belkin F9K1009_WW_2.00.09 hardcoded credential

product

vendor: belkin

product: F9K1009

version: up to F9K1009_WW_2.00.09

support url: https://www.belkin.com/support-article/?articleNum=109400

Description

In belkin F9K1009_WW_2.00.09, hard-coded credential on the Web Interface allows anyone to log in to the firmware directly to perform administrative functions. Malicious attacker can reverse the firmware and use hard-coded credential with username '00E0A6-111' and password '00E0A6-111' for authentication.

details

In function `0x4042E0` of the web service of the firmware, which is `mini_httpd`, the following code handles authentication. The following code reads credential stored inside firmware

```
    v13 = (const char *)env_getvar("ConnectionRequestUsername");// get firmware username
    dword_4505DC = (int)v13;
    if ( !safestrcmp(v31, "1") )
    {
      if ( !v13 )
      {
        syslog(3, "TR069Mini_httpd: Error reading the user name or user name not set\nUser - %s\n", 0);
        goto LABEL_38;
      }
      if ( safestrcmp(v13, v30) )
      {
        syslog(4, "TR069Mini_httpd: Unauthorised user\nUser - %s\n", v13);
        sub_4041B8();
      }
    }
    sys_passwd = (_BYTE *)env_getvar("ConnectionRequestPassword");// get firmware password
    if ( sys_passwd )
      goto do_digest;
    syslog(3, "TR069Mini_httpd: Error reading the password or password not set\n");
LABEL_38:
    exit(1);
  }
```

The read credential is then send into function `CalcDigest` to do digest calculation. Note that in the following code, if user's username is "00E0A6-111", then sys_passwd will be automatically replaced with static value "00E0A6-111"

```
    if ( !dword_45180C )
      unauthorized();
    if ( safestrncmp(dword_45180C, "Digest ", 7) )
      unauthorized();
    input_username = sub_402450(dword_45180C, "username=");
    v27 = sub_402450(dword_45180C, "realm=");
    v25 = sub_402450(dword_45180C, "nonce=");
    v29 = sub_402450(dword_45180C, "uri=");
    v26 = sub_402450(dword_45180C, "qop=");
    user_response = sub_402450(dword_45180C, "response=");
    v23 = sub_402450(dword_45180C, "cnonce=");
    v24 = sub_402450(dword_45180C, "nc=");
    if ( !safestrcmp(input_username, "00E0A6-111") )
    {
      v6 = malloc(256);
      dword_4505DC = v6;
      if ( !v6 )
        goto LABEL_22;
      memset(v6, 0, 256);
      v7 = (_BYTE *)dword_4505DC;
      strcpy(v32, "11");
      *(_BYTE *)(dword_4505DC + 10) = a00e0a6111[10];
      qmemcpy(v7, "00E0A6-1", 8);
      v7[8] = v32[0];
      v7[9] = v32[1];
      v8 = malloc(256);
      sys_passwd = (_BYTE *)v8;
      if ( !v8 )
      {
LABEL_22:
        syslog(3, "TR069Mini_httpd: Malloc Failure\n");
        goto LABEL_38;
      }
      memset(v8, 0, 256);
      qmemcpy(sys_passwd, "00E0A6-1", 8);
      sys_passwd[8] = v32[0];
      sys_passwd[9] = v32[1];
      sys_passwd[10] = v32[2];
do_digest:
      CalcDigest(input_username, sys_passwd, v27, v29, v25, v26, v24, v23, dword_45182C, &calculated_res);
      free(input_username);
      free(dword_4505DC);
      free(sys_passwd);
      free(v27);
      free(v29);
      free(v25);
      free(v23);
      free(v26);
      free(v24);
      free(0);
      if ( dword_45182C )
        free(dword_45182C);
      if ( !safestrcmp(v31, "1") && safestrcmp(calculated_res, user_response) )
      {
```

if user input username is "00E0A6-111",
then sys_password will become "00E0A6-111"

read from system, or from backdoor

calculation output

The login procedure takes `calculated_res` from CalcDigest's result, and compares it against with user;s input in the `response` field

```
          CalcDigest(local_38,puVar8,local_44,local_3c,local_4c,local_48,local_50,local_54,
                     DAT_0045182c,&result);
          free(local_38);
          free(DAT_004505dc);
          free(puVar8);
          free(local_44);
          free(local_3c);
          free(local_4c);
          free(local_54);
          free(local_48);
          free(local_50);
          free((void *)0x0);
          if (DAT_0045182c != 0) {
            free((void *)DAT_0045182c);
          }
          iVar7 = safestrcmp(local_34,PTR_DAT_00450270 + -0x1f00);
          if ((iVar7 == 0) && (iVar7 = safestrcmp(result,response_field), iVar7 != 0)) {
            syslog(4,PTR_DAT_00450270 + -0x1e0c);
            unauthorized();
          }
```

Attackers can effectively guess the calculation output from the hard-coded username and password and use the hard-coded credential to log into the firmware.

Note that CalcDigest (in libssap.so) uses md5 to hash user inputs. Since all inputs are known under this scenario, attackers can easily guess the right digest result.

timeline

[05/08/2025] report to vendor