_j}}^{N_j} \left((-1)^{i-1} \sum_{\substack{I \subset \{1,\ldots,N_j\} \\ |I|=i}} \prod_{k \in I} p(m_k) \right) \tag{9}$$

Calculating Residual Risk for Each Risk Domain

To calculate the residual risk associated with vulnerability j, we multiply the inherent risk $L_j I_j$ associated with vulnerability j times the unmitigated risk percentage $(1 - M_j)$. When calculating the residual risk Z_i for risk domain i, we carry out a weighted sum of the residual risk associated with each vulnerability in risk domain i where the weights are the vulnerability prevalency scores. This computation is given in the equation below where V_i represents the set of vulnerabilities associated with risk domain i.

$$Z_i = \sum_{v_j \in V_i} p(v_j) L_j I_j (1 - M_j) \tag{10}$$

The residual risk can be normalized in the same way as the inherent risk as shown in the equation below where Z_i^{norm} represents the normalized residual risk for risk domain i.

$$Z_i^{norm} = \frac{Z_i}{\sum_{v_j \in V} p(v_j) I_j^{max}} (r_{max} - r_{min}) + r_{min} \tag{11}$$

Sensitivity Analysis

While the measures of inherent and residual risk in each risk domain are helpful metrics for the client, a sensitivity analysis provides more detailed information about which specific controls should be better implemented to further reduce residual risk. A one-at-a-time sensitivity analysis is conducted where a small value $\Delta p(c_j)$ is added to the control score $p(c_j)$ for control j, representing a small improvement in the implementation of control j. The normalized residual risk Z_i^{norm} is recomputed as described above, and a sensitivity score ΔZ_{ij}^{norm} for risk domain i is assigned to control j. The sensitivity score ΔZ_{ij}^{norm} is simply the difference between the original residual risk and the recomputed residual risk. This process is repeated for each control j in each risk domain i until all sensitivity scores ΔZ_{ij}^{norm} have been assigned. The sensitivity scores in risk domain i can then be ordered in decreasing magnitude to show which controls have the largest effect on further reducing residual risk.