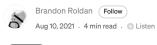
Openinapp 7

Sign up Sign In







Hacking the Tenda AC10-1200 Router Part 3: Yet Another Buffer Overflow

Hi. This is my third writeup in my hacking the tenda ac10 series where i try to get a cve. Lets get started.

So while looking through the functions that accept user inputs, i found this one function called $\label{eq:final_pmacbind} \textit{fromSetIpMacBind}$

Here's what it do, first it get the value of the parameter list then store its value to the variable called var_3f8_1

```
Lw $v0, -0x7500($gp) {data_53e930}
addiu $a1, $v0, -0x6a40 {data_5195c0, "list"}
lw $v0, -0x7500($gp) {data_53e930}
addiu $a2, $v0, -0x6a38 {0x5195c8}
lw $v0, -0x7c84($gp) {websGetVar} {data_53e1ac}
move $t9, $v0 {websGetVar}
jalr $t9 {websGetVar}
nop
lw $gp, 0x18($fp) {var_418}
sw $v0, 0x38($fp) {var_3f8_1}
```

Then, it move this var_3f8_1 variable to another variable var_40c_1

```
lw $v0, 0x38($fp) {var_3f8_1}
sw $v0, 0x24($fp) {var_40c_1}
addiu $v0, $zero, 1
sw $v0, 0x20($fp) {var_410_1} {0x1}
b 0x4a7550
nop
```

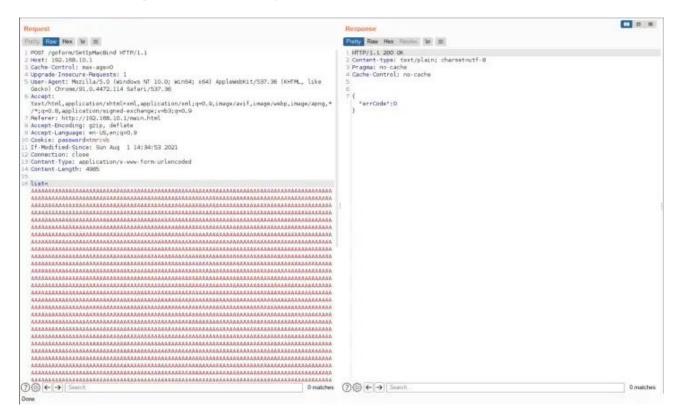
Then check if 0xa is in var_40c_1 using strchr which doesnt matter since either way, there will be a strcpy which will cause the buffer overflow.



Now lets try it out. fromSetIpMacBind is referenced in formDefineTenda

```
lw $v0, -0x7ae4($gp) {data_53e34c}
addiu $a0, $v0, -0x6150 {data_509eb0, "SetIpMacBind"}
lw $a1, -0x7818($gp) {fromSetIpMacBind} {data_53e618}
lw $v0, -0x7c88($gp) {websFormDefine} {data_53e1a8}
move $t9, $v0 {websFormDefine}
jalr $t9 {websFormDefine}
```

This means our vulnerable endpoint is SetIpMacBind. Now lets try it out



It doesnt work. For some reason, it didnt reach our strcpy. Lets track out why.

```
lw $v1, 0x20($fp) {var_410_1}
lw $v0, 0x2c($fp) {var_404_1}
slt $v0, $v0, $v1
beqz $v0, 0x4a724c
nop
```

This is the code before the strchr that we expected. Here, we can see that it checks if var_404_1 is less than var_410_1. If it is, it will jump somewhere else and will not execute the strchr and strcpy that we expected. Now lets see what is the value of these two. Starting with var_404_1

```
lw $v0, -0x7500($gp) {data_53e930}
addiu $a1, $v0, -0x67dc {data_519824, "bindnum"}
lw $v0, -0x7500($gp) {data_53e930}
addiu $a2, $v0, -0x6a58 {0x5195a8}
lw $v0, -0x7c84($gp) {websGetVar} {data_53e1ac}
move $t9, $v0 {websGetVar}
jalr $t9 {websGetVar}
nop
lw $gp, 0x18($fp) {var_418}
sw $v0, 0x34($fp) {var_3fc_1}
```

```
lw $a0, 0x34($fp) {var_3fc_1}
lw $v0, -0x7098($gp) {atoi}
move $t9, $v0
jalr $t9
nop
lw $gp, 0x18($fp) {var_418}
sw $v0, 0x2c($fp) {var_404_1}
```

We can see that its value is the atol of the bindnum parameter. What atol does is it convert our string input to integer. Now lets find out what is the value of var_410_1. This one is a little complicated

```
lw $v0, 0x2c($fp) {var_404_1}
bltz $v0, 0x4a7200
nop

lw $v0, 0x2c($fp) {var_404_1}
slti $v0, $v0, 0x21
bne $v0, $zero, 0x4a7234
nop

lw $v0, 0x38($fp) {var_3f8_1}
sw $v0, 0x24($fp) {var_40c_1}
addiu $v0, $zero, 1
sw $v0, 0x20($fp) {var_410_1} {0x1}
b 0x4a7550
nop
```

Here, it checks if var_404_1 is not less than zero and less than 0x21. (var_404_1 is the atoi of the bindnum parameter). If yes, it will set the value of var_410_1 to 1 or 0x1. The other variables are not that important. Then, the check happens

```
lw $v1, 0x20($fp) {var_410_1} lw $v0, 0x2c($fp) {var_404_1} slt $v0, $v0, $v1 beqz $v0, 0x4a724c nop
```

If var_404_1 is less than var_410_1, it will not execute our strepy and the exploit will fail. So what we have to do is make var_401_1 to 1, because then, var_404_1 will be greater than var_410_1 which then will execute our strepy.

Now to set var_410_1 to 1, the value of the parameter bindnum should be greater than 0 and less than 0x21 as stated here

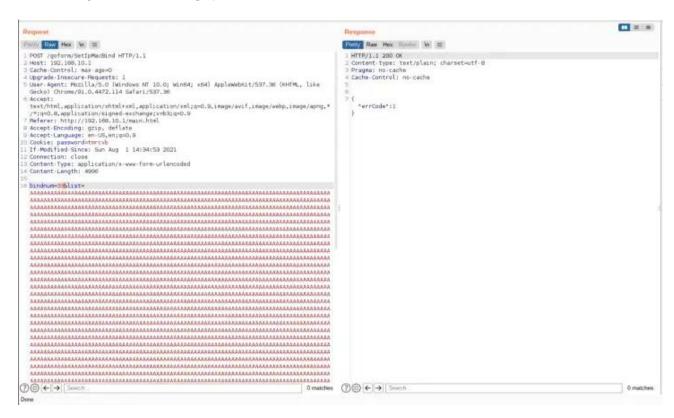
```
lw $v0, 0x2c($fp) {var_404_1}
bltz $v0, 0x4a7200
nop

lw $v0, 0x2c($fp) {var_404_1}
slti $v0, $v0, 0x21
bne $v0, $zero, 0x4a7234
nop

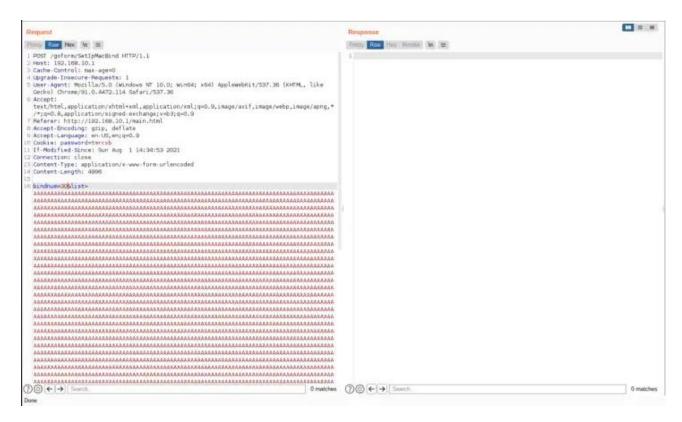
lw $v0, 0x38($fp) {var_3f8_1}
sw $v0, 0x24($fp) {var_40c_1}
addiu $v0, $zero, 1
sw $v0, 0x20($fp) {var_410_1} {0x1}
b 0x4a7550
nop
```

Now lets try it again but this time lets provide the bindnum parameter.

If we set bindnum parameter to 33(0x21) or higher, we can see that it will fail



But if we set it between 1 and 32, it will succeed



No response. That means we crashed the server and our exploit is successful.

We can further confirm it by looking at the emulation

This is the end of the writeup. I tried contacting tenda but they havent responded so i decided to disclose it now.

Thanks for reading

Join the discord server: https://discord.gg/bugbounty

Hacking Infosec Io T Hack Security

About Help Terms Privacy

Get the Medium ann