ivd38 / exim_invalid_free Public

GPL-3.0 license

6 stars 2 forks

Star Notifications

Code Issues Pull requests Actions Projects Security Insights

main Go to file

ivd38 Add files via upload ... on Aug 19 12

View code

README.md

[CVE-2022-37451] Exim 4.95 invalid free

Silently fixed in Exim 4.96 - https://github.com/Exim/exim/commit/51be321b27825c01829dffd90f11bfff256f7e42

```
static int
pam_converse (int num_msg, PAM_CONVERSE_ARG2_TYPE **msg,
  struct pam_response **resp, void *appdata_ptr)
{
...
    if (  pam_arg_ended
   || !(reply = malloc(sizeof(struct pam_response) * num_msg)))
  return PAM_CONV_ERR;

for (int i = 0; i < num_msg; i++)
  {
  uschar *arg;
  switch (msg[i]->msg_style)
    {
    case PAM_PROMPT_ECHO_ON:
    case PAM_PROMPT_ECHO_OFF:
      if (!(arg = string_nextinlist(&pam_args, &sep, NULL, 0)))
        {
        arg = US"";
        pam_arg_ended = TRUE;
```

```
        }
[1]      reply[i].resp = CS string_copy_malloc(arg); /* PAM frees resp */
      reply[i].resp_retcode = PAM_SUCCESS;
      break;

...
}

uschar *
string_copy_malloc(const uschar *s)
{
int len = Ustrlen(s) + 1;
[2] uschar *ss = store_malloc(len);
memcpy(ss, s, len);
return ss;
}

static void *
internal_store_malloc(int size, const char *func, int line)
{
void * yield;

if (size < 0 || size >= INT_MAX/2)
  log_write(0, LOG_MAIN|LOG_PANIC_DIE,
            "bad memory allocation requested (%d bytes) at %s %d",
            size, func, line);

size += sizeof(int);    /* space to store the size, used under debug */
if (size < 16) size = 16;

if (!(yield = malloc((size_t)size)))
  log_write(0, LOG_MAIN|LOG_PANIC_DIE, "failed to malloc %d bytes of memory: "
    "called from line %d in %s", size, line, func);

[3] DEBUG(D_any) *(int *)yield = size;
[4] yield = US yield + sizeof(int);
...
return yield;
```

Note on line #1 reply[i].resp is allocated using store_malloc().

store_malloc() allocates size+4 bytes using malloc(), and returns ptr + 4 (see line #4), i.e. store_malloc() should be matched with store_free().

In our case reply[i].resp will be freed by libpam using free().

How to reproduce (compile from src):

```
1. Build exim with asan

Enable SUPPORT_PAM=yes and AUTH_PLAINTEXT=yes

2. Create /etc/pam.d/exim
auth            required       pam_unix.so
account         required       pam_permit.so
session         required       pam_permit.so

3. Edit exim.conf:
begin authenticators
PLAIN:
   driver = plaintext
   server_prompts = :
   # Check password in $3 for user in $2
   server_condition = "${if pam{$auth2:${sg{$auth3}{:}{::}}}}"
   server_set_id = $auth2

LOGIN:
   driver = plaintext
   server_prompts = Username:: : Password::
   # Check password in $2 for user in $1
   server_condition = "${if pam{$auth1:${sg{$auth2}{:}{::}}}}"
   server_set_id = $auth1



4. Run exim:
#./build-Linux-x86_64/exim -bd -d

5. Run test script
$ ./t1.py localhost
```

Asan log attached.

There is another store_malloc()/free() mismatch in Exim 4.96 spf.c:SPF_dns_exim_new(), but it seems like it is not exploitable at all.

## Releases

No releases published

## Packages

No packages published

## Languages

- **Python** 100.0%