NAME

hardening−check − check binaries for security hardening features

SYNOPSIS

hardening-check [options] [ELF ...]

Examine a given set of ELF binaries and check for several security hardening features, failing if they are not all found.

DESCRIPTION

This utility checks a given list of ELF binaries for several security hardening features that can be compiled into an executable. These features are:

Position Independent Executable

This indicates that the executable was built in such a way (PIE) that the "text" section of the program can be relocated in memory. To take full advantage of this feature, the executing kernel must support text Address Space Layout Randomization (ASLR).

Stack Protected

This indicates that there is evidence that the ELF was compiled with the **gcc**(1) option **−fstack−protector** (e.g. uses **__stack_chk_fail**). The program will be resistant to having its stack overflowed.

When an executable was built without any character arrays being allocated on the stack, this check will lead to false alarms (since there is no use of **__stack_chk_fail**), even though it was compiled with the correct options.

Fortify Source functions

This indicates that the executable was compiled with **−D_FORTIFY_SOURCE=2** and **−O1** or higher. This causes certain unsafe glibc functions with their safer counterparts (e.g. **strncpy** instead of **strcpy**), or replaces calls that are verifiable at runtime with the runtime-check version (e.g. **__memcpy_chk** insteade of **memcpy**).

When an executable was built such that the fortified versions of the glibc functions are not useful (e.g. use is verified as safe at compile time, or use cannot be verified at runtime), this check will lead to false alarms. In an effort to mitigate this, the check will pass if any fortified function is found, and will fail if only unfortified functions are found. Uncheckable conditions also pass (e.g. no functions that could be fortified are found, or not linked against glibc).

Read-only relocations

This indicates that the executable was build with **−Wl,−z,relro** to have ELF markings (RELRO) that ask the runtime linker to mark any regions of the relocation table as "read-only" if they were resolved before execution begins. This reduces the possible areas of memory in a program that can be used by an attacker that performs a successful memory corruption exploit.

Immediate binding

This indicates that the executable was built with **−Wl,−z,now** to have ELF markings (BIND_NOW) that ask the runtime linker to resolve all relocations before starting program execution. When combined with RELRO above, this further reduces the regions of memory available to memory corruption attacks.

OPTIONS

−−nopie, **−p**

Do not require that the checked binaries be built as PIE.

−−nostackprotector, **−s**

Do not require that the checked binaries be built with the stack protector.

−−nofortify, **−f**

Do not require that the checked binaries be built with Fortify Source.

−−norelro, **−r**
Do not require that the checked binaries be built with RELRO.

−−nobindnow, **−b**
Do not require that the checked binaries be built with BIND_NOW.

−−quiet, **−q**
Only report failures.

−−verbose, **−v**
Report verbosely on failures.

−−report−functions, **−R**
After the report, display all external functions needed by the ELF.

−−find−libc−functions, **−F**
Instead of the regular report, locate the libc for the first ELF on the command line and report all the known ''fortified'' functions exported by libc.

−−color, **−c**
Enable colorized status output.

−−lintian, **−l**
Switch reporting to lintian-check-parsable output.

−−debug
Report some debugging during processing.

−−help, **−h**, **−?**
Print a brief help message and exit.

−−man, **−H**
Print the manual page and exit.

RETURN VALUE

When all checked binaries have all checkable hardening features detected, this program will finish with an exit code of 0. If any check fails, the exit code with be 1. Individual checks can be disabled via command line options.

AUTHOR

Kees Cook <kees@debian.org>

COPYRIGHT AND LICENSE

SEE ALSO

gcc (1), **hardening−wrapper** (1)