In today’s enterprise networks shared and computing resources are becoming very common. Individuals in an organization should have well defined and precise roles. Access control to all resources should be based purely on these roles. The access rights to all shared resources should automatically change when a user changes their role. Users create security misconfigurations when they have access to resources that they shouldn’t have access to. One reason for the high rate of security misconfigurations in organizations is the change of roles in employees. Another reason is because when employees work together across organizational groups, the dynamic nature of information sharing causes some problems. When there’s no complete high-level manifests, it makes it difficult for administrators to manually manage access control. Baaz, is a system that monitors access control metadata of various shared resources across an enterprise. This can address the limitations of existing access control management systems. It uses fast and efficient algorithms to find security and accessibility misconfigurations. Baaz is the first system that helps an administrator audit access to the control mechanisms and discover critical security and accessibility vulnerabilities. It uses group mapping and object clustering algorithms. The operational context and main characteristics of bazz are No assumption of well-defined policy which means that it doesn’t require a high-level policy manifest. Proactive vs Reactive which means that it takes input static permissions such as access control lists, rather than access logs. And timeliness means that baaz continuously monitors access control so it can be configured to detect and report misconfigurations on sensitive data items as they occur.

The group mapping algorithm comprises of elimination of all groups in which more than half of the users are not members. Then when the number of groups is less than 20, we use a modified version of the greedy set-cover algorithm to do the matching. We use this algorithm to find more valid misconfigurations. When baaz presents the misconfiguration report to the administrator, it lists the candidates in a priority order. When the mismatches between a user-set and its covering reference group is smaller, the possibility of the misconfiguration candidate being a valid issue is higher. That is how the ranking function is used for prioritization of both accessibility and security candidates capture this measure of difference in similarity between a user-set and its cover. Another technique for finding misconfiguration candidates is object clustering. Object clustering is used as a fallback in situations where there do not exist suit-able reference groups to flag misconfiguration candidates through group mapping. There are some issues that impact the quality of the misconfiguration reports produced by baaz. One of them is server design issues. An administrator needs to use domain knowledge to choose the right reference dataset for a given subject dataset. f a reference dataset contains a large number of reference groups, such as a set of email distribution lists, the report will contain fewer candidates because the chances of finding exact covers increases. The algorithm can miss some misconfigurations as a result. Another is modeling access control. This is because access control mechanisms can be complicated and its hard to capture complete semantics in a stub. Another is stub customization. Stubs may need to be customized to different data layouts containing group membership data. Access mechanisms of different kinds of resources will require custom stub implementations that can specifically understand the underlying access controls.