COURSE PROJECT

Enforcement Mechanisms and Strategies

Fred Schneider

Faculty Author
Cornell Bowers Computing and Information Science

Cornell University

Project Completed by: Germaine Wong

COURSE PROJECT — PART ONE

Enforcement With Monitoring

1. For this part of the project, you will identify which of the enforcement mechanisms designed previously are examples of a monitor. For each, evaluate to what extent that enforcement mechanism implements complete mediation. Are there any security policies for your system that cannot be enforced with a monitor? Feel free to revise some of your designs from the previous course if appropriate.

The enforcement mechanisms that are examples of a monitor are Policy Enforcement Points (PEPs), reference monitors, and guards. PEP ensures no subject can interact with an object without being checked first. It intercepts all access requests, is tamper-resistant through the enforcement logic being a part of a trusted computing base and it is verifiable because logic can be analyzed and tested against policy specifications. It actively enforces security policy at the runtime layer. A reference monitor intercepts every attempted subject-object access, it validates a subject's clearance against the object's classification, and it enforces MLS rules (Bell La-Padula for confidentiality, Biba for integrity, or a combination of both). A guard is a type of monitor because it actively intercepts and enforces policy on the entire information flow it mediates. It implements complete mediation by being the only channel for information flow between two domains, intercepting every request (message, packet, file transfer), applying policy consistently and without exception.

I do not have any security policies with my system that cannot be enforced with a monitor. My security policies are: 1. based on classification levels of subject, object, and the labels associated with them, which require a monitor because the policy can only be reliably enforced if a monitor is in place to validate every subject-object access against labels 2. security label-based access controls, which require a monitor because labels themselves are not a monitor but label-based access controls can be enforced with a monitor 3. security labels with clearance and classification labels, which require a monitor because the monitor enforces rules and labels.

Facilitator’s Comments

Thank you Germaine,

&nbsp;

You have covered the example for the reference monitor pretty well.&nbsp;

Grade

100%

COURSE PROJECT — PART TWO

Enforcement With Rewriting

1. For this part of the project, you will identify which of the enforcement mechanisms designed previously are examples of rewriters. If none, identify one or more security policies in your system that could be enforced with a rewriter. For each, evaluate the trade-offs between using a rewriter and alternative enforcement mechanisms. Feel free to revise some of your designs from the previous module if appropriate.

I do not have any enforcement mechanisms that are an example of a rewriter. The security policies I have in my system are based on on classification levels of subject, object, and labels associated with them; and security label-based access controls. In the first policy, which references the grant command, a rewriter can enforce the policy by transforming a program's code to insert guards that compare subject and object labels before any access is given. For the second policy, which involves a different grant command, the rewriter can enforce it if the security labels are available and interpretable by the rewriter and access patterns are predictable enough for rewriting to be effective.

The tradeoffs of the first policy are that a rewriter is good for simple policies that are easy to use in controlled environments that are performance-sensitive with static labels, but it would need access to a subject's identity or clearance. An alternative enforcement mechanism would use anonymous credentials and is suitable for MLS, good for dynamic labels, and prevent write-downs. For a secure anonymous communication system, alternative enforcement mechanisms should be used to maintain strong confidentiality and anonymity. The mechanisms should ensure policies are enforced at the system level, enforced at runtime, and include trusted components.

The tradeoffs for the second policy are that the rewriter is good for when enforcement is integrated into app logic, when the priority is performance and the ease of integration and enforcing simple policies, when the rules are static, but requires knowing a user's clearance and it's difficult to enforce write rules (no write-downs). An alternative enforcement mechanism would have read and write controls, policy completeness and be used with policies that have dynamic labels, but is more complex. Rewriting is incompatible with anonymity, so alternative enforcement mechanisms are better suited (reference monitor, secure enclave, etc).

Facilitator’s Comments

The facilitator did not supply a comment.

Grade

100%

COURSE PROJECT — PART THREE

Enforcement With Isolation

1. The final step is to identify which of the enforcement mechanisms designed previously are examples of isolation. If none, identify one or more security policies in your system that could be enforced with isolation. Feel free to revise some of your designs from the previous module if appropriate. Would the security of your system benefit or suffer from being run on a cloud-based co-located virtual machine instead of on a dedicated server maintained by your organization?

The enforcement mechanisms that are examples of isolation in my secure anonymous communication system are: reference monitors, guards, and ACLs. Reference monitors are isolated from other parts of a system so that no one can bypass or change it, they are placed in an OS kernel; they are separated, protected, and in control of access decisions. The isolation mechanisms a reference monitor uses are: TCB separation (small, making it easy to test and verify) and the mechanism that protects the kernel, where the monitor is located, that isolates the execution of the reference monitor. A reference monitor can also work with time multiplexing to maintain isolation between processes - when a system switches between processes, no process can access any data it should not be.

Guards act as independent gatekeepers between systems and are often in a separate part of the network. By being isolated, guards ensure that all traffic must go through it, which prevents bad data from spreading between networks. The isolation mechanisms that a guard uses are: physical isolation, where in some cases a guard is run on a separate machine or network segment to prevent direct interactions between systems; and logical isolation through mediation of interactions between systems.

ACLs use logical isolation by separating users from resources via the use of permissions that ensure one person's actions don't interfere with another person's protected data. The isolation mechanisms than an ACL uses are: logical (permissions-based) isolation, via creating barriers between different segments of a network; OS enforcement - the OS mediates access requests according to an ACL which prevents unauthorized users from accessing protected data; and process level isolation, which is indirect - when multiple users or processes access a system, ACLs help prevent one process from affecting another processes' resources or memory.

The security of a secure anonymous communication system would benefit from being run on a cloud-based co-located virtual machine instead of on a dedicated server maintained by the organization that makes the app. Since VMs are handled under the IaaS shared responsibility cloud service, the CSP will handle the patches while the organization that makes the communication system handles installation, configuration, and managing of their software apps and manages their security software and encryption keys. Communication servers, network software and encryption tools would all be installed on the VM by the company that makes the system. Because a secure anonymous communication system uses end-to-end-encryption and is encrypted on the sender's device and decrypted on a receiver's device, a CSP would only see ciphertext as it crosses through the cloud.

Facilitator’s Comments

The facilitator did not supply a comment.

Grade

100%

eCornell

Provide Feedback on this PDF

Submit Feedback