M10915027 石成峰 HW11

[fastbinlab]

1. 解開Backdoor的鎖需要修改某個記憶體位址，因此需要能夠存取特定記憶體。
2. 由於可以任意設定calloc來的記憶體free後存放fd位置的值，並且calloc會跳過tcache，因此要使用fastbin上的double free攻擊手法。
3. 第一步需要先將讓free後的bin能進到fastbin中才能在calloc時從fastbin拿取，因此需要先將tcache填滿，先free 7個bin進tcache將它填滿。
4. 之後使用交替A->B->A的方式成功double free，此時fastbin中的link狀態為Root->A->B->A-null。
5. 之後calloc大小與A相同的bin後，狀態會變為Root->B->A，將得到的這塊A存放fd位置的記憶體改寫成目標後，fastbin會變為Root->B->A->target->???。
6. 之後再3次calloc後就可以得到目標記憶體，但在這裡需要目標記憶體存放size的值與calloc的大小相同，因此微調target使size的地方為0x23，等同於0x20的大小。
7. 拿到target後算準offset在lock的地方寫入目標數值，之後就可以執行Backdoor了。

[tcachelab]

1. 和fastbinlab類似，但改成malloc因此能夠存取到tcache，並多了edit功能且在delete note後也能使用進行存取。
2. 同樣也能任意修改取得的記憶體fd和bk位置的值，因此可以使用tcache中的double free攻擊。
3. 首先free一次再使用edit修改bk位置也就是key的值，然後再free第2次，成功再tcache中實現double free。tcache的link變為Root->A->A。
4. 同樣的，先把A拿回來並修改fd位置的值變成target，構造tcache變為Root->A->target。
5. 再malloc 2次本以為可以拿到target卻發生了錯誤，經查過資料後發現是tcache中bin的數目對不上所導致，因此步驟3時要多free一次將tcache構造成Root->A->A->A。
6. 之後便可以成功拿到target記憶體，修改lock成目標值後成功執行Backdoor。

[stashlab]

1. 同時可以使用malloc和calloc的功能了，並且只要將目標記憶體修改為非0即可。
2. 但由於這次fd的位置放的是data size，無法任意寫入數值，因此無法使用double free來獲得任意位置的記憶體。
3. 因此要使用的攻擊原理為利用從small bin取出chunk後，small bin會進行整理，將chunk放進tcache後會修改下一個chunk的fd指向small bin的Root。
4. 因此需要構造出的狀況為small bin中已經有一個chunk A，用於取出之後才會進行整理small bin放入tcache的動作。
5. 並且chunk A的bk需要能修改指向chunk B，偽造small bin的link list，而chunk B的fd必須指向chunk A用於通過small bin的檢查，bk則必須指向target會剛好在fd上的位址。
6. 並且tcache必須只差一個空位就滿了，因為如果small bin嘗試把target也送進tcache中會出現位址錯誤。
7. 因此首先要先拿7個chunk然後通通free掉將tcache塞滿。
8. 之後在用calloc繞過tcache拿到新的chunk A然後free掉，這時A會進入unsorted bin，之後在malloc一個更大的chunk使得tcache找不到unsorted bin也找不到，但這時會將chunk A整理進small bin中。
9. 之後用malloc從tcache中拿出一個chunk B，並且讓tcache變成差一個就滿的狀態。
10. 將chunk B的data寫入chunk A的位址(fd)加上target-0x10的位址(bk)，之後在用edit修改chunk A的bk指向chunk B+0x10(data)的位址，就完成我們想要的情境了。
11. 之後用calloc從small bin取出chunk A後，small bin會開始進行整理，將chunk B放入tcache中並把target-0x10的fd(target)寫成small bin的位址，並且會因為tcache已滿而結束不會出現錯誤。
12. target成功修改後通過檢查即可執行Backdoor。