第158回東京エリアDebian勉強会

gccのpieオプションとdebianにおける状況について

Norimitsu Sugimoto (杉本 典充)
dictoss@live.jp

2017-12-16

アジェンダ

- 自己紹介
- はじめに
- PIE について
- PIE の採用状況
- Debian における PIE の状況
- まとめ
- 参考資料

自己紹介

- Norimitsu Sugimoto (杉本 典充)
- dictoss@live.jp
- Twitter: @dictoss
- Debian 使って 10 年以上、FreeBSD 使って 5 年以上
- Debian GNU/kFreeBSD が気になっておりウォッチ中
- 仕事はソフトウェア開発者をやってます
- python と Django の組み合わせで使うことが多いです

はじめに

- Debian 9 のリリースノート
  - 実行ファイルはデフォルトで PIE が有効でコンパイルされています
  - `https://www.debian.org/releases/stretch/amd64/release-notes/ch-information.ja.html#pie-is-now-default`
- file コマンドの結果が「ELF 64-bit LSB shared object」
- よく知らないため、調べてみよう！

PIE　について

PIE とは(1)

- PIC（Position Independent Code)、位置独立コード
  - 共有ライブラリ（.so）に含むソースコードは通常"-fPIC で"コンパイルする
  - オブジェクトファイルに含まれる機械語は相対アドレスで記述
  - 仮想アドレスのどの番地に配置されても実行可能
- PIE（Position Independent Executable)、位置独立実行形式
- PIE = PIC のみのオブジェクトファイルで構成した実行ファイル

PIE とは(2)

- （非 PIE な）今までの実行ファイル
  - 機械語を仮想アドレスのどの位置に配置するかはリンク時に決定
  - 実行時の仮想アドレスは毎回同じ位置に同じデータが配置
- PIE な実行ファイル
  - 実行時にダイナミックリンカーが相対アドレスで記述した機械語を絶対アドレスに変換し、仮想アドレスに配置する
  - 実行ファイル内の機械語と共有ライブラリの機械語の両方が変換対象
  - 上記処理が完了後、プログラムを実行する

PIE とは(3)

非 PIE な実行ファイルを実行（Debian 8 jessie）

```
$ gcc test.c
$ ./a.out
IN foo()
0x400546
$ ./a.out
IN foo()
0x400546
```

PIE な実行ファイルを実行（Debian 9 stretch）

```
$ gcc test.c
$ ./a.out
IN foo()
0xae27f6f0
$ ./a.out
IN foo()
0xbb1466f0
```

PIE の長所と短所

- 長所
  - アドレス変換処理を行うため、実行プログラムの仮想アドレスの位置が変わる
  - 脆弱性があるプログラムは特定コードを実行されにくくなる
  - セキュリティが向上する
- 短所
  - アドレス変換処理そのもののオーバーヘッドがあり、非 PIE に比べて実行速度が遅い
- 短所の克服
  - gcc-5.0 において PIC なコードの実行性能が改善したというレポート [1]

[1] https://software.intel.com/en-us/blogs/2014/12/26/new-optimizations-for-x86-in-upcoming-gcc-50-32bit-pic-mode

PIE の実行ファイルを生成する

- gcc と実行環境
  - gcc のコンパイルオプション"-fPIE"を付与
  - gcc のリンクオプション"-pie"を付与
  - PIE な実行ファイルを実行には、対応しているダイナミックローダーが必要
- PIE な実行ファイルを扱うために必要なソフトウェア
  - gcc-3.4 以降
  - binutils-2.15 以降（ld コマンド）
  - ダイナミックリンカー（ld.so、ld-linux.so）
  - gdb-7.1 以降

PIE の採用状況

PIE の採用状況

- OpenBSD 5.3 (2013-05-01)
- Fedora 23 (2015-11-03)
- Ubuntu 16.10 (2016-10-13)
- Debian 9 (2017-06-16)
- HardenedBSD
- Android 5.0 (2014-10-17)

Debian における PIE の状況

PIE に関する情報提供

- Debian Wiki "Hardening"
  - https://wiki.debian.org/Hardening
  - https://wiki.debian.org/Hardening/PIEByDefaultTransition
- 「Porter roll call for Debian Stretch」
  - stretch の実行ファイルは PIE をデフォルトにするかの意見募集メール
  - https://lists.debian.org/debian-devel/2016/08/msg00324.html
- Lintian
  - hardening-no-pie
  - https://lintian.debian.org/tags/hardening-no-pie.html

gcc パッケージ

- stretch の gcc は、gcc-6.3.0
- gcc の configure オプションに"–enable-default-pie" を指定している
- gcc のマニュアルには「Turn on -fPIE and -pie by default. 」とある
- stretch の提供する gcc でアプリケーションをビルドすると、デフォルトで PIE な実行ファイルが生成される
- PIE な実行ファイルが動かない CPU アーキテクチャがあり、その場合は無効になっている（hppa、m68k）

PIE形式にしたくない場合

- gccのリンクオプション"-no-pie"を指定して実行ファイルを生成する
- 「./a.out: ELF 64-bit LSB executable」

まとめ

- PIE とは、PIC のみの機械語を含む実行ファイル
- gcc が生成する機械語の改善があり、実行性能問題が解決した
- セキュリティを高めるために PIE な実行ファイルをデフォルトで提供を始めた
- gcc 自体のビルドオプションを変更して有効にしており、アプリケーション開発者としてはうまく付き合う必要がある

参考文献

- Debian Wiki "Hardening" [2]
- ももいろテクノロジー ELF 実行ファイルのメモリ配置はどのように決まるのか [3]
- Red Hat Security Blog Position Independent Executables (PIE) [4]
- OpenBSD's Position Independent Executable (PIE) Implementation [5]
- New optimizations for X86 in upcoming GCC 5.0: PIC in 32 bit mode. [6]

---

[2] https://wiki.debian.org/Hardening

[3] http://inaz2.hatenablog.com/entry/2014/07/27/205913

[4] https://access.redhat.com/blogs/766093/posts/1975793

[5] http://www.openbsd.org/papers/nycbsdcon08-pie/mgp00001.html

[6] https://software.intel.com/en-us/blogs/2014/12/26/new-optimizations-for-x86-in-upcoming-gcc-50-32bit-pic-mode