

# Security Bootloader Manager

사용자를 위한, 좀더 깨끗하고, 안전한 세상을 만들고  
안전하지 못한 상황들로 부터 사용자를 보호하기 위한 오픈소스 프로젝트

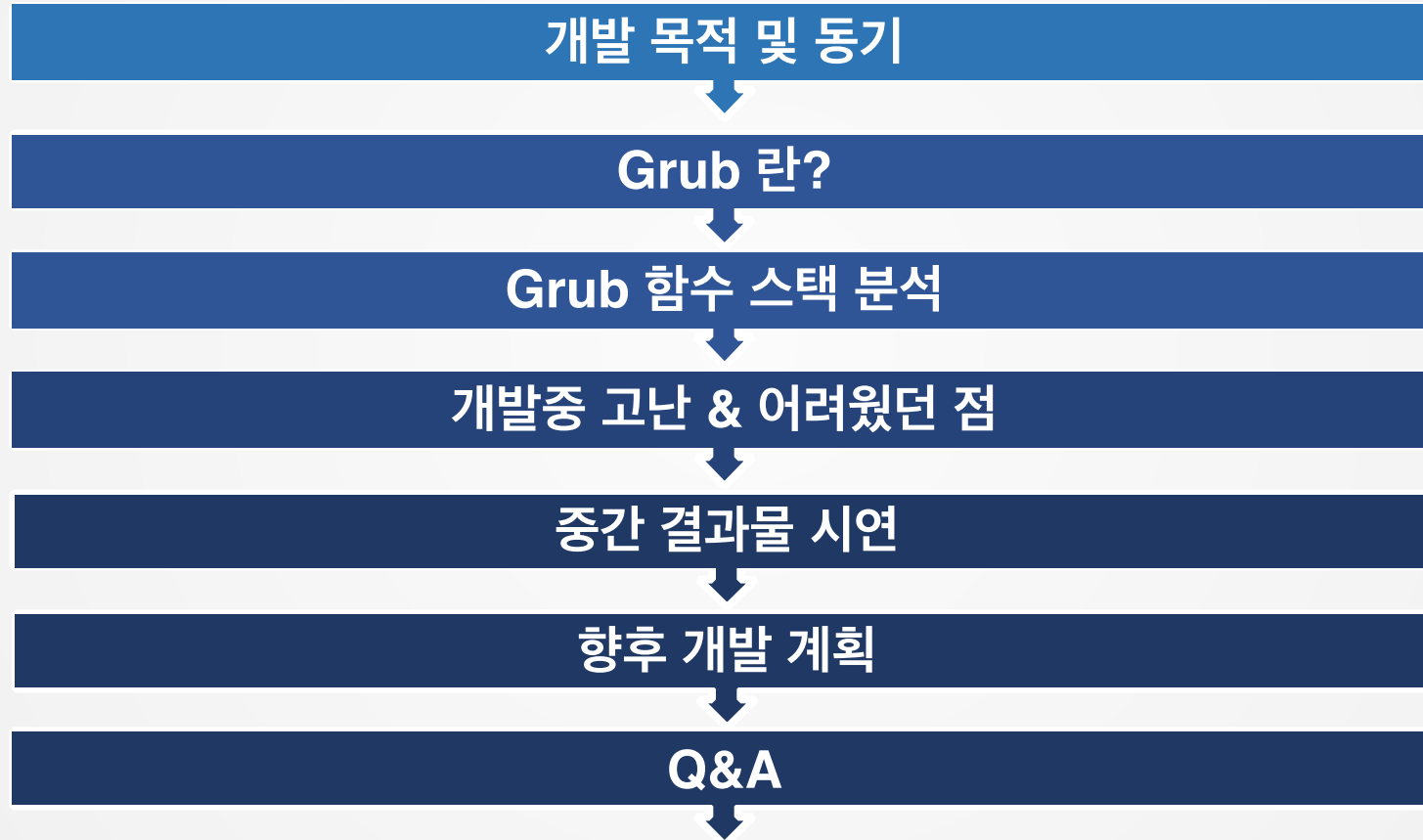


팀 명 : TNTeams

강호용  
김병욱  
김하진  
유주현

# Section 1 목 차

---



## Section 2 개발 목적 및 동기



노트북이 도난 당했을 경우 대부분의 사람들이 범인의 인상착의를 알 수 없다!



자신의 노트북이 자신의 의지와 상관없이 팔려가는 것을 방지할 수는 없을까?

# Section 3 Grub 란?

---

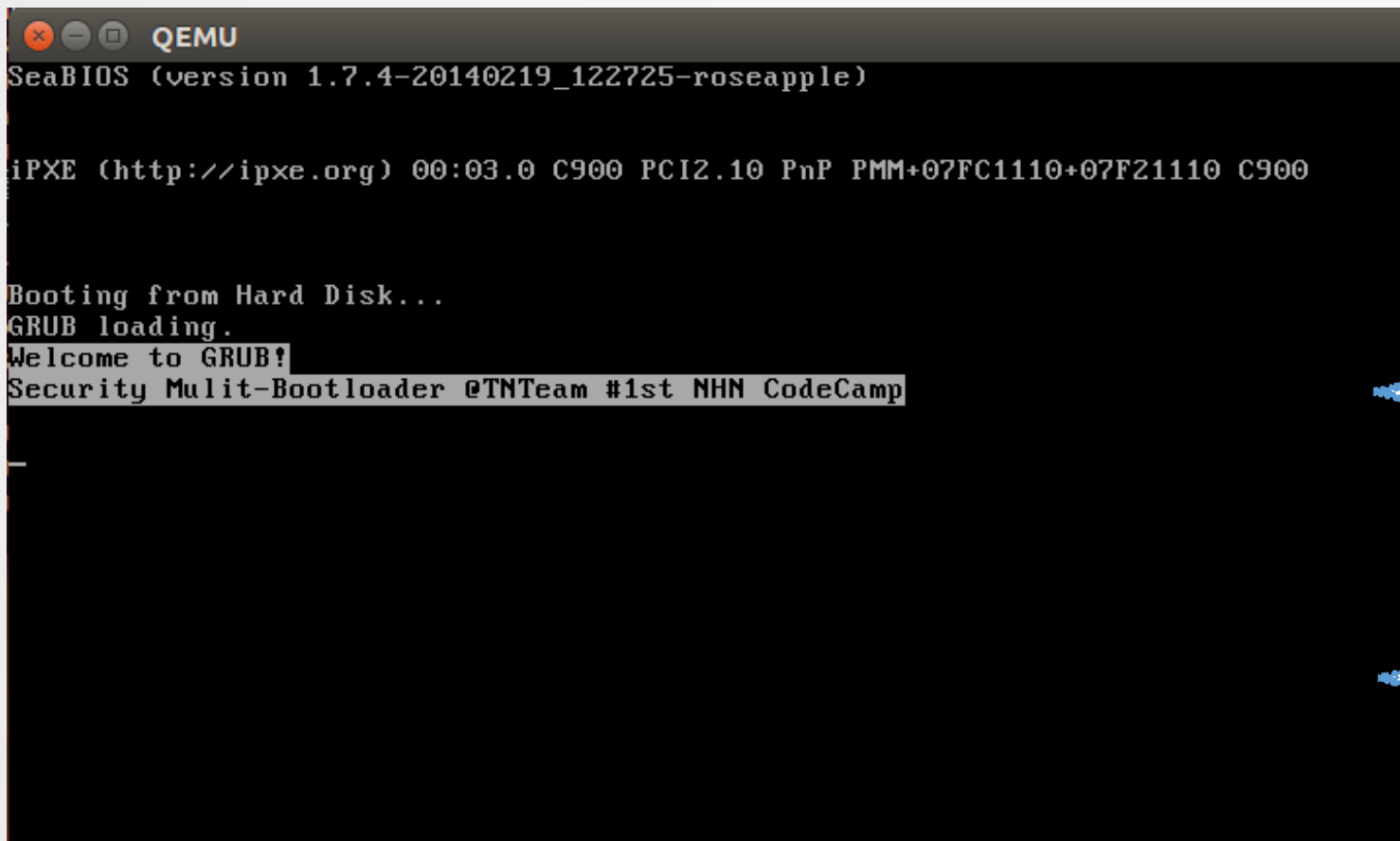


## Grub 란?

리눅스에 한정되지 않고 다른 운영체제에서도 사용가능한 멀티부트로더  
LICENSE : **LGPL 3.0**

GNU Free Software Foundation  
**Grub 2.0**

# Section 4 Grub 함수 호출 스택 분석



```
QEMU
SeaBIOS (version 1.7.4-20140219_122725-roseapple)

iPXE (http://ipxe.org) 00:03.0 C900 PCI2.10 PnP PMM+07FC1110+07F21110 C900

Booting from Hard Disk...
GRUB loading.
Welcome to GRUB!
Security Mulit-Bootloader @TNTeam #1st NHN CodeCamp
```

## QEMU를 이용한 i386 시뮬레이션

- 하드웨어 가상화 시뮬레이션 툴을 이용한 테스트

## Grub main 문구 수정

- grub/grub-core/kern/main.c -> grub\_main()
- grub\_printf() 를 이용한 문구 추가

## Grub Boot Welcome : Sleep(3)

- grub/include/grub/time.h -> grub\_sleep()
- grub\_sleep() 으로 3초간 delay test

# Section 4 Grub 함수 호출 스택 분석

```
264  /* The main routine. */
265  void __attribute__((noreturn))
266  grub_main (void)
267  {
268      /* First of all, initialize the machine. */
269      grub_machine_init ();
270
271      grub_boot_time ("After machine init.");
272
273      /* Hello. */
274      grub_setcolorstate (GRUB_TERM_COLOR_HIGHLIGHT);
275      grub_printf ("Welcome to GRUB!\n");
276      grub_printf ("Security Mult-Bootloader @TNTeam #1st NHN CodeCamp\n\n");
277      grub_setcolorstate (GRUB_TERM_COLOR_STANDARD);
278      ...
315  grub_load_normal_mode ();
316  grub_rescue_run ();
317  }
```

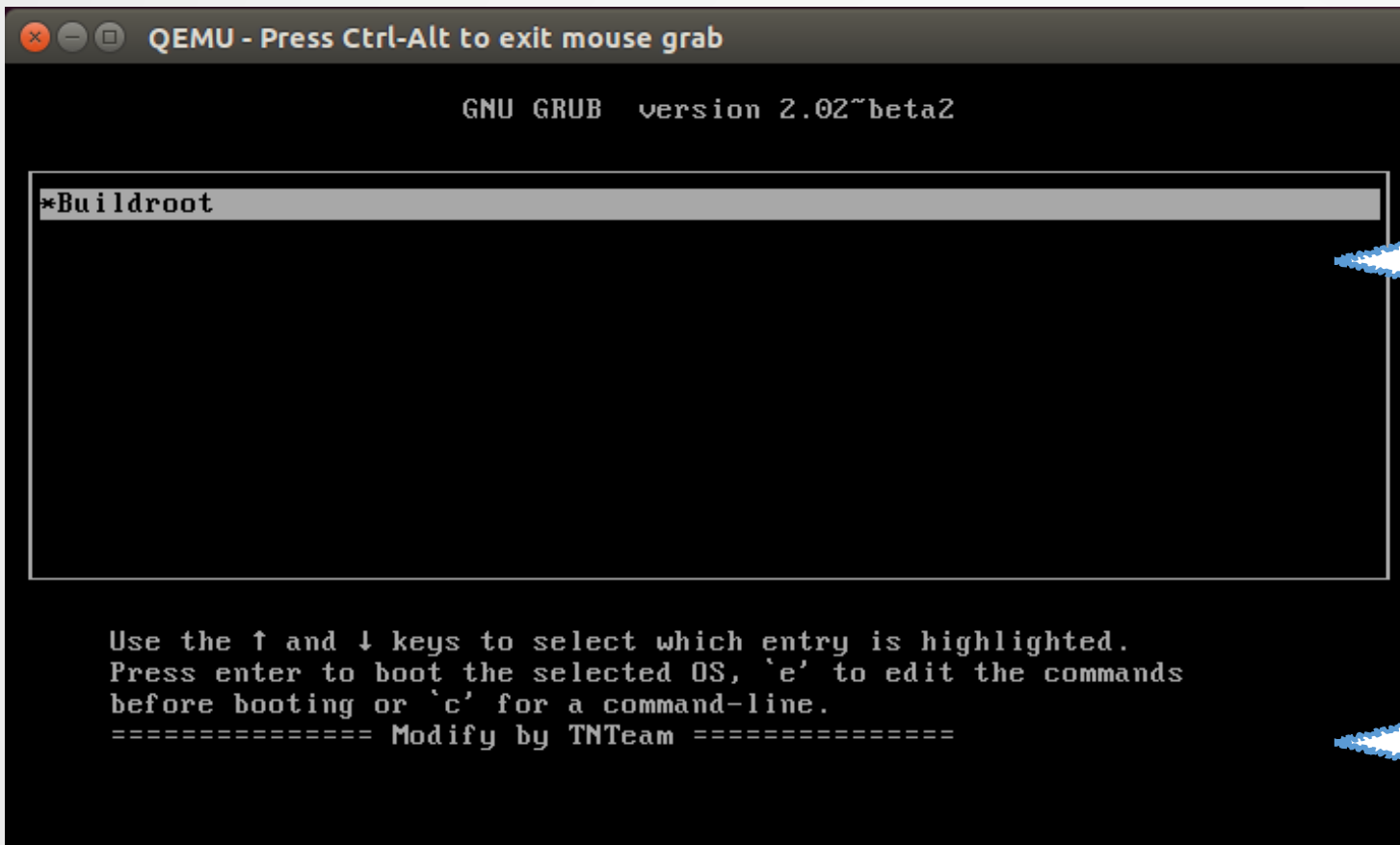
## grub\_printf()

위치 : stdio.h => printf() redirect function  
기능 : Boot Console Video Memory 영역 (0xB8000)에 문자열 복사 => 출력

## grub\_main()

위치 : grub/grub-core/kern/main.c#L265  
기능 : Grub의 초기 시작 지점 및 각 종 모듈 & 장치 초기화를 진행

# Section 4 Grub 함수 호출 스택 분석



```
QEMU - Press Ctrl-Alt to exit mouse grab

GNU GRUB  version 2.02~beta2

*Buildroot

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.
===== Modify by TNTeam =====
```

**/boot/grub/grub.cfg**

- 실제 부팅 가능한 연결된 커널 목록
- grub/grub-core/normal/menu\_entry.c#L1228

**grub\_menu\_init\_page()**

- grub/grub-core/normal/menu\_text.c
- grub\_print\_message\_indented\_real() 문구 추가

# Section 4 Grub 함수 호출 스택 분석

```
329 static grub_err_t
330 grub_dl_resolve_symbols (grub_dl_t mod, Elf_Ehdr *e)
331 {
332     unsigned i;
333     Elf_Shdr *s;
334     Elf_Sym *sym;
335     const char *str;
336     Elf_Word size, entsize;
```

...

```
411     if (bind != STB_LOCAL)
412         if (grub_dl_register_symbol (name, (void *) sym->s
413             return grub_errno;
414     if (grub_strcmp (name, "grub_mod_init") == 0)
415         mod->init = (void *) (grub_dl_t) sym->st_value;
416     else if (grub_strcmp (name, "grub_mod_fini") == 0)
417         mod->fini = (void *) (void) sym->st_value;
418     break;
```

## grub\_mod\_init()

Grub의 경우 각각 서브 모듈이 elf file format 으로 되어 있으므로 이를 바이너리 분석을 통해 각 모듈의 초기화 부분의 함수 포인터를 얻어옴

## grub\_mod\_fini()

바이너리 분석을 통해 각 모듈의 종료시 호출되어야하는 처리 부분의 함수 포인터를 얻어옴



# Section 4 Grub 함수 호출 스택 분석

```
332 grub_menu_init_page (int nested, int edit,  
333                      struct grub_term_screen_geometry *geo,  
334                      struct grub_term_output *term)  
335 {  
336     grub_uint8_t old_color_normal, old_color_highlight;
```

...

```
346     geo->first_entry_y = 2 /* two empty lines*/  
347       + 1 /* GNU GRUB version text */ + 1 /* top border */;  
348  
349     geo->timeout_lines = 2;  
350  
351     /* 3 lines for timeout message and bottom margin. 2 lines for the  
352     geo->num_entries = grub_term_height (term) - geo->first_entry_y  
353       - 1 /* bottom border */  
354       - 1 /* empty line before info message*/  
355       - geo->timeout_lines /* timeout */  
356       - 1 /* empty final line */;  
357     msg_num_lines = print_message (nested, edit, term, 1);
```

## grub\_term\_height()

Grub GUI 화면에서 직선을 그리는 함수로 굵기 및 길이 등을 지정하여 화면에 렌더링이 가능하다

## print\_message()

각 모드 별로 출력될 메인 메시지들에 대한 char\* 를 리턴하는 함수  
해당 함수를 이용하여 안내 문구를 추가, 수정이 가능하다.

## Section 4 Grub 함수 호출 스택 분석

```
152 static int
153 print_message (int nested, int edit, struct grub_term_output *term, int dry_run)
154 {
155     int ret = 0;
156     grub_term_setcolorstate (term, GRUB_TERM_COLOR_NORMAL);
157
158     if (edit)
159     {
160         ret += grub_print_message_indented_real (_("Minimum Emacs-like screen editing is \
161 supported. TAB lists completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a \
162 command-line or ESC to discard edits and return to the GRUB menu."),
```

...

```
199     ret += grub_print_message_indented_real
200         (_("==== Modify by TNTeam ====="),
201         STANDARD_MARGIN, STANDARD_MARGIN, term, dry_run);
202
203     return ret;
204 }
```

### grub\_print\_message\_indented\_real()

UCS4(UTF32) 형식의 문자열을 UTF8 형식으로 인코딩 하여 해당 라인수 만큼 화면에 출력하는 함수

# Section 7 향후 개발 계획

