CLAIMS:

1. A device for reinforcing control flow integrity of a software application comprising:

a processor; and

a non-transitory media readable by the processor, the non-transitory media storing instructions that when executed by the processor, cause the processor to:

receive a first source code file for the software application whereby the first source code file comprises functions and indirect function calls;

compile the first source code file to generate a first binary file whereby for each compiled indirect function call,

associating a control flow integrity check function to the each compiled indirect function call in the first binary file such that the associated control flow integrity check function will be executed prior to the execution of the each compiled indirect function call, and

adding a function, referred to in the each compiled indirect function call, to a reduced function table;

load the first binary file for execution;

update a binary map with an address referred to in a load command when the load command in the first binary file is executed;

update a runtime function table with a function referred to in an indirect external function call when an assignment of the indirect external function call is executed in the first binary file;

execute the compiled first binary file, whereby for each control flow integrity check function executed, the control flow integrity check function determines if a function referred to in an associated indirect function call is contained within the reduced function table; and

execute the associated indirect function call when it is determined that the function referred to in the associated indirect function call is contained within the reduced function table.

2. The device according to claim 1 further comprising:

instructions for directing the processor to:

execute the associated indirect function call,

when it is determined that the function referred to in the associated indirect function call is not contained within the reduced function table,

when the function referred to in the associated indirect function call is contained within a second binary file having an address contained in the binary map, and

when the second binary file is determined to have been compiled in a legacy manner.

3. The device according to claim 1 further comprising:

instructions for directing the processor to:

execute the associated indirect function call,

when it is determined that the function referred to in the associated indirect function call is not contained within the reduced function table,

when the function referred to in the associated indirect function call is contained within a second binary file having an address contained in the binary map,

when the second binary file is determined to not have been compiled in a legacy manner, and

when the function referred to in the associated indirect function call is contained within a reduced function table associated with the second binary file.

4. The device according to claim 1 further comprising:

instructions for directing the processor to:

execute the associated indirect function call,

when it is determined that the function referred to in the associated indirect function call is not contained within the reduced function table,

when the function referred to in the associated indirect function call is contained within a second binary file having an address contained in the binary map,

when the second binary file is determined to not have been compiled in a legacy manner,

when the function referred to in the associated indirect function call is not contained within a reduced function table associated with the second binary file, and

when the function referred to in the associated indirect function call is contained in the runtime function table.

5. The device according to claim 1 further comprising:

instructions for directing the processor to:

execute the associated indirect function call,

when it is determined that the function referred to in the associated indirect function call is not contained within the reduced function table,

when the function referred to in the associated indirect function call is contained within a second binary file having an address contained in the binary map, and

when the function referred to in the associated indirect function call is contained in the runtime function table.

6. The device according to any one of claims 1 to 5 wherein the runtime function table is stored in a protected non-transitory media readable by the processor.

7. The device according to any one of claims 1 to 6 wherein the binary map is stored in a protected non-transitory media readable by the processor.

8. The device according to any one of claims 1 to 6 wherein the indirect function calls comprise indirect function calls assigned as part of a data structure.

9. The device according to claim 8 wherein the indirect function calls further comprise indirect function calls directly assigned to functions.

10. The device according to claim 9 wherein the indirect function calls further comprise indirect function calls assigned as assembly function calls.

11. A method for reinforcing control flow integrity of a software application using a computing device, the method comprising:

receiving a first source code file for the software application whereby the first source code file comprises functions and indirect function calls;

compiling the first source code file to generate a first binary file whereby for each compiled indirect function call,

associating a control flow integrity check function to the each compiled indirect function call in the first binary file such that the associated control flow integrity check function will be executed prior to the execution of the each compiled indirect function call, and

adding a function, referred to in the each compiled indirect function call, to a reduced function table;

loading the first binary file for execution;

updating a binary map with an address referred to in a load command when the load command in the first binary file is executed;

updating a runtime function table with a function referred to in an indirect external function call when an assignment of the indirect external function call is executed in the first binary file;

executing the compiled first binary file, whereby for each control flow integrity check function executed, the control flow integrity check function determines if a function referred to in an associated indirect function call is contained within the reduced function table; and

executing the associated indirect function call when it is determined that the function referred to in the associated indirect function call is contained within the reduced function table.

12. The method according to claim 11 further comprising:

executing the associated indirect function call,

when it is determined that the function referred to in the associated indirect function call is not contained within the reduced function table,

when the function referred to in the associated indirect function call is contained within a second binary file having an address contained in the binary map, and

when the second binary file is determined to have been compiled in a legacy manner.

13. The method according to claim 11 further comprising:

executing the associated indirect function call,

when it is determined that the function referred to in the associated indirect function call is not contained within the reduced function table,

when the function referred to in the associated indirect function call is contained within a second binary file having an address contained in the binary map,

when the second binary file is determined to not have been compiled in a legacy manner, and

when the function referred to in the associated indirect function call is contained within a reduced function table associated with the second binary file.

14. The method according to claim 11 further comprising:

executing the associated indirect function call,

when it is determined that the function referred to in the associated indirect function call is not contained within the reduced function table,

when the function referred to in the associated indirect function call is contained within a second binary file having an address contained in the binary map,

when the second binary file is determined to not have been compiled in a legacy manner,

when the function referred to in the associated indirect function call is not contained within a reduced function table associated with the second binary file, and

when the function referred to in the associated indirect function call is contained in the runtime function table.

15. The method according to claim 11 further comprising:

executing the associated indirect function call,

when it is determined that the function referred to in the associated indirect function call is not contained within the reduced function table,

when the function referred to in the associated indirect function call is contained within a second binary file having an address contained in the binary map, and

when the function referred to in the associated indirect function call is contained in the runtime function table.

16. The method according to any one of claims 11 to 15 wherein the runtime function table is stored in a protected non-transitory media readable by the processor.

17. The method according to any one of claims 11 to 16 wherein the binary map is stored in a protected non-transitory media readable by the processor.

18. The method according to any one of claims 11 to 17 wherein the indirect function calls comprise indirect function calls assigned as part of a data structure.

19. The method according to claim 18 wherein the indirect function calls further comprise indirect function calls directly assigned to functions.

20. The method according to claim 19 wherein the indirect function calls further comprise indirect function calls assigned as assembly function calls.