DieHard

DieHard protects applications from as-yet unfixed bugs and security vulnerabilities that exploit them. Think of DieHard as a new line of defense against hackers, together with anti-virus protection and firewalls.

Download

Download DieHard on GitHub.

More Information

UMass Amherst press release

video interview with Ben Zorn on DieHard/RobustHeap

DieHard eliminates — or greatly reduces the likelihood of — a class of bugs and security vulnerabilities called *memory errors*. DieHard actually prevents certain kinds of errors from happening at all. It also reduces the probability that a bug will have any effect at all. DieHard wo by randomly locating program objects far apart from each other in memory. This scattering of memory objects all over memory not only makes some errors unlikely to happen, it also makes i virtually impossible for a hacker to know where vulnerable parts of the program's data are. This thwarts a wide class of exploits.

Technical Details

DieHard *prevents* invalid and multiple frees and heap corruption, and *probabilistically avoids* bul overflows, dangling pointer errors, and uninitialized reads. **This sample program** illustrates a w range of errors that DieHard prevents. For more details, see the following (technical) paper:

DieHard helps buggy programs run correctly and protects them from a range of security vulnerabilities.

Download DieHard now.

Do the following errors look familiar?

```
*** glibc detected *** double free or corruption
*** glibc detected *** free(): invalid pointer
*** glibc detected *** free(): invalid next size
```

DieHard can help! See the FAQs and our technical paper for detailed information.

DieHard: Probabilistic Memory Safety for Unsafe Languages
Emery D. Berger and Benjamin G. Zorn, ACM SIGPLAN 2006 Conference on Programming Language Design and Implementation (PLDI 2006).

Or view this **PowerPoint presentation**.

DieHard works in two modes: **standalone** and **replicated**. The standalone version replaces the memory manager with the DieHard *randomized* memory manager. This randomization increases the odds that buffer overflows will have no effect, and reduces the risk of dangling pointers. The replicated version provides greater protection against errors by running several instances of the application simultaneously and *voting* on their output. Because each *replica* is randomized differently, each replica will likely have a different output if it has an error, and some replicas are likely to run correctly despite the error.

The standalone version works for Linux, Solaris, and Windows, while the replicated version currently only supports Linux or Solaris console applications.

Terms of use

This work is supported in part by the National Science Foundation, Intel Corporation, and Microsoft Research. This material is based upon work supported by the National Science Foundation under Grant No CNS-0615211. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).


mozilla.exe
Debug
Don't Send