What is claimed is:

1. A method comprising:

executing, by a computer system, a program in a first mode of operation in a controlled environment in accordance with a normal operation without malicious behavior;

generating, by the computer system, a record of events comprising a plurality of sequences of events that occur during the normal operation of the program;

generating, by the computer system using the record of events, an acceptable behavior model that is indicative of normal behavior flow control, flow status, or data flow of actions performed by of the program that occur during the normal operation without the malicious behavior;

executing, by the computer system, the program in a second mode of operation after the program has been deployed in runtime in a non-isolated, real-world, operational network environment;

determining, by the computer system, an operational sequence of events of the program during execution of the program in the second mode of operation, the operational sequence of events including a current action;

comparing, by the computer system, the operational sequence of events with the acceptable behavior model;

when the comparing step results in a match between the operational sequence of events and the acceptable behavior model, performing the current action in the second mode of operation; and

when the comparing step does not result in the match between the operational sequence of events and the acceptable behavior model, not performing the current action and generating an alert to stop the executing of the program.

2. The method of claim 1, wherein:

each sequence of events of the plurality of sequences of events comprises a plurality of events that occur during the normal operation of the program.

3. The method of claim 1, wherein:

the program is written in an interpreted language; and

the method further comprises:

instrumenting code of the program to include instrumented code; and

interpreting the program by a code interpreter to generate a first portion of executable code that is based on the code of the program and a second portion of executable code that is based on the instrumented code.

4. The method of claim 3, wherein:

the executing of the program in the first mode of operation and in the second mode of operation is performed after interpreting the program by the code interpreter; and

the generating of the record of events is performed by executing the second portion of executable code.

5. The method of claim 3, wherein:

the instrumenting of the code of the program to include the instrumented code is performed before the program is provided to the code interpreter.

6. The method of claim 3, wherein:

the instrumenting of the code of the program to include the instrumented code is performed by the code interpreter while interpreting the code of the program.

7. The method of claim 1, further comprising:

instrumenting, by the computer system using a compiler, code of the program, the program containing instrumented executable code;

wherein:

the executing of the program in the first mode of operation results in the generating of the record of events in accordance with instrumentation of the instrumented executable code of the program.

8. The method of claim 1, wherein:

the executing of the program in the first mode of operation is performed by a first processor of the computer system; and

the generating of the record of events is performed by a second processor separate from the first processor.

9. The method of claim 1, wherein:

the executing of the program in the second mode of operation is performed by a first processor of the computer system;

the determining of the operational sequence of events is performed by a second processor of the computer system separate from the first processor.

10. The method of claim 1, wherein:

the acceptable behavior model is an artificial intelligence model trained on the plurality of sequences of events, which are known to occur during the normal operation of the program without the malicious behavior.

11. A method comprising:

executing, by a computer system, a program in a first mode of operation in a controlled environment in accordance with a normal operation without malicious behavior;

generating, by the computer system, a record of events comprising a plurality of sequences of events that occur during the normal operation of the program; and

generating, by the computer system using the record of events, an acceptable behavior model that is indicative of normal behavior of flow control, flow status, or data flow of actions performed by the program that occur during the normal operation without the malicious behavior;

wherein:

the acceptable behavior model is configured to be used to prevent execution of a current action of the program in a second mode of operation after the program has been deployed in runtime in a non-isolated, real-world, operational network environment when it is determined that the current action of the program is part of an operational sequence of events that does not match the acceptable behavior model.

12. The method of claim 11, wherein:

each sequence of events of the plurality of sequences of events comprises a plurality of events that occur during the normal operation of the program.

13. The method of claim 11, wherein:

the program is written in an interpreted language; and

the method further comprises instrumenting code of the program to include instrumented code;

the method further comprises interpreting the program by a code interpreter to generate a first portion of executable code that is based on the code of the program and a second portion of executable code that is based on the instrumented code;

the executing of the program in the first mode of operation and in the second mode of operation is performed after interpreting the program by the code interpreter; and

the generating of the record of events is performed by executing the second portion of executable code.

14. The method of claim 11, further comprising:

instrumenting, by the computer system using a compiler, code of the program, the program containing instrumented executable code;

wherein:

the executing of the program in the first mode of operation results in the generating of the record of events in accordance with instrumentation of the instrumented executable code of the program.

15. The method of claim 11, wherein:

the executing of the program in the first mode of operation is performed by a first processor of the computer system; and

the generating of the record of events is performed by a second processor of the computer system separate from the first processor.

16. The method of claim 11, wherein:

the acceptable behavior model is an artificial intelligence model trained on the plurality of sequences of events, which are known to occur during the normal operation of the program without the malicious behavior.

17. A method comprising:

receiving, by a computer system, an acceptable behavior model that is indicative of normal operation of flow control, flow status, or data flow of actions performed by a program without malicious behavior as determined by execution of

the program in a first mode of operation in a controlled environment in accordance with the normal operation without malicious behavior;

executing, by the computer system, the program in a second mode of operation after the program has been deployed in runtime in a non-isolated, real-world, operational network environment;

determining, by the computer system, an operational sequence of events of the program during execution of the program in the second mode of operation, the operational sequence of events including a current action;

comparing, by the computer system, the operational sequence of events with the acceptable behavior model;

when the comparing step results in a match between the operational sequence of events and the acceptable behavior model, performing the current action in the second mode of operation; and

when the comparing step does not result in the match between the operational sequence of events and the acceptable behavior model, not performing the current action and generating an alert to stop the executing of the program.

18. The method of claim 17, wherein:

the program is written in an interpreted language; and

the method further comprises instrumenting code of the program to include instrumented code;

the method further comprises interpreting the program by a code interpreter to generate a first portion of executable code that is based on the code of the program and a second portion of executable code that is based on the instrumented code;

the executing of the program in the second mode of operation is performed after interpreting the program by the code interpreter; and

the determining of the operational sequence of events is performed by executing the second portion of executable code.

19. The method of claim 17, wherein:

the program contains instrumented executable code; and

the executing of the program in the second mode of operation results in the generating of the operational sequence of events in accordance with instrumentation of the instrumented executable code of the program.

20. The method of claim 17, wherein:

the executing of the program in the second mode of operation is performed by a first processor of the computer system; and

the determining of the operational sequence of events is performed by a second processor of the computer system separate from the first processor.

21. A computerized system comprising:

a physical memory storing executable instructions; and

a hardware processor, coupled to the memory, that performs a method by executing the instructions stored in the memory, the method comprising:

receiving, by a computer system, an acceptable behavior model that is indicative of normal operation of flow control, flow status, or data flow of actions performed by a program without malicious behavior as determined by execution of the program in a first mode of operation in a controlled environment in accordance with the normal operation without malicious behavior;

executing, by the computer system, the program in a second mode of operation after the program has been deployed in runtime in a non-isolated, real-world, operational network environment;

determining, by the computer system, an operational sequence of events of the program during execution of the program in the second mode of operation, the operational sequence of events including a current action;

comparing, by the computer system, the operational sequence of events with the acceptable behavior model;

when the comparing step results in a match between the operational sequence of events and the acceptable behavior model, performing the current action in the second mode of operation; and

when the comparing step does not result in the match between the operational sequence of events and the acceptable behavior model, not performing the current action and generating an alert to stop the executing of the program.

22. The computerized system of claim 21, wherein:

the program is written in an interpreted language; and

the method further comprises instrumenting code of the program to include instrumented code;

the method further comprises interpreting the program by a code interpreter to generate a first portion of executable code that is based on the code of the program and a second portion of executable code that is based on the instrumented code;

the executing of the program in the second mode of operation is performed after interpreting the program by the code interpreter; and

the determining of the operational sequence of events is performed by executing the second portion of executable code.

23. The computerized system of claim 21, wherein:

the program contains instrumented executable code; and

the executing of the program in the second mode of operation results in the generating of the operational sequence of events in accordance with instrumentation of the instrumented executable code of the program.

24. The computerized system of claim 21, wherein:

the executing of the program in the second mode of operation is performed by a first processor of the computer system; and

the determining of the operational sequence of events is performed by a second processor of the computer system separate from the first processor.