Access Control Flaws: What Are They and How Can They Be Avoided?

(InfoBeyond Technology LLC)

Abstract

Security Policy Tool is a software tool for Access Control Security Managers, Policy Authors, and other IT Security Professionals specializing in the performance of access control systems. Access control policies are designed to protect the accessibility of online resources in networks, IoTs, healthcare systems, financial service systems, enterprise IT and clouds, military systems, and other online environments. There are several challenges in building robust access control models for these systems including (i) effectively composing secure policies and rules, (ii) testing these policies systematically, (iii) verifying these policies to prevent access control leaks. Security Policy Tool solves these issues by providing powerful access control policy modeling, testing, and verification features that empower organizations to close the door to access control leaks.

Index Terms

Access control, attribute-based access control (abac), role-based access control (rbac), security policy editing, test, verification, deployment, access control leaks, XACML, software tool.

1 INTRODUCTION

Access control flaws can be defined as unintended access decisions caused by misconfigured rules, policies, or algorithms within an access control system. Often these flaws are hidden from detection due to access decisions being connected to more than one rule or policy. Identifying and removing these access control flaws is paramount to ensuring network resources are safe from cyber threats. Security Policy Tool facilitates secure access control policy development by providing rich modeling, testing, and verification functions. It enables policy authors to easily edit, manage, and review a number of ABAC, RBAC, MLS, and Workflow rules and policies to verify they are free of access control flaws.

After Modeling and Testing, Security Policy Tool can easily locate all rules, attributes, and inheritance relations engaged with a particular Verification Result (e.g., True or False). This enables the policy author to quickly make changes to rules or policies that are creating vulnerabilities. Security Policy Tool empowers the policy author to find: (i) what is the access control flaw, (ii) where to fix the flaw, (iii) how to correct the flaw, and (iv) what is the access control effectiveness after the modification. This next section describes typical access control flaws that can be resolved by leveraging Security Policy Tool's policy analysis features.

- *Contact us at: E-mail: Info@Securitypolicytool.com*

Security Policy Tool (www.Securitypolicytool.com) is a commercial version of NIST(National Institute of Standards and Technology)'s ACPT (Access Control Policy Tool). With tremendous consultation with NIST experts, Security Policy Tool substantially enhances and expands the NIST's ACPT design with advanced features for achieving high security confidence access control levels such that it can be commercialized. The development of Security Policy Tool is financially sponsored by NIST via a SBIR (Small Business Innovation Research) Phase I and II programs. It specifically improves the NIST's ACPT design to provide a robust, unified, professional, and functionally powerful access control policy tool.

Access Control Flaws	Description
Block Privilege	User should have access but doesn't have access
Leak Privilege	User should not have access but does have access
Not Protected Resource	Defined resource is not protected under any rules
Rule Conflict	Two or more rules, defining opposite authorization
Inconsistent Assignment	Error in attribute labeling during policy creation
Inheritance Loop	User granted recursive and subsequent inheritance
Undecided Rules	Rule is not properly defined or missing a step
Separation of Duty	Access rule creates unintended conflict of interest

TABLE 1: **Typical Access Control Flaws**

2 Typical Access Control Flaws

Error Type 1 (Block Privilege): Suppose you know "'Subject A"' should be able to access "'Resource B"'. However, when Subject A goes to access Resource B the subject is Denied access. This type of error is referred to as a Privilege Block. It blocks legitimate access to rightful resources. It can also occur when the properties of an access control policy cannot render a Permit/Deny decision, or if there is no available logic in the policy algorithm for evaluating the access request. It can also be a result of the deadlock of access rules where a rule has a dependency on other rule(s), which eventually depend back on the rule itself, so that a subjects request will never reach a decision because of the cyclic referencing.

Error Type 2 (Leak Privilege): Suppose you know "'Subject A"' should not be able to access "'Resource B"'. However, when Subject A goes to access Resource B the Subject is Permitted access. This type of error is referred to as a Privilege Leakage. It authorizes wrongful access to prohibited resources. Such leakage may cause either the privilege escalation from one resource domain or class to prohibited ones such as leakage from lower to higher ranks of an MLS policy, or privilege leak such as from one role to other prohibited ones of an RBAC policy. Leak Privilege can be caused by mistaken privilege assignment directly or careless privilege inheritance indirectly.

Error Type 3 (No Protected Resource): Suppose you know "'Resource B"' is available for review in a shared folder on "'Subject A's"' work laptop. However, when Subject A goes to view this network resource the system provides an unexpected decision. Upon further investigation and testing the policy author realizes that there have been no rules defined for Subject A's access to Resource B. This error is referred to as a No Protected Resource flaw. This is an error that a Resource (e.g., B) is not protected by any rules and policies so thus the system will provide a decision based on your enforcement algorithm instead of an actual rule. In this case, the unprotected resource would likely cause an error type of Block Privilege.

Error Type 4 (Rule Conflict): Suppose you know that "'Subject A"' should be able to access "'Resource B"' however the individual is Denied access after attempting to view the resource. Upon further investigation and testing the policy author realizes that within their policy their is both a rule Permitting access to Resource B and Denying access to Resource B. This error is referred to as a Rule Conflict or "'Privilege Conflict"'. It occurs when two or more rules

are allowing access while the other declines. Unlike regular programming logic that a later value assignment of a variable overwrites the previous assigned value of the same variable, the rules of an access control policy typically has no precedence consideration in permission evaluation. In other words, access control rules will not be overwritten by other rules unless specifically allowed to. Thus, privilege conflicts appear when the specifications of two or more access rules result in the conflicting decisions of permitting subjects access requests by either direct or indirect (inherit) access assignments. In addition, when multiple policies are evoked for permission, conflicting decisions between policies may occur.

Error Type 5 (Inconsistent Assignment): Suppose a policy author edits a number of XACML policy documents separately in a text editor. In doing so, the author mistakenly defines attributes, conditions, rule or other policy variables/values inconsistently in different policies. For example, the policy author intended to define a rule for Attribute Nurse but inconsistently termed the attribute as "'Murse"' in different policy documents. This error could go onto create a security vulnerability depending on the resources included in the flawed rule. Security Policy Tool prevents this error by default with its integrity verification and syntax error detection features included in the XACML Editor.

Error Type 6 (Inheritance Loop): Suppose a policy author has intended for "'Subject A"' to inherit all of "'Subject C's"' access privileges. However, the Policy Author has forgotten that in the same policy earlier Subject C has already been defined to inherit all of Subject A's access privileges. By defining both these relationships the author would be creating an Inheritance Loop Error. This is caused by recursive and subsequent privilege inheritance. An access control Inheritance Loop leads to undecidable or infinite access evaluation process. Security Policy Tool is able to automatically detect and prevent inheritance loops.

Error Type 7 (Undecided Rules): Similar to the Not Protected Resource flaw described earlier this error occurs when an access request is made but there is no rule defined instructing the system what decision to make. An Undecided Rule error in a Workflow policy could result from a Resource that is unassigned with the necessary actions to determine a decision. For example, a purchase order is assigned for a person to create however there are no assigned rules to view the order.

Error Type 8 (Separation of Duty Error): This type of flaw is related to being cautious to not giving individuals too much privileges to the point they could misuse the system. For example, Permitting a Loan Officer to create Loans but Denying them to being able to Approve loans in an effort to minimize the potential for Loan Officers to create improper loans.

3 CONCLUSION

Now you should have a better understanding of what to look for as you go onto verify your access control policies with Security Policy Tool. In addition to this document there are other resources located in the Learning Center in your My account page that will help you start leveraging Security Policy Tool to prevent access control leaks, today!

If you have not yet, download Security Policy Tool – Lite Version for FREE now! Close the door the Access Control Leaks and save time and cost creating, modeling, testing, and verifying your access control policies, today.

Click here to begin securing your policies now → Lite Version.

InfoBeyond Technology, LLC is an innovative company specializing in Network, Machine Learning and Data Security within the Information Technology industry. The mission of InfoBeyond is to research, develop, and deliver viable software products for network communication and security. Some of our research is sponsored by Department of Defense, Department of Energy, Missile Defense Agency, Department of Transportation, NIST (National Institute of Standards and Technology), etc. Security Policy Tool is Awarded the 2017 Innovative Security Solution Award at the 2017 Big Data and SDN/NFV Summit. NXdrive is a fragment-based cybersecurity storage system and more information can be found at www.NXdrive.com. The company is featured as one of 50 fast growth IT small businesses in 2017 by The Silicon Review.