[RElog] Dlaczego nie umieszczać haseł w kodzie źródłowym (sheadovas/artykuly/relog/dlaczego-nie-umieszczac-hasel-w-kodzie-zrodlowym/)

Gru 28, 2016 / RElog (sheadovas/category/artykuly/relog/)

Oraz po części o tym jak poprawnie(j) zrealizować logowanie do np. bazy danych

Hej, jakiś czas temu poruszałem podobny temat w [RElogu (sheadovas/artykuly/relog/relog-ukrywanie-danych-stringow-wewnatrz-pliku-wykownywalnego/)], dzisiaj chciałbym spojrzeć na niego jeszcze raz, ale tym razem od nieco innej strony.

Scenariusz: jesteśmy programistą, który ma program używający globalnej bazy danych napisany w języku kompilowanym do postaci binarnej. Chcielibyśmy udostępnić ten program swojemu złośliwemu znajomemu, jednak nie wiemy na ile jest to bezpieczne.

Poniżej przedstawię kilka „implementacji”, gdzie przekazujemy dane potrzebne do zalogowania, a następnie przedstawię słabości danych rozwiązań.

W poniższych przykładach korzystam z kompilatora GCC i debuggera GDB (z pluginem peda).

Naiwnie

Ktoś kiedyś powiedział, że „prostota jest szczytem wyrafinowania” (~*Leonardo da Vinci*), więc zgodnie z tym cytatem nie przemęczajmy się i zrealizujmy całość w najprostszy możliwy sposób:

C

```c
#include <stdio.h>
#include <stdbool.h>

bool connect(const char* hostname, const char* username, const char* password)
{
  // ... <- realizacja polaczenia z baza
  printf("Connecting...\n");
  return true;
}

int main(int argc, char** argv)
{
  const char* hostname = "127.0.0.1";
  char login[] = "root";

  if(connect(hostname, login, "toor"))
  {
    // ... <- zrob cos, jezeli sie polaczyles z sukcesem
  }
  else
  {
    return 1;
  }

  return 0;
}
```

Pseudo-logowanie używające stringów zapisanych w różnej postaci. Program po prostu wyświetla słowo „Connecting...”

Debug vs Release

(dzięki @maly za zwrócenie uwagi) Większość z Was pewnie korzysta z dużych środowisk programistycznych i zauważyła, że zazwyczaj dostępne są 2 tryby: *Debug* oraz *Release*. Tryb Debug zawiera dodatkowe symbole pomagające przy debugowaniu programu, które mogą być też potencjalnie przydatne przy jego analizie, jak np w pełni zrozumiałe nazwy funkcji, mogą się też zdarzyć dodatkowe komunikaty etc

Poprzedni kod jako „Release”

Co jeżeli skompilujemy program bez dodatkowych symboli, czy kompilator nie zamieni czasem wszystkiego w „binarne krzaczki”, przekonajmy się:

```
$ gcc simplelogin.c -o simplelogin
$ strip simplelogin
$ xxd simplelogin
00000000: 7f45 4c46 0201 0100 0000 0000 0000 0000  .ELF............
00000010: 0200 3e00 0100 0000 c004 4000 0000 0000  ..>.......@.....
00000020: 4000 0000 0000 0000 7011 0000 0000 0000  @.......p.......
00000030: 0000 0000 4000 3800 0900 4000 1d00 1c00  ....@.8...@.....
00000040: 0600 0000 0500 0000 4000 0000 0000 0000  ........@.......
00000050: 4000 4000 0000 0000 4000 4000 0000 0000  @.@.....@.@.....
00000060: f801 0000 0000 0000 f801 0000 0000 0000  ................
00000070: 0800 0000 0000 0000 0300 0000 0400 0000  ................
00000080: 3802 0000 0000 0000 3802 4000 0000 0000  8.......8.@.....
00000090: 3802 4000 0000 0000 1c00 0000 0000 0000  8.@.............
000000a0: 1c00 0000 0000 0000 0100 0000 0000 0000  ................
000000b0: 0100 0000 0500 0000 0000 0000 0000 0000  ................
000000c0: 0000 4000 0000 0000 0000 4000 0000 0000  ..@.......@.....
000000d0: 4408 0000 0000 0000 4408 0000 0000 0000  D.......D.......
000000e0: 0000 2000 0000 0000 0100 0000 0600 0000  .. ............
000000f0: 100e 0000 0000 0000 100e 6000 0000 0000  ..........[crayon-5aa822a2caea8795343825  ].....
00000100: 100e 6000 0000 0000 3002 0000 0000 0000  ..
```

.....0.....
00000110: 3802 0000 0000 0000 0000 2000 0000 0000 8..... .....
00000120: 0200 0000 0600 0000 280e 0000 0000 0000 ....(.......
00000130: 280e 6000 0000 0000 280e 6000 0000 0000 (.

```
.....(.
```

.....
00000140: d001 0000 0000 0000 d001 0000 0000 0000 ....
00000150: 0800 0000 0000 0000 0400 0000 0400 0000 ....
00000160: 5402 0000 0000 0000 5402 4000 0000 0000 T.......T.@.....
00000170: 5402 4000 0000 0000 4400 0000 0000 0000 T.@.....D.......
00000180: 4400 0000 0000 0000 0400 0000 0000 0000 D...............
00000190: 50e5 7464 0400 0000 f406 0000 0000 0000 P.td............
000001a0: f406 4000 0000 0000 f406 4000 0000 0000 ..@.......@.....
000001b0: 3c00 0000 0000 0000 3c00 0000 0000 0000 <.......<.......
000001c0: 0400 0000 0000 0000 51e5 7464 0600 0000 ....Q.td....
000001d0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
000001e0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
000001f0: 0000 0000 0000 0000 1000 0000 0000 0000 ....
00000200: 52e5 7464 0400 0000 100e 0000 0000 0000 R.td............
00000210: 100e 6000 0000 0000 100e 6000 0000 0000 ..

```
.......
```

.....
00000220: f001 0000 0000 0000 f001 0000 0000 0000 ....
00000230: 0100 0000 0000 0000 2f6c 6962 3634 2f6c ..../lib64/l
00000240: 642d 6c69 6e75 782d 7838 362d 3634 2e73 d-linux-x86-64.s
00000250: 6f2e 3200 0400 0000 1000 0000 0100 0000 o.2.............
00000260: 474e 5500 0000 0000 0200 0000 0600 0000 GNU.............
00000270: 2000 0000 0400 0000 1400 0000 0300 0000 ....
00000280: 474e 5500 214c 233f 38e5 451a c86c 6ef1 GNU.!L#?8.E..ln.
00000290: c481 899c d15f 04f7 0200 0000 0500 0000 ....._..........
000002a0: 0100 0000 0600 0000 0080 0000 0000 8000 ....
000002b0: 0000 0000 0500 0000 cf4d 76d3 0000 0000 ....Mv.....
000002c0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
000002d0: 0000 0000 0000 0000 1300 0000 1200 0000 ....
000002e0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
000002f0: 1800 0000 1200 0000 0000 0000 0000 0000 ....
00000300: 0000 0000 0000 0000 2900 0000 1200 0000 ....).......
00000310: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000320: 3b00 0000 2000 0000 0000 0000 0000 0000 ;... ....
00000330: 0000 0000 0000 0000 0b00 0000 1200 0e00 ....
00000340: b605 4000 0000 0000 2500 0000 0000 0000 ..@.....%.......
00000350: 006c 6962 632e 736f 2e36 0063 6f6e 6e65 .libc.so.6.conne
00000360: 6374 0070 7574 7300 5f5f 7374 6163 6b5f ct.puts.__stack_
00000370: 6368 6b5f 6661 696c 005f 5f6c 6962 635f chk_fail.__libc_
00000380: 7374 6172 745f 6d61 696e 005f 5f67 6d6f start_main.__gmo
00000390: 6e5f 7374 6172 745f 5f00 474c 4942 435f n_start__.GLIBC_
000003a0: 322e 3400 474c 4942 435f 322e 322e 3500 2.4.GLIBC_2.2.5.
000003b0: 0000 0200 0300 0200 0000 0100 0000 0000 ....
000003c0: 0100 0200 0100 0000 1000 0000 0000 0000 ....
000003d0: 1469 690d 0000 0300 4a00 0000 1000 0000 .ii.....J.......
000003e0: 751a 6909 0000 0200 5400 0000 0000 0000 u.i.....T.......
000003f0: f80f 6000 0000 0000 0600 0000 0400 0000 ..

```
.............
00000400: 0000 0000 0000 0000 1810 6000 0000 0000  ..........
```

.....
00000410: 0700 0000 0100 0000 0000 0000 0000 0000 ....
00000420: 2010 6000 0000 0000 0700 0000 0200 0000 .

```
.............
00000430: 0000 0000 0000 0000 2810 6000 0000 0000  ........(.
```

.....
00000440: 0700 0000 0300 0000 0000 0000 0000 0000 ....
00000450: 4883 ec08 488b 059d 0b20 0048 85c0 7405 H...H.... .H..t.

00000460: e84b 0000 0048 83c4 08c3 0000 0000 0000 .K...H..........
00000470: ff35 920b 2000 ff25 940b 2000 0f1f 4000 .5.. ..%.. ...@.
00000480: ff25 920b 2000 6800 0000 00e9 e0ff ffff .%.. .h.........
00000490: ff25 8a0b 2000 6801 0000 00e9 d0ff ffff .%.. .h.........
000004a0: ff25 820b 2000 6802 0000 00e9 c0ff ffff .%.. .h.........
000004b0: ff25 420b 2000 6690 0000 0000 0000 0000 .%B. .f........
000004c0: 31ed 4989 d15e 4889 e248 83e4 f050 5449 1.I..^H..H...PTI
000004d0: c7c0 c006 4000 48c7 c150 0640 0048 c7c7 ....@.H..P.@.H..
000004e0: db05 4000 e8b7 ffff fff4 660f 1f44 0000 ..@.......f..D..
000004f0: b847 1060 0055 482d 4010 6000 4883 f80e .G.

```
.UH-@.
```

.H...
00000500: 4889 e576 1bb8 0000 0000 4885 c074 115d H..v......H..t.]
00000510: bf40 1060 00ff e066 0f1f 8400 0000 0000 .@.

```
...f........
00000520: 5dc3 0f1f 4000 662e 0f1f 8400 0000 0000  ]...@.f.........
00000530: be40 1060 0055 4881 ee40 1060 0048 c1fe  .@.
```

.UH..@.

```
.H..
00000540: 0348 89e5 4889 f048 c1e8 3f48 01c6 48d1  .H..H..H..?H..H.
00000550: fe74 15b8 0000 0000 4885 c074 0b5d bf40  .t......H..t.].@
00000560: 1060 00ff e00f 1f00 5dc3 660f 1f44 0000  .
```

......].f..D..
00000570: 803d c90a 2000 0075 1155 4889 e5e8 6eff .=.. ..u.UH...n.
00000580: ffff 5dc6 05b6 0a20 0001 f3c3 0f1f 4000 ..].... ......@.
00000590: bf20 0e60 0048 833f 0075 05eb 930f 1f00 . .

```
```

```
.H.?.u......
000005a0: b800 0000 0048 85c0 74f1 5548 89e5 ffd0  .....H..t.UH....
000005b0: 5de9 7aff ffff 5548 89e5 4883 ec20 4889  ].z...UH..H.. H.
000005c0: 7df8 4889 75f0 4889 55e8 bfd4 0640 00e8  }.H.u.H.U....@..
000005d0: acfe ffff b801 0000 00c9 c355 4889 e548  ..........UH..H
000005e0: 83ec 3089 7ddc 4889 75d0 6448 8b04 2528  ..0.}.H.u.dH..%(
000005f0: 0000 0048 8945 f831 c048 c745 e8e2 0640  ...H.E.1.H.E...@
00000600: 00c7 45f0 726f 6f74 c645 f400 488d 4df0  ..E.root.E..H.M.
00000610: 488b 45e8 baec 0640 0048 89ce 4889 c7e8  H.E....@.H..H...
00000620: 92ff ffff 84c0 7507 b801 0000 00eb 05b8  ......u.........
00000630: 0000 0000 488b 55f8 6448 3314 2528 0000  ....H.U.dH3.%(..
00000640: 0074 05e8 48fe ffff c9c3 660f 1f44 0000  .t..H.....f..D..
00000650: 4157 4156 4189 ff41 5541 544c 8d25 ae07  AWAVA..AUATL.%..
00000660: 2000 5548 8d2d ae07 2000 5349 89f6 4989   .UH.-.. .SI..I.
00000670: d54c 29e5 4883 ec08 48c1 fd03 e8cf fdff  .L).H...H.......
00000680: ff48 85ed 7420 31db 0f1f 8400 0000 0000  .H..t 1.........
00000690: 4c89 ea4c 89f6 4489 ff41 ff14 dc48 83c3  L..L..D..A...H..
000006a0: 0148 39eb 75ea 4883 c408 5b5d 415c 415d  .H9.u.H...[]A\A]
000006b0: 415e 415f c390 662e 0f1f 8400 0000 0000  A^A_..f.........
000006c0: f3c3 0000 4883 ec08 4883 c408 c300 0000  ....H...H.......
000006d0: 0100 0200 436f 6e6e 6563 7469 6e67 2e2e  ....Connecting..
000006e0: 2e00 3132 372e 302e 302e 3100 746f 6f72  ..127.0.0.1.toor
000006f0: 0000 0000 011b 033b 3800 0000 0600 0000  .......;8.......
00000700: 7cfd ffff 8400 0000 ccfd ffff 5400 0000  |...........T...
00000710: c2fe ffff ac00 0000 e7fe ffff cc00 0000  ................
00000720: 5cff ffff ec00 0000 ccff ffff 3401 0000  \...........4...
00000730: 1400 0000 0000 0000 017a 5200 0178 1001  .........zR..x..
00000740: 1b0c 0708 9001 0710 1400 0000 1c00 0000  ................
00000750: 70fd ffff 2a00 0000 0000 0000 0000 0000  p...*...........
00000760: 1400 0000 0000 0000 017a 5200 0178 1001  .........zR..x..
00000770: 1b0c 0708 9001 0000 2400 0000 1c00 0000  ........$.......
00000780: f0fc ffff 4000 0000 000e 1046 0e18 4a0f  ....@......F..J.
00000790: 0b77 0880 003f 1a3b 2a33 2422 0000 0000  .w...?.;*3$"....
```

```
......
000007c0: 1c00 0000 6400 0000 13fe ffff 6f00 0000 ....d.......o...
000007d0: 0041 0e10 8602 430d 0602 6a0c 0708 0000 .A....C...j.....
000007e0: 4400 0000 8400 0000 68fe ffff 6500 0000 D.......h...e...
000007f0: 0042 0e10 8f02 420e 188e 0345 0e20 8d04 .B....B....E. ..
00000800: 420e 288c 0548 0e30 8606 480e 3883 074d B.(..H.0..H.8..M
00000810: 0e40 720e 3841 0e30 410e 2842 0e20 420e .@r.8A.0A.(B. B.
00000820: 1842 0e10 420e 0800 1400 0000 cc00 0000 .B..B...........
00000830: 90fe ffff 0200 0000 0000 0000 0000 0000 ................
00000840: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000850: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000860: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000870: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000880: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000890: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000008a0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000008b0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000008c0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000008d0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000008e0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
000008f0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000900: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000910: 0000 0000 0000 0000 0000 0000 0000 0000 ................
```

00000920: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000930: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000940: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000950: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000960: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000970: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000980: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000990: 0000 0000 0000 0000 0000 0000 0000 0000 ....
000009a0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
000009b0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
000009c0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
000009d0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
000009e0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
000009f0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000a00: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000a10: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000a20: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000a30: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000a40: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000a50: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000a60: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000a70: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000a80: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000a90: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000aa0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000ab0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000ac0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000ad0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000ae0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000af0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000b00: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000b10: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000b20: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000b30: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000b40: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000b50: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000b60: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000b70: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000b80: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000b90: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000ba0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000bb0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000bc0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000bd0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000be0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000bf0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000c00: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000c10: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000c20: 0000 0000 0000 0000 0000 0000 0000 0000 ....

00000c30: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000c40: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000c50: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000c60: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000c70: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000c80: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000c90: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000ca0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000cb0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000cc0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000cd0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000ce0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000cf0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000d00: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000d10: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000d20: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000d30: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000d40: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000d50: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000d60: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000d70: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000d80: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000d90: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000da0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000db0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000dc0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000dd0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000de0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000df0: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000e00: 0000 0000 0000 0000 0000 0000 0000 0000 ....
00000e10: 9005 4000 0000 0000 7005 4000 0000 0000 ..@.....p.@.....
00000e20: 0000 0000 0000 0000 0100 0000 0000 0000 ....
00000e30: 0100 0000 0000 0000 0c00 0000 0000 0000 ....
00000e40: 5004 4000 0000 0000 0d00 0000 0000 0000 P.@.............
00000e50: c406 4000 0000 0000 1900 0000 0000 0000 ..@.............
00000e60: 100e 6000 0000 0000 1b00 0000 0000 0000 ..

```
.............
00000e70: 0800 0000 0000 0000 1a00 0000 0000 0000  ................
00000e80: 180e 6000 0000 0000 1c00 0000 0000 0000  ..
```

....
00000e90: 0800 0000 0000 0000 f5fe ff6f 0000 0000 ....o....
00000ea0: 9802 4000 0000 0000 0500 0000 0000 0000 ..@.............
00000eb0: 5003 4000 0000 0000 0600 0000 0000 0000 P.@.............
00000ec0: c002 4000 0000 0000 0a00 0000 0000 0000 ..@.............
00000ed0: 6000 0000 0000 0000 0b00 0000 0000 0000

```
................
00000ee0: 1800 0000 0000 0000 1500 0000 0000 0000  ................
00000ef0: 0000 0000 0000 0000 0300 0000 0000 0000  ................
00000f00: 0010 6000 0000 0000 0200 0000 0000 0000  ..
```

```
.............
00000f10: 4800 0000 0000 0000 1400 0000 0000 0000 H...............
00000f20: 0700 0000 0000 0000 1700 0000 0000 0000 ................
00000f30: 0804 4000 0000 0000 0700 0000 0000 0000 ..@.............
00000f40: f003 4000 0000 0000 0800 0000 0000 0000 ..@.............
00000f50: 1800 0000 0000 0000 0900 0000 0000 0000 ................
00000f60: 1800 0000 0000 0000 feff ff6f 0000 0000 ...........o....
00000f70: c003 4000 0000 0000 ffff ff6f 0000 0000 ..@........o....
00000f80: 0100 0000 0000 0000 f0ff ff6f 0000 0000 ...........o....
00000f90: b003 4000 0000 0000 0000 0000 0000 0000 ..@.............
00000fa0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000fb0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000fc0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000fd0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000fe0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00000ff0: 0000 0000 0000 0000 0000 0000 0000 0000 ................
00001000: 280e 6000 0000 0000 0000 0000 0000 0000 (.
```

```
.............
00001010: 0000 0000 0000 0000 8604 4000 0000 0000  ..........@.....
00001020: 9604 4000 0000 0000 a604 4000 0000 0000  ..@.......@.....
00001030: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001040: 4743 433a 2028 5562 756e 7475 2035 2e34  GCC: (Ubuntu 5.4
00001050: 2e30 2d36 7562 756e 7475 317e 3136 2e30  .0-6ubuntu1~16.0
00001060: 342e 3429 2035 2e34 2e30 2032 3031 3630  4.4) 5.4.0 20160
00001070: 3630 3900 002e 7368 7374 7274 6162 002e  609...shstrtab..
00001080: 696e 7465 7270 002e 6e6f 7465 2e41 4249  interp..note.ABI
00001090: 2d74 6167 002e 6e6f 7465 2e67 6e75 2e62  -tag..note.gnu.b
000010a0: 7569 6c64 2d69 6400 2e67 6e75 2e68 6173  uild-id..gnu.has
000010b0: 6800 2e64 796e 7379 6d00 2e64 796e 7374  h..dynsym..dynst
000010c0: 7200 2e67 6e75 2e76 6572 7369 6f6e 002e  r..gnu.version..
000010d0: 676e 752e 7665 7273 696f 6e5f 7200 2e72  gnu.version_r..r
000010e0: 656c 612e 6479 6e00 2e72 656c 612e 706c  ela.dyn..rela.pl
000010f0: 7400 2e69 6e69 7400 2e70 6c74 2e67 6f74  t..init..plt.got
00001100: 002e 7465 7874 002e 6669 6e69 002e 726f  ..text..fini..ro
00001110: 6461 7461 002e 6568 5f66 7261 6d65 5f68  data..eh_frame_h
00001120: 6472 002e 6568 5f66 7261 6d65 002e 696e  dr..eh_frame..in
00001130: 6974 5f61 7272 6179 002e 6669 6e69 5f61  it_array..fini_a
00001140: 7272 6179 002e 6a63 7200 2e64 796e 616d  rray..jcr..dynam
00001150: 6963 002e 676f 742e 706c 7400 2e64 6174  ic..got.plt..dat
00001160: 6100 2e62 7373 002e 636f 6d6d 656e 7400  a..bss..comment.
00001170: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001180: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001190: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000011a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000011b0: 0b00 0000 0100 0000 0200 0000 0000 0000  ................
000011c0: 3802 4000 0000 0000 3802 0000 0000 0000  8.@.....8.......
000011d0: 1c00 0000 0000 0000 0000 0000 0000 0000  ................
000011e0: 0100 0000 0000 0000 0000 0000 0000 0000  ................
000011f0: 1300 0000 0700 0000 0200 0000 0000 0000  ................
00001200: 5402 4000 0000 0000 5402 0000 0000 0000  T.@.....T.......
```

```
................
00001320: 0100 0000 0000 0000 0000 0000 0000 0000 ................
00001330: 4e00 0000 ffff ff6f 0200 0000 0000 0000 N......o........
00001340: b003 4000 0000 0000 b003 0000 0000 0000 ..@.............
00001350: 0c00 0000 0000 0000 0500 0000 0000 0000 ................
00001360: 0200 0000 0000 0000 0200 0000 0000 0000 ................
00001370: 5b00 0000 feff ff6f 0200 0000 0000 0000 [......o........
00001380: c003 4000 0000 0000 c003 0000 0000 0000 ..@.............
00001390: 3000 0000 0000 0000 0600 0000 0100 0000 0...............
000013a0: 0800 0000 0000 0000 0000 0000 0000 0000 ................
000013b0: 6a00 0000 0400 0000 0200 0000 0000 0000 j...............
000013c0: f003 4000 0000 0000 f003 0000 0000 0000 ..@.............
000013d0: 1800 0000 0000 0000 0500 0000 0000 0000 ................
000013e0: 0800 0000 0000 0000 1800 0000 0000 0000 ................
000013f0: 7400 0000 0400 0000 4200 0000 0000 0000 t.......B.......
00001400: 0804 4000 0000 0000 0804 0000 0000 0000 ..@.............
00001410: 4800 0000 0000 0000 0500 0000 1800 0000 H...............
00001420: 0800 0000 0000 0000 1800 0000 0000 0000 ................
00001430: 7e00 0000 0100 0000 0600 0000 0000 0000 ~...............
00001440: 5004 4000 0000 0000 5004 0000 0000 0000 P.@.....P.......
00001450: 1a00 0000 0000 0000 0000 0000 0000 0000 ................
00001460: 0400 0000 0000 0000 0000 0000 0000 0000 ................
00001470: 7900 0000 0100 0000 0600 0000 0000 0000 y...............
00001480: 7004 4000 0000 0000 7004 0000 0000 0000 p.@.....p.......
00001490: 4000 0000 0000 0000 0000 0000 0000 0000 @...............
000014a0: 1000 0000 0000 0000 1000 0000 0000 0000 ................
000014b0: 8400 0000 0100 0000 0600 0000 0000 0000 ................
000014c0: b004 4000 0000 0000 b004 0000 0000 0000 ..@.............
000014d0: 0800 0000 0000 0000 0000 0000 0000 0000 ................
000014e0: 0800 0000 0000 0000 0000 0000 0000 0000 ................
000014f0: 8d00 0000 0100 0000 0600 0000 0000 0000 ................
00001500: c004 4000 0000 0000 c004 0000 0000 0000 ..@.............
00001510: 0202 0000 0000 0000 0000 0000 0000 0000 ................
00001520: 1000 0000 0000 0000 0000 0000 0000 0000 ................
00001530: 9300 0000 0100 0000 0600 0000 0000 0000 ................
00001540: c406 4000 0000 0000 c406 0000 0000 0000 ..@.............
00001550: 0900 0000 0000 0000 0000 0000 0000 0000 ................
00001560: 0400 0000 0000 0000 0000 0000 0000 0000 ................
00001570: 9900 0000 0100 0000 0200 0000 0000 0000 ................
00001580: d006 4000 0000 0000 d006 0000 0000 0000 ..@.............
00001590: 2100 0000 0000 0000 0000 0000 0000 0000 !...............
000015a0: 0400 0000 0000 0000 0000 0000 0000 0000 ................
000015b0: a100 0000 0100 0000 0200 0000 0000 0000 ................
000015c0: f406 4000 0000 0000 f406 0000 0000 0000 ..@.............
000015d0: 3c00 0000 0000 0000 0000 0000 0000 0000 <...............
000015e0: 0400 0000 0000 0000 0000 0000 0000 0000 ................
000015f0: af00 0000 0100 0000 0200 0000 0000 0000 ................
00001600: 3007 4000 0000 0000 3007 0000 0000 0000 0.@.....0.......
00001610: 1401 0000 0000 0000 0000 0000 0000 0000 ................
```

00001620: 0800 0000 0000 0000 0000 0000 0000 0000 ....
00001630: b900 0000 0e00 0000 0300 0000 0000 0000 ....
00001640: 100e 6000 0000 0000 100e 0000 0000 0000 ..

```
..............
00001650: 0800 0000 0000 0000 0000 0000 0000 0000  ................
00001660: 0800 0000 0000 0000 0000 0000 0000 0000  ................
00001670: c500 0000 0f00 0000 0300 0000 0000 0000  ................
00001680: 180e 6000 0000 0000 180e 0000 0000 0000  ..
```

....
00001690: 0800 0000 0000 0000 0000 0000 0000 0000 ....
000016a0: 0800 0000 0000 0000 0000 0000 0000 0000 ....
000016b0: d100 0000 0100 0000 0300 0000 0000 0000 ....
000016c0: 200e 6000 0000 0000 200e 0000 0000 0000 .

```
..... .......
000016d0: 0800 0000 0000 0000 0000 0000 0000 0000  ................
000016e0: 0800 0000 0000 0000 0000 0000 0000 0000  ................
000016f0: d600 0000 0600 0000 0300 0000 0000 0000  ................
00001700: 280e 6000 0000 0000 280e 0000 0000 0000  (.
```

.....(.......
00001710: d001 0000 0000 0000 0600 0000 0000 0000 ....
00001720: 0800 0000 0000 0000 1000 0000 0000 0000 ....
00001730: 8800 0000 0100 0000 0300 0000 0000 0000 ....
00001740: f80f 6000 0000 0000 f80f 0000 0000 0000 ..

```
..............
00001750: 0800 0000 0000 0000 0000 0000 0000 0000  ................
00001760: 0800 0000 0000 0000 0800 0000 0000 0000  ................
00001770: df00 0000 0100 0000 0300 0000 0000 0000  ................
00001780: 0010 6000 0000 0000 0010 0000 0000 0000  ..
```

....
00001790: 3000 0000 0000 0000 0000 0000 0000 0000 0...............
000017a0: 0800 0000 0000 0000 0800 0000 0000 0000 ....
000017b0: e800 0000 0100 0000 0300 0000 0000 0000 ....
000017c0: 3010 6000 0000 0000 3010 0000 0000 0000 0.

```
.....0.......
000017d0: 1000 0000 0000 0000 0000 0000 0000 0000  ................
000017e0: 0800 0000 0000 0000 0000 0000 0000 0000  ................
000017f0: ee00 0000 0800 0000 0300 0000 0000 0000  ................
00001800: 4010 6000 0000 0000 4010 0000 0000 0000  @.
```

.....@.......
00001810: 0800 0000 0000 0000 0000 0000 0000 0000 ....
00001820: 0100 0000 0000 0000 0000 0000 0000 0000 ....

00001830: f300 0000 0100 0000 3000 0000 0000 0000 ....0.......
00001840: 0000 0000 0000 0000 4010 0000 0000 0000 ....@.......
00001850: 3400 0000 0000 0000 0000 0000 0000 0000 4...............
00001860: 0100 0000 0000 0000 0100 0000 0000 0000 ....
00001870: 0100 0000 0300 0000 0000 0000 0000 0000 ....
00001880: 0000 0000 0000 0000 7410 0000 0000 0000 ....t.......
00001890: fc00 0000 0000 0000 0000 0000 0000 0000 ....
000018a0: 0100 0000 0000 0000 0000 0000 0000 0000 ....[/crayon]

Powyższy przykład pokazuje kompilację w trybie „Release” oraz dodatkowo został „strip’nięty” z symboli. Czyli mamy czystą binarkę, a mimo to w liniach 100-114 jesteśmy w stanie zauważyć dane logowania.

Oczywiście wraz ze wzrostem binarki, tak ręczne wyszukiwanie stringów w takiej postaci bywa uciążliwe, to razem z kompilatorem *gcc* powinniśmy otrzymać narzędzie *strings*:

```
$ strings simplelogin
/lib64/ld-linux-x86-64.so.2
!L#?8
libc.so.6
connect
puts
__stack_chk_fail
__libc_start_main
__gmon_start__
GLIBC_2.4
GLIBC_2.2.5
UH-@
root
AWAVA
AUATL
[]A\A]A^A_
Connecting...
127.0.0.1
toor
;*3$"
GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609
.shstrtab
.interp
.note.ABI-tag
.note.gnu.build-id
.gnu.hash
.dynsym
.dynstr
.gnu.version
.gnu.version_r
.rela.dyn
.rela.plt
.init
```

W tym przypadku widzimy, że trzymanie string’ów jako plain-text mija się z celem i musimy zastosować inną strategię.

Lepsze podejście – obfuskacja

Innym rozwiązaniem jest częściowe lub całkowite zaciemnienie tekstu, czy to przy użyciu zewnętrznych narzędzi, czy też napisanemu własnemu kodowi.

Obfuskacja stringów

W ramach przykładu posłużę się gotową binarką z zadania Over the Wire – Leviathan (Level3) (http://overthewire.org/wargames/leviathan/), plik binarny możecie pobrać [stąd]. Co prawda mamy jakieś symbole debugowe, ale nie mamy ani kodu źródłowego, ani hasła zapisanego jako plain text (przynajmniej nic nie pasuje nam na hasło).

```
$ gdb level3
Reading symbols from level3...done.
gdb-peda$ disas main
Dump of assembler code for function main:
---------=[CUT SNIPPET]=---------
   0x080486a9 <+171>:   call   0x804854d <do_stuff>
---------=[CUT SNIPPET]=---------
   0x080486c5 <+199>:   leave
   0x080486c6 <+200>:   ret
End of assembler dump.
gdb-peda$ disas do_stuff
Dump of assembler code for function do_stuff:
   0x0804854d <+0>:     push   ebp
   0x0804854e <+1>:     mov    ebp,esp
   0x08048550 <+3>:     sub    esp,0x128
   0x08048556 <+9>:     mov    eax,gs:0x14
   0x0804855c <+15>:    mov    DWORD PTR [ebp-0xc],eax
   0x0804855f <+18>:    xor    eax,eax
   0x08048561 <+20>:    mov    DWORD PTR [ebp-0x117],0x706c6e73
   0x0804856b <+30>:    mov    DWORD PTR [ebp-0x113],0x746e6972
   0x08048575 <+40>:    mov    WORD PTR [ebp-0x10f],0xa66
   0x0804857e <+49>:    mov    BYTE PTR [ebp-0x10d],0x0
   0x08048585 <+56>:    mov    eax,ds:0x804a03c
   0x0804858a <+61>:    mov    DWORD PTR [esp+0x8],eax
   0x0804858e <+65>:    mov    DWORD PTR [esp+0x4],0x100
   0x08048596 <+73>:    lea    eax,[ebp-0x10c]
   0x0804859c <+79>:    mov    DWORD PTR [esp],eax
   0x0804859f <+82>:    call   0x80483f0 <fgets@plt>
   0x080485a4 <+87>:    lea    eax,[ebp-0x117]
   0x080485aa <+93>:    mov    DWORD PTR [esp+0x4],eax
   0x080485ae <+97>:    lea    eax,[ebp-0x10c]
   0x080485b4 <+103>:   mov    DWORD PTR [esp],eax
   0x080485b7 <+106>:   call   0x80483d0 <strcmp@plt>
```

W powyższej sekcji do znalezienia hasła wystarczyły jedynie 3 linie:

- `0x080486a9 <+171>: call 0x804854d <do_stuff>` wejście do funkcji w której pobierane jest hasło od użytkownika i jest sprawdzana jego poprawność;
- `0x080485b7 <+106>: call 0x80483d0 <strcmp@plt>` funkcja porównująca stringi, jako miejsce w którym należy szukać poprawnego hasła w pamięci;
- `0004| 0xffffd354 --> 0xffffd361 ("snlprintf\n")` zrzut stosu, gdzie było przygotowane hasło.

Zanim przedstawię wnioski chciałbym pokazać jeszcze jedno podejście.

Obfuskacja – hasła jako liczby

Powyższy przykład pokazał, że dzisiejsze narzędzia są „mądre” i potrafią zamienić liczby na tekst ASCII. A co w przypadku gdybyśmy zamienili litery na nie-drukowalne znaki, niezwiązane z żadnym kodowaniem.

Sprawa wydaje się wyglądać nieco lepiej, ponieważ takie wartości na pewno nie pojawią się po użyciu narzędzia strings, a i zaawansowane narzędzia do wyszukiwania tekstu będą miały problem i nie obędzie się bez ręcznej analizy.

Za trywialny przykład może posłużyć nam kod:

```c
```

```c
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>

bool checkkey(int k)
{
  volatile int res = 0xdeadbeef;
  res |= k - 0xdeadbab3;
  int passkey = (k & 0xdeadc0de - k ^ 3 - 229) + 15;
  res ^= 123;
  return !passkey;
}

int main(int argc, char * argv[])
{
  if (argc != 2)
  {
    printf("./%s <passkey>\n", argv[0]);
    return 1;
  }


  if (checkkey(atoi(argv[1])))
  {
    printf("%s\n", "Horray!");
  }
  else
  {
    printf("%s\n", ":(");
  }

  return 0;
}
```

```
$ gdb -q better.exe
Reading symbols from C:\Users\shead\Downloads\better.exe...(no debugging symbols found)...done.
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
   0x0040148b <+0>:     push   ebp
   0x0040148c <+1>:     mov    ebp,esp
   0x0040148e <+3>:     and    esp,0xfffffff0
   0x00401491 <+6>:     sub    esp,0x10
   0x00401494 <+9>:     call   0x4019f0 <__main>
   0x00401499 <+14>:    cmp    DWORD PTR [ebp+0x8],0x2
   0x0040149d <+18>:    je     0x4014bb <main+48>
   0x0040149f <+20>:    mov    eax,DWORD PTR [ebp+0xc]
   0x004014a2 <+23>:    mov    eax,DWORD PTR [eax]
   0x004014a4 <+25>:    mov    DWORD PTR [esp+0x4],eax
   0x004014a8 <+29>:    mov    DWORD PTR [esp],0x40b064
   0x004014af <+36>:    call   0x408c38 <printf>
   0x004014b4 <+41>:    mov    eax,0x1
   0x004014b9 <+46>:    jmp    0x4014f6 <main+107>
   0x004014bb <+48>:    mov    eax,DWORD PTR [ebp+0xc]
   0x004014be <+51>:    add    eax,0x4
   0x004014c1 <+54>:    mov    eax,DWORD PTR [eax]
   0x004014c3 <+56>:    mov    DWORD PTR [esp],eax
   0x004014c6 <+59>:    call   0x408c88 <atoi>
   0x004014cb <+64>:    mov    DWORD PTR [esp],eax
   0x004014ce <+67>:    call   0x401440 <checkkey>
   0x004014d3 <+72>:    test   al,al
   0x004014d5 <+74>:    je     0x4014e5 <main+90>
   0x004014d7 <+76>:    mov    DWORD PTR [esp],0x40b074
   0x004014de <+83>:    call   0x408c30 <puts>
   0x004014e3 <+88>:    jmp    0x4014f1 <main+102>
   0x004014e5 <+90>:    mov    DWORD PTR [esp],0x40b07c
   0x004014ec <+97>:    call   0x408c30 <puts>
```

Śledzenie wartości liczbowych jest znacznie trudniejsze niż tekstowych, a sama metoda ich wyliczenia może być trudna do analizy. Powyższy przykład wykonuje fake'owe operacja które mają dodatkowo utrudnić analizę kodu. Mimo, że na pierwszy rzut oka może być nie widać jaka jest poprawna wartość, to akurat składowych jest na tyle mało można do tego dość szybko dojść (równanie z jedną niewiadomą):

1. Wiemy, że *passkey* na końcu musi być równe 0, bo przed zwróceniem wartości jest robiona negacja tej wartości, a jedyna wartość dla jakiej negacja jest równa *true* to 0
2. Wiemy też że x & y = y oraz x ^ b = c <=> c ^b = a
3. Ostatecznie passkey = 0xdeadbeef

Wnioski

Na podstawie powyższych przykładów jesteśmy w stanie stwierdzić, że nie możemy ufać środowisku na którym uruchamiamy nasz program, bo to użytkownik jest u siebie gospodarzem, a my gośćmi. On ma pełnię władzy, a my tylko pewne przywileje.

Zauważmy też, że obfuskacja jest jedynie odroczeniem momentu gdy atakujący zdobędzie to czego szuka, może zastosowane metody spowolnią go na godzin, dni, miesięcy, ale ostatecznie on i tak „wygra".

Skoro nie możemy ufać środowisku na którym uruchamiamy program, to gdzie należy trzymać dane logowania / informacje, których nie chcemy mu udostępnić? Odpowiedź jest prosta: na maszynie której ufamy – naszej.

Jeszcze lepsze podejście

Uwaga! Poniższy scenariusz wciąż nie jest najlepszy, ani uniwersalny! Ma za zadanie jedynie przedstawić ideę rozwiązania.

Skoro nie możemy zaufać użytkownikowi, to nie wyposażajmy go w narzędzia które mogą nas skrzywdzić. Przede wszystkim musimy zmienić sposób myślenia:

- oczywistością jest to, że nie może dostać danych logowania do roota, niech używa konta które może tylko np dodawać wpisy;
- idąc o krok dalej zauważamy, że apliakcja użytkownika wcale nie musi łączyć się z bazą danych! Niech do komunikacji z bazą używa jedynie gotowych metod;
- będzie komunikował się z dodatkową warstwą w postaci serwera, który będzie wykonywał jego polecenia w przypadku poprawnej autoryzacji;
- całą weryfikację zrzucimy na serwer.

Jak wygląda nowy schemat połączenia (opisowo):

1. Podajemy swoje dane logowania do Programu (P)
2. (P) hashuje hasło (+ ewentualnie login) i wysyła do Serwera (S) wraz z doklejonym loginem jako zwykły tekst
3. (S) sprawdza poprawność poprzez pobranie hasha hasła danego użytkownika i porównanie obu hashy
    1. Jeżeli oba hashe są identyczne, tzn podane hasło jest poprawne to zwraca czasowy token (#), który będzie służył (P) do autoryzacji wykonywanych akcji
    2. Jeżeli się nie zgadzają to zostaje zwrócona informacja o złym loginie
4. Do komunikacji z bazą danych (dodawania nowych rekordów, etc) używane są predefiniowane funkcje wraz z jednoczesnym podaniem (#). Tzn zostaje wysłany komunikat do (S), on sprawdza poprawność żądania i czy użytkownik może wykonać daną akcję poprzez weryfikację (#), następnie (P) wykonuje odpowiednią kwerendę na bazie.

Dodatkowo komunikację należałoby zanurzyć w jakimś algorytmie szyfrującym.

W ten sposób likwidujemy możliwość zdobycia danych logowania na samym etapie analizy kodu binarnego oraz likwidujemy potrzebę umieszczenia tych danych w samym pliku wykonywalnym.

Słowo końcowe

Kończąc ten artykuł należy zauważyć, że sam temat ma jeszcze wiele aspektów które należałoby poruszyć. Tutaj rzuciliśmy okiem na kilka podstawowych problemów oraz przedstawiliśmy sobie propozycję, która do pewnego stopnia likwiduje te problemy.

Zapraszam do dyskusji w komentarzach,

Code ON!