Formal Methods for MILS: Formalisations of the GWV Firewall

Ruud Koolen and Julien Schmaltz

Eindhoven University of Technology
Department of Mathematics and Computer Science
Model Driven Software Engineering
Formal System Analysis

20 January 2015

EURO MILS

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

MILS Formal correctness proofs

Compositionality: prove properties of the system by proving properties of its components.

- Prove that the separation kernel component behaves as it should
- For a high-security component *C*, prove that it does not leak any secrets
- Using these two properties, prove that any untrusted applications cannot gain access to *C* secrets

This results in strong assurance that *C* secrets are safe even in the presence of low-assurance components.

General approach: devise axioms for components to satisfy that function as their verification interface.

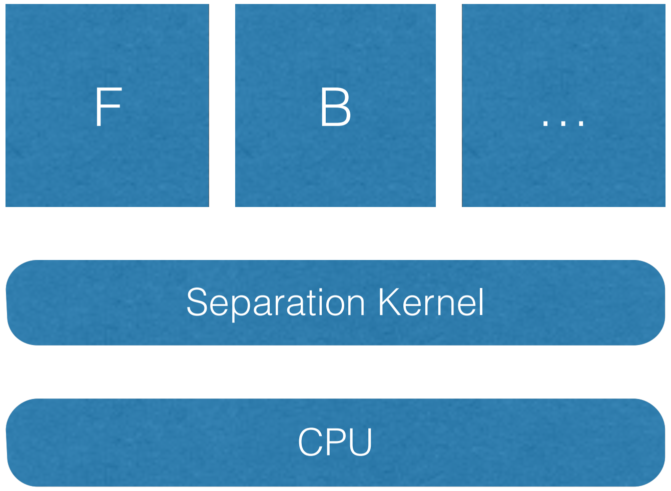
The GWV Firewall

The Firewall is an application that:

- Collects information from high-security components
- Removes the sensitive bits (that is, “blackens”)
- Forwards the sanitised information to a low-security untrusted application

The separation kernel ensures that the Firewall is the only source of information for the untrusted application.

From this, we want to prove that only sanitised information can enter the untrusted applications.

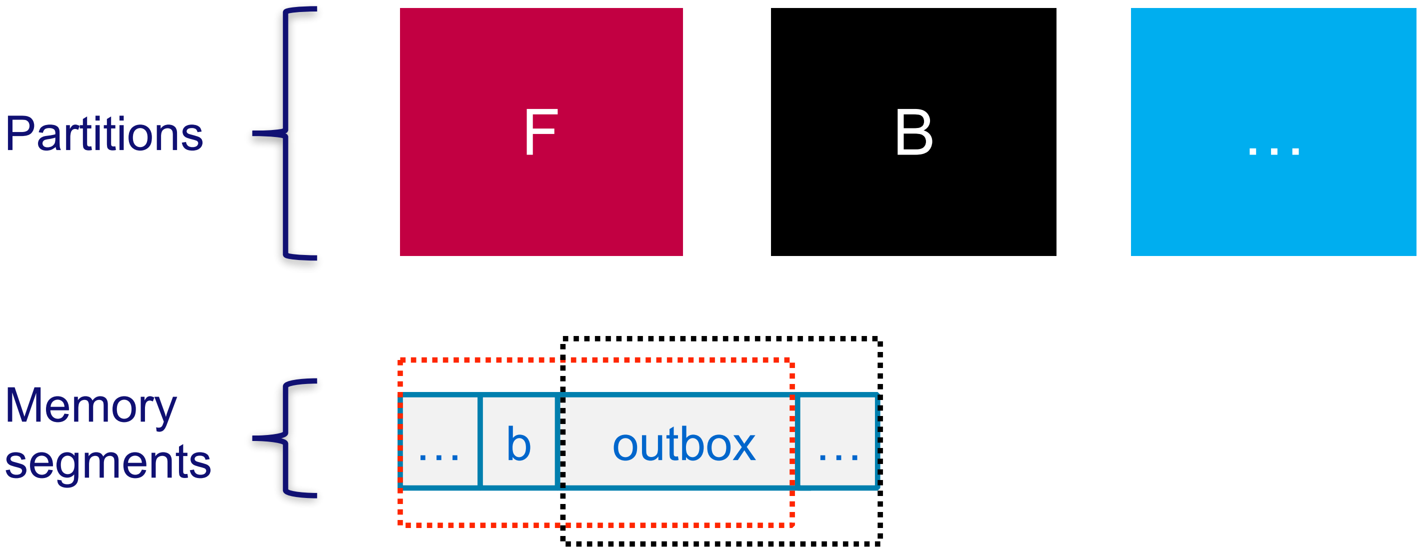


F is a partition running the firewall application.

B is partition reading data sent by the firewall.

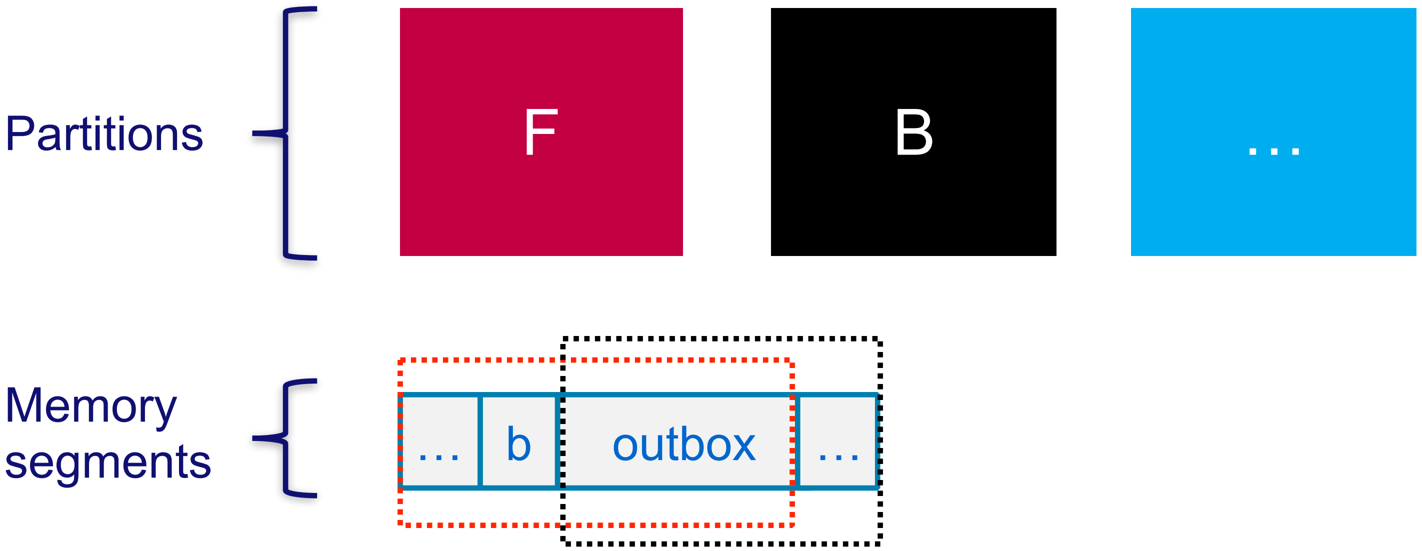
… denotes any other partition.

GWV - Formalising the separation kernel



Sep: The next value of segment *a* is a function of the current active partition and the values of the segments active and allowed to flow information to *a.*

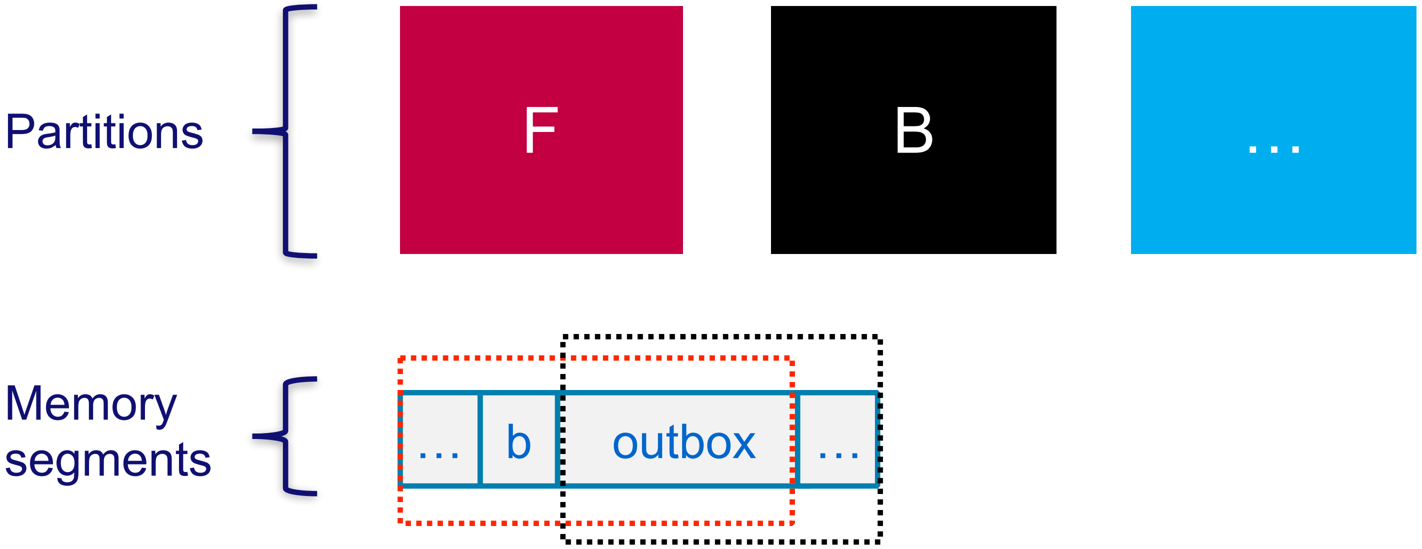
Firewall behaviour - FW_Blackens



Sep: The next value of segment *a* is a function of the current active partition and the values of the segments active and allowed to flow information to *a.*

FW_Blackens: The Firewall writes black data to segment outbox. More formally: If F is active and outbox is black, then outbox remains black in the next state.

Firewall Information Flow Policy - FW_Pol

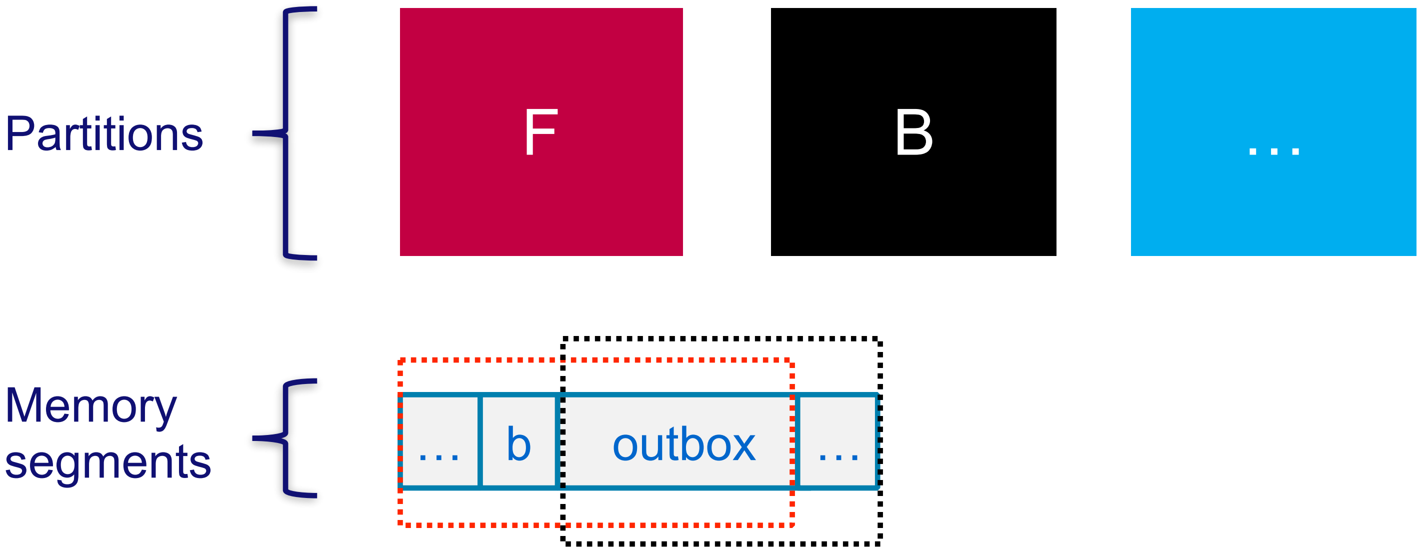


Sep: The next value of segment *a* is a function of the current active partition and the values of the segments active and allowed to flow information to *a.*

FW_Blackens: The Firewall writes black data to segment outbox. More formally: If F is active and outbox is black, then outbox remains black in the next state.

FW_Pol:
The only partition besides B from which information is permitted to flow to B is F. Such a flow is only permitted via segment outbox.

Firewall Correctness - FW_Correct



Sep: The next value of segment *a* is a function of the current active partition and the values of the segments active and allowed to flow information to *a.*

FW_Blackens: The Firewall writes black data to segment outbox. More formally: If F is active and outbox is black, then outbox remains black in the next state.

FW_Pol:
The only partition besides B from which information is permitted to flow to B is F. Such a flow is only permitted via segment outbox.

Desired system property:
FW_Correct: All segments of B once black always remain black.

Firewall correctness proof

Sep: The next value of segment *a* is a function of the current active partition and the values of the segments active and allowed to flow information to *a.*

FW_Blackens: The Firewall writes black data to segment outbox. More formally: If F is active and outbox is black, then outbox remains black in the next state.

FW_Pol:
The only partition besides B from which information is permitted to flow to B is F. Such a flow is only permitted via segment outbox.

Desired system property:
FW_Correct: All segments of B once black always remain black.

We cannot prove FW_Correct from Sep, FW_Blackens, and FW_Pol.

We need additional assumptions characterising the behaviour of blackness. For instance, if nonsensitive information could suddenly become sensitive on its own, then clearly the system security requirement does not hold.

Formalising blackness — our contribution

The behaviour of sensitive information turns out to be surprisingly hard to formalise.

We compare different formalisations of blackness:

- Original by **GWV** (2003) - function scrub + 6 axioms (ACL2)
- By **Rushby** (2004) - function blacken + 5 axioms (PVS)
- By **Van der Meyden** (2010) - predicate Black (Pencil&Paper)

We study how they relate to each other, and the strengths and weaknesses of the different approaches.

Due to time constraints, we compare in the talk only Rushby and Van der Meyden approaches.

Note that all our definitions and proofs have been formalised using the Isabelle/HOL theorem prover.

Blackness according to Rushby

According to Rushby, blackness should have three key properties:

- Blackness is a function of the contents of a memory segment. That is, if two states have the same contents for a segment, they also have the same blackness.
- Whenever all segments in the system are black, they are still black in the next state.
- For any state, we can construct a state in which all nonblack segments are replaced with black variants
  - Intuition: take all nonblack segments and wipe them. This gets you a state with all segments black.

Together these properties formalise the idea that nonblack data cannot be generated from black data.

Issues with Rushby blackness

The Rushby formalisation is sufficient to prove the security property of the system containing the firewall.

But the Rushby formalisation has issues:

- The existence of all-black states is a strong requirement on the available states of the system. Is this reasonable? What about chunks of memory that are always sensitive?
- Blackness is not really a function of memory contents. Whether or not a piece of data is sensitive is not a property of the data, it is a property of where the data came from.
    - Is “foobar123” a password (sensitive) or random noise (not sensitive)?

Blackness according to Van der Meyden

Van der Meyden requires more directly that nonblack data cannot be generated from black data.

The required property is as follows:

- Whenever the contents of segment *a* in the next state depends only on the contents of the set of segments *X*, if *X* are all black, *a* will be black in the next state.

Blackness according to Van der Meyden

Van der Meyden requires more directly that nonblack data cannot be generated from black data.

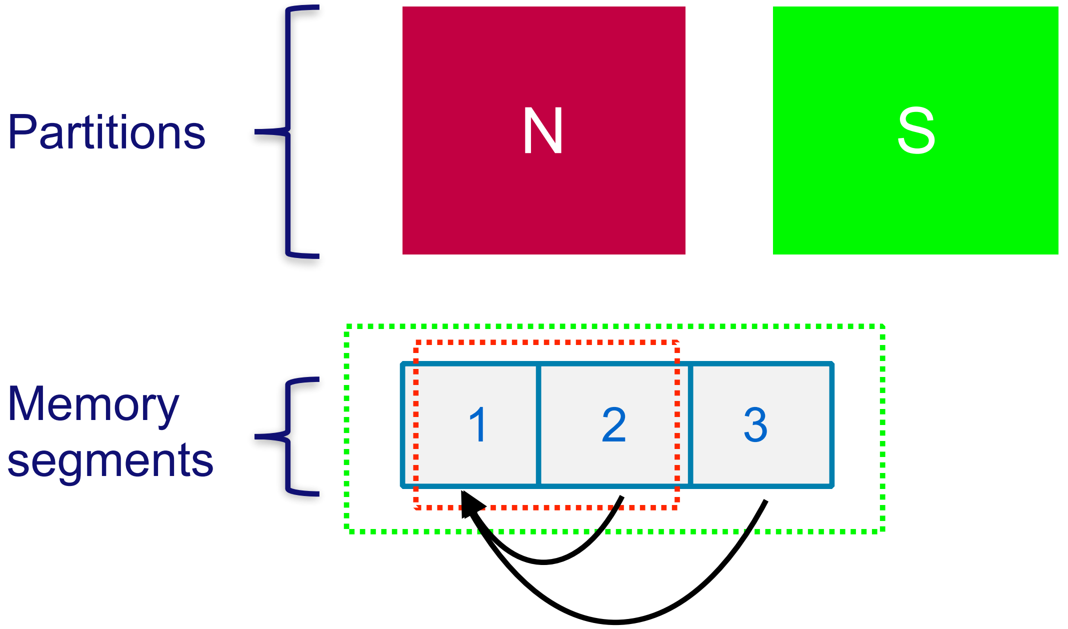
The required property is as follows:

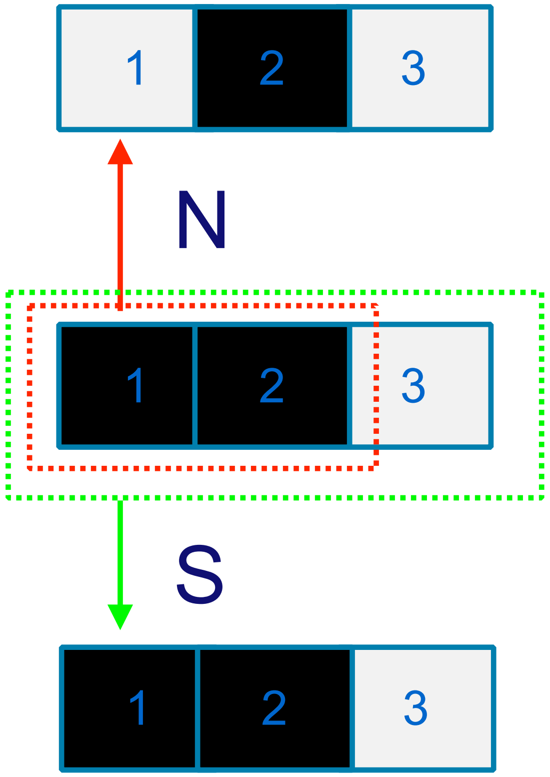
- Whenever the contents of segment a in the next state depends only on the contents of the set of segments *X*, if *X* are all black, *a* will be black in the next state.

But this is insufficient.

What if those segments of *X* that are accessible to the active partition are black, but the rest are not?

Issue with Van der Meyden Black predicate





S uses 1, 2, and 3 to compute 1; so, 1 depends on 1, 2, and 3.

So the Black axiom only requires N to make 1 black when 1, 2, and 3 are black — even though N cannot access 3.

This **insecure** system is indeed Black.

Possible fix

A possible fix is to require more directly that only the segments accessible to the active partition need to be black:

- Whenever the contents of segment *a* in the next state depends only on the contents of the set of segments *X*, if the segments of *X* accessible to the active partition are all black, *a* will be black in the next state.

This is sufficient to prove the security property we are aiming for …

But it has problems. We are embedding the security policy enforced by the separation kernel into the definition of black.

We require details of the separation kernel to prove this property of black, breaking independence of components!

Conclusion

None of the characterisations of blackness that we looked at are really satisfactory.

Yet a good axiomatisation of the flow of secrets will be essential for formally proving security properties of MILS systems in a compositional way.

Separation kernels restrict the flow of information between components. But that does not help us much if we cannot prove that the flow of secrets follows the flow of information.

Note that all our definitions and proofs have been formalised in Isabelle/HOL.

EURO-MILS CONTRACT N0: 318353

“This project has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement n °318353.”

If you need further information, please contact the coordinator:

Technikon Forschungs- und Planungsgesellschaft mbH
Burgplatz 3a, 9500 Villach, AUSTRIA
Tel: +43 4242 233 55   Fax: +43 4242 233 55 77
E-Mail: coordination@euromils.eu

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.