Attribute-based low-complexity network access control policy with optimal grouping algorithm

Shohei Mitani1(a), Taniya Singh1, Nakul Ghate1, and Hirofumi Ueda1
1 Secure System Research Laboratories, NEC Corporation
1753 Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa 211-8666, Japan
a) s.mitani@nec.com

Abstract: Zero Trust (ZT) Model [1] is a security approach to prevent malicious activities assuming the presence of an attacker in the environment. The fine-grained access control should be executed on ZT in accordance with various information, which requires a large complexity of access control policy due to the large patterns of attributes [2]. Our focus is the low-complexity of policy management. We propose a method to reduce and evaluate the complexity of policies for network access control. This letter discloses the optimal grouping algorithm to reduce the complexity, and shows the higher performance in comparison with the existing methods.

Keywords: network access control, policy, complexity, clustering, attribute-based access control

Classification: Network management/operation

References

[1] S. Rose, O. Borchert, S. Mitchel, and S.Connelly, Zero Trust Architecture. NIST Special Publication 800-207, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2020. DOI:10.6028/NIST.SP.800-207

[2] M. W. Sanders and C. Yue, “Mining least privilege attribute based access control policies,” ACSAC ’19: Proceedings of the 35th Annual Computer Security Applications Conference, San Juan, USA, pp. 404–416, Dec. 2019 DOI:10.1145/3359789.3359805

[3] R. Vanickis, P. Jacob, S.Dehghanzadeh, and B. Lee, “Access control policy enforcement for zero-trust-networking,” 2018 29th Irish Signals and Systems Conference (ISSC), Belfast, UK, 6 pages, Dec. 2018 DOI: 10.1109/ISSC.2018.8585365

[4] L. Karimi and J. Joshi, “An unsupervised learning based approach for mining attribute based access control policies,” 2018 IEEE International Conference on Big Data (Big Data), Seattle, USA, 10 pages, Dec. 2018 DOI:10.1109/ISSC.2018.8585365

[5] F. Isik, “An entropy-based approach for measuring complexity in supply chains,” International Journal of Production Research, vol. 48, no. 12, pp. 3681–3696, Jun. 2009. DOI:10.1080/00207540902810593
1 Introduction

In an enterprise network, a variety of devices, users, and resources are connected at anytime from anywhere. There is always a possibility of presence of attackers in the internal network. To deal with it, Zero Trust (ZT) model [1] evaluates all access requests from devices, and executes access control with fine-granularity. An important functionality on ZT is the policy definition. Network operators have to configure the rule sets to define how access control should be performed. Attribute-based access control (ABAC) policy is a common choice to implement the fine-grained access control, such as ZT. A detailed ABAC policy definition is very costly because the privilege space becomes too large to deal with as the attributes’ pattern increases. A solution to reduce the policy definition cost is to systematically generate the policy by rule-mining with audit logs [2]. Logic programming or natural language processing-based rule extractions are also known.

However, such methods require log data, declaration of constraints without excess or deficiency, enough to execute the expected access controls. Insufficient assumption will lead to undesirable policy decisions. The decrease of initial policy definition cost will increase the management cost to verify or modify the generated policy to apply it to the network.

We propose a method to reduce the policy management cost by reducing the complexity of the policy, targeting the network access control (NAC) policy. We assume that a fine-grained policy is given by systematic policy generation methods, one of which we introduce in this letter to improve fineness without detailed assumptions. Our main proposal reduces the complexity without affecting the fineness of the given generated policy. We disclose the grouping-based optimization algorithm and the evaluation result.

2 Proposed method

2.1 Overview of access control model

In this section, we briefly explain the overview of our access control model, the policy generation method to reduce policy definition cost, and the complexity reduction method to reduce the management cost of the generated policy.

Network access control model Our target is the network access control (NAC) policy because of its feasibility. It’s defined with IP address, port number, and any additional variables on network or application layer. We can easily deploy it with policy enforcement points (PEPs) as gateways. Our model has the policy generator (PG) which performs the systematic policy generation. For feasibility, the PG also updates the NAC policy in response to changes of information sources such as asset database and activity logs [3]. Our method reduces the complexity of the policy generated by PG at each time by grouping the nodes, e.g. IP address, as shown in Fig. 1. The figure also shows Access control list (ACL) policy and the grouped policy.
Policy generation method  We briefly describe the simple PG mechanism by modeling the network operators’ intents for partially delegating the policy definition process to the system. Our model expresses the positive, negative effects and the trade-off between them. All values of attributes should be represented as scores to indicate the positive effects to the policy decision. The score of location of user = “home” will be lower than “office”. Combinations of attributes have scores to express access rights. The policy is the result of fine evaluation of trade-off, and is updated dynamically. It’s difficult to manage, verifying and modifying due to the fineness.

Complexity reduction method  We provide an abstract of our complexity reduction method. The previous study [4] had proposed the ABAC policy extraction from ACL tuple using k-means clustering algorithm, achieving both low complexity and fine-granularity i.e. reproduction rate of the original policy. They quantified the complexity with the number of rules, and grouped the similar rules based on attributes. We can group nodes or tuples based on their attributes as well. However, this is not the optimal in NAC. It has the difficulty of parameter tuning and additional pruning of the same rules. These are caused by the conflicts of two independent objectives, the squared error on attributes, and the policy quality metric.

Our contribution is twofold. First, we disclose an algorithm to achieve a NAC policy with the lowest complexity and the finest granularity. We propose a distance metric instead of the squared error. Second, we quantify the entropy-based complexity to evaluate the policy management cost.

Fig. 1. Overview of the access control policy.
2.2 Complexity reduction algorithm based on optimal grouping

In this section, we disclose an algorithm to minimize the complexity of NAC policy, while keeping the fineness of the policy, using the following definitions.

NAC policy consists of a set of node \( X \), content \( E \) such as port number, and action \( A = \{ \text{"Accept"}, \text{"Deny"} \} \). We assume that a generated policy \( P_G \) which defines actions as scores is given, i.e. \( P_G : X \times X \times E \rightarrow [0, 1] \subset \mathbb{R} \). \( P_G \) is generated based on the relevance between node \( x \) and attributes, e.g. location of the users assigned to that node. The two nodes \( X \times X \) correspond to source and destination, respectively. The meaning of the score is as follows.

The access flow containing \( x, x' \in X \) and \( e \in E \) should be absolutely denied if \( P_G(x, x', e) = 0 \), whereas accepted if \( P_G(x, x', e) = 1 \). When \( P_G(x, x', e) = 0.4 \), the flow should be denied with low confidence. In our proposal, we define a set of group \( C \) and a grouping function \( \zeta : X \rightarrow C \) to reduce the complexity. We describe the set of nodes in \( c \) as \( X_c = \{ x \in X \mid \zeta(x) = c \} \). The cardinalities of \( X, E, C \) are \( n, m, k \), respectively. We describe the policy with fixed nodes \( R_{x \rightarrow x'} \) so that \( R_{x \rightarrow x'}(c) = P_G(x, x', e) \). The policy can be represented by source and destination groups \( c, c' \in C \), such as \( \mathcal{P}_{c \rightarrow c'} : E \rightarrow [0, 1] \). The actual actions are obtained as \( \mathcal{A}_{c \rightarrow c'} : E \rightarrow A \) discretizing \( \mathcal{P}_{c \rightarrow c'}(e) \).

Our algorithm converts a given policy \( R_{x \rightarrow x'} \) into a less complex policy \( \mathcal{P}_{\zeta(x) \rightarrow \zeta(x')} \). It solves the optimization problem to minimize \( k \), while limiting the dissimilarity between \( R_{x \rightarrow x'} \) and \( \mathcal{P}_{\zeta(x) \rightarrow \zeta(x')} \) lower than a threshold \( \epsilon_{\text{max}} \) in two steps. First, we optimally group the nodes \( c = \zeta(x) \). Second, we select the policies \( \mathcal{P}_{c \rightarrow c'} \) for each pair of groups \( c, c' \in C \).

**Optimal grouping** At first, the algorithm calculates \( X_c \) in group \( c \) with \( k \)-medoid clustering. To achieve the optimal solution, the algorithm defines a distance matrix based on the policy \( R_{x \rightarrow x'} \) instead of using the relevant attributes as a feature vector. We define the distance between nodes \( d(x, x') \) to describe how much the whole policy changes if \( x \) and \( x' \) are exchanged.

\[
d(x, x') = \sum_{z \in X \setminus \{x, x'\}} (D(R_{x \rightarrow z}, R_{x' \rightarrow z}) + D(R_{z \rightarrow x}, R_{z \rightarrow x'})) + D(R_{x \rightarrow x'}, R_{x' \rightarrow x}) \tag{1}
\]

where \( D(R_{x \rightarrow y}, R_{x' \rightarrow y'}) = (\sum_{e \in E} (R_{x \rightarrow y}(e) - R_{x' \rightarrow y'}(e))^p)^{1/p} \) describes the dissimilarity between policies measured by \( L^p \)-norm. We used \( p = 1 \). The algorithm evaluates \( L_{\text{global}} = \max_{c \in C} [\max_{x, x'} \in X_c |d(x, x')|] \), the maximum intra-class loss. It finds the smallest \( k \) satisfying \( L_{\text{global}} \leq \epsilon_{\text{max}} \).

**Policy selection** In the rest of Algorithm 1, the policies for each pair of groups \( \mathcal{P}_{c \rightarrow c'}(e) \) are chosen as representative policies. The algorithm averages the original policies defined between groups. \( |X_c| \) is the cardinality of \( X_c \). The policies are discretized, and grouped NAC policy \( \mathcal{A}_{c \rightarrow c'} \) is achieved.

2.3 Quantification of policy management costs with entropy-based complexity

In this section, we propose a quantification of management cost of NAC policy with entropy in a similar way with [5]. Shannon entropy corresponds to the complexity of optimally designed management interface. The entropy of an original policy generated by PG is \( H_{\text{original}} = n(n - 1) \), i.e. \( O(n^2) \).
Algorithm 1 Optimal complexity reduction algorithm

Require: Original policy $R_{x \rightarrow x'}$ for all $x, x' \in X$ and $\epsilon_{\text{max}}$ are given.

Ensure: Optimize groups $X_c$ and grouped policies $P_{c \rightarrow c'}$.

\begin{algorithm}
\begin{algorithmic}
\State \textbf{for} $i = 1, 2, \cdots, n$ \textbf{do}
\State \hspace{1em} \textbf{for} $j = 1, 2, \cdots, n$ \textbf{do}
\State \hspace{2em} $d_{i,j} \leftarrow d(x_i, x_j)$
\State \hspace{1em} \textbf{end for}
\State \textbf{end for}
\State $k \leftarrow 0$
\State $L_{\text{global}} \leftarrow \epsilon_{\text{max}} + 1$
\While{$L_{\text{global}} > \epsilon_{\text{max}}$}
\State $k \leftarrow k + 1$
\State $L_{\text{global}} \leftarrow 0$
\For{$c = 1, 2, \cdots, k$}
\State $X_c \leftarrow$ Group $c$ obtained by $k$-medoids with the distance matrix \{d_{i,j}\}.
\State $L_{\text{global}} \leftarrow \max (L_{\text{global}}, \max_{x_i, x_j \in X_c} [d_{i,j}])$
\EndFor
\EndWhile
\For{$c = 1, 2, \cdots, k$}
\For{$c' = 1, 2, \cdots, k$}
\State $P_{c \rightarrow c'} \leftarrow \frac{1}{|X_c||X_{c'}|} \sum_{x_i \in X_c, x_j \in X_{c'}} [R(x_i, x_j)]$
\EndFor
\EndFor
\end{algorithmic}
\end{algorithm}

We describe two formulae of complexity of grouped $P_G$. The first one is the complexity when the nodes are grouped. The algorithm in the previous section is one example of this kind. The grouping function is $\zeta : X \rightarrow C$. The shape of the policy is “$C \times C \times E \rightarrow A$.” The first term indicates the number of grouped policies. The second term is the complexity to manage the grouping. The symbol \{\} means the Stirling number of the second kind.

\begin{equation}
H_{\text{node}} = k^2 + \frac{1}{m} \log_2 \left\{ \begin{array}{c} n \\ k \end{array} \right\}
\end{equation}

The second one is grouping tuples of nodes as $\zeta : X \times X \rightarrow C$ instead of grouping node itself. The shape of the policy is “$C \times E \rightarrow A$.”

\begin{equation}
H_{\text{tuple}} = k + \frac{1}{m} \log_2 \left\{ \begin{array}{c} n(n-1) \\ k \end{array} \right\}
\end{equation}

For large $n$, the orders are $O(nm^{-1}\log_2 k)$, $O(n^2m^{-1}\log_2 k)$, respectively. $H_{\text{node}}$ is lower than $H_{\text{tuple}}$ for the same number of grouped policies $k^2, k$.

3 Evaluation

3.1 Simulation conditions

To evaluate the performance of the complexity reduction, we set the number of IP address to $n = 30$, the number of port to $m = 10$. We simulated the attributes of users, devices, IP addresses, port numbers, user roles, resources.
In the evaluation scenario, we updated one of the device trust from high value to low. Trust in the whole network was re-evaluated and policies were updated by the PG. We used this as the original policy. We compared our proposed method with the performance of the non-optimal node clustering method like [4] (Attribute-based node shrinkage), and with the other optimal strategy grouping tuples (Policy shrinkage).

### 3.2 Results

Fig. 2 shows the result. We plotted the coarseness $i.e.$ reproduction error rate $\frac{1}{mn^2} \sum_{x,x' \in X} D\left(R_{x \rightarrow x'}, P_{\zeta(x) \rightarrow \zeta(x')}\right)$ while varying the complexity. The complexity is $H_{node}$ (Eq. 2) for Attribute-based node shrinkage and the proposed method, and is $H_{tuple}$ (Eq. 3) for policy shrinkage. Original policy granularity is the coarseness that only a single rule is different to the original policy ($\epsilon_{max} = 1$). Static RBAC (Role-based access control) is for reference which have two pre-defined user roles (groups). This is the coarsest policy.

Our method achieved the same policy as that of the original one (less than a single error) at the complexity 72, corresponding to eight number of groups. It is 89 percent lower than that of the attribute-based node shrinkage, and 81 percent lower than that of policy shrinkage. The proposed method is plotted in the lower-left side of the figure, indicating the best performance.

![Fig. 2. Complexity vs Coarseness of grouped policies.](image)

### 4 Conclusions

This letter proposed the complexity reduction method of the systematically generated fine-grained policy for NAC. Our proposed method groups nodes whereas the objective describes the whole policy. The reason of the best performance is that the policy definition process is implicitly affected by two kinds of information, node specific attributes and other than that, e.g. conditional properties and trade-offs. The practicality of the method depends upon the policy generation, which will also be evaluated as the whole system.