Some arbitrary address read vulnerabilities in readelf #244

Closed · liyansong2018 opened this issue on Jun 10 · 0 comments

liyansong2018 commented on Jun 10 • edited

Hi,

there are many out-of-bounds read leading to possible temporary denial of service in readelf.

PoC

poc_elf_out_of_bounds.zip

```
./readelf -a poc_elf_out_of_bounds
ELF Header:
...
Program Headers:
  Type           Offset             VirtAddr           PhysAddr
                 FileSiz            MemSiz              Flags  Align
  PHDR           0x0000000000000040 0x0000000000000040 0x0000000000000040
                 0x00000000000002d8 0x00000000000002d8  R      0x8
...

Relocation section '' at offset 0x200000007 contains 159629617834 entries.
  Offset          Info           Type           Sym. Value    Sym. Name + Addend
zsh: segmentation fault  ./readelf -a poc_elf_out_of_bounds
```

In fact, when printing external data in%s format, `readelf` need to judge the legitimacy of the address, which cannot exceed the range of the ELF file.

```
git diff
diff --git a/apps/readelf.c b/apps/readelf.c
index ce25d5e1..5832f88f 100644
--- a/apps/readelf.c
+++ b/apps/readelf.c
@@ -670,9 +670,11 @@ int main(int argc, char * argv[]) {
                                        break;
                                case SHT_RELA:
                                        if (show_bits & SHOW_RELOCATIONS) {
```

```
-                                        printf("\nRelocation section '%s' at offset 0x%lx con
+                                        if (is_valid(stringTable + sectionHeader.sh_name)) {
+                                               printf("\nRelocation section '%s' at offset 0
                                                stringTable + sectionHeader.sh_name, sectionH
                                                sectionHeader.sh_size / sizeof(Elf64_Rela));
+                                        }
                                        printf("  Offset          Info           Type

                                        /* Section this relocation is in */
```

klange closed this as completed in 5d36d27 on Aug 17

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant