6.40 Templates and Generics [SYM]

Fortran does not support templates or generics, so the vulnerability specified in ISO/IEC TR 24772-1:2019 clause 6.40 does not apply to Fortran.

6.41 Inheritance [RIP]

6.41.1 Applicability to language

The vulnerability specified in ISO/IEC TR 24772-1:2019 clause 6.41 applies to Fortran since Fortran supports inheritance and redefinition of type-bound subprograms. Fortran supports single inheritance only, so the complexities associated with multiple inheritance do not apply. The problem of accidental redefinition is partially mitigated by the non overridable attribute which prevents overriding by all subclasses. There is no mechanism to restrict a type-bound subprogram to be a redefinition or a new subprogram, respectively. Hence the vulnerabilities of accidental redefinition apply.

6.41.2 Guidance to language users

Follow the guidance of ISO/IEC TR 24772-1:2019 clause 6.41.5.

- Declare a type-bound procedure to be non overridable when necessary to ensure that it is not overridden by subclasses.
- Provide a private component to store the version control identifier of the derived type, together with an accessor routine. <<<interesting idea, but needs substantiation in 6.41.1.>>

6.42 Violations of the Liskov Substitution Principle or the Contract Model [BLP]

6.42.1 Applicability to language

The vulnerability specified in ISO/IEC TR 24772-1:2019 clause 6.42 applies to Fortran. Fortran has no means to specifiy and enforce pre- and postconditions, or to prevent "has-a"-inheritance.

6.42.2 Guidance to language users

Follow the guidance of ISO/IEC TR 24772-1:2019 clause 6.42.5.

6.43 Redispatching [PPH]

6.43.1 Applicability to language

The vulnerability specified in ISO/IEC TR 24772-1:2019 clause 6.45 applies to Fortran, since calls to type-bound procedures inside inherited implementation dispatch to the dynamic type of the object in question. To prevent redispatching, Fortan provides <<<what? Other languages use view-conversion of the object or qualifications of the call by the class>>>>

6.43. 2 Guidance to language users

Follow the guidance of ISO/IEC TR 24772-1:2019 clause 6.43.5.

6.44 Polymorphic Variables

6.44.1 Applicability to language

The vulnerability specified in ISO/IEC TR 24772-1:2019 clause 6.45 applies to Fortran, as Fortran provides polymorphic variables. Upcasts, as described in ISO/IEC TR 24772-1:2019 clause 6.45, are implicit in assignments and parameter passing, which always allow a value of a class to be assigned to a variable declared to be of any of its superclasses. Crosscasts or other unsafe casts are not possible in Fortran. Downcasts are realized by type select statements, where a variable selected upon assumes the selected type as its declared type for the extent of the respective branch. <<< presuming that the check is an inclusion, not an equality check>>. Care needs to be taken to check for more specific class first, because an early check on a superclass makes all subsequent branches on its subclasses unreachable. The vulnerability of not handling the error situation by default brach in the type select statement remains and needs to be handled. See <<< the clause on error handling>>>

6.44. 2 Guidance to language users

Software developers can avoid the vulnerability or mitigate its ill effects in the following ways:

- When upcasting:
 - Ensure functional consistency of the subclass-specific data to the changes affected via the upcasted reference.
- Make sure that you handle the handle the default case in type select statements.
- Always handle the more specific classes first in type select statements.
- •