

2. BuildStubThumb

```

Arm mode
shellcode_stub_start:
PUSH    {R0, R1, R2, R3}
.....
MOV     R0, SP
LDR     R3,
_hookstub function_addr
BLX     R3
LDR     R3, _old_function_addr
BIC     R3, R3, #1
ADD     R3, R3, #0x1
STR     R3, _old_function_addr
LDR     R3, [SP, #-0x34] -2
LDR     R0, [SP, #0x3C]
.....
LDR     SP, [R13]
LDR     PC, _old_function_addr+1
  
```

Arm mode
User's hook
stub function

4. RebuildHookTargetThumb

```

Thumb mode
STR     R0, [SP, #0x20+var_10]
STR     R1, [SP, #0x20+var_14]
MOVS    R0, #1
LDR.W   PC, _shellcode_stub_start
_shellcode_stub_start
ADD     R0, R12
STR     R0, [R1]
  
```

```

Thumb mode
old_function_addr:
STR     R0, [SP, #8]
LDR     R0, [SP, #8]
LDR     R1, =(dword_75c1d004-PC-2)
ADD     R1, PC
LDR.W   R12, [R1] (4 bytes)
LDR     PC, _hook_addr+_backup_length
_hook_addr+_backup_length
  
```

3. BuildOldFunctionThumb

```

Thumb mode
old_function_addr:
STR     R0, [SP, #8]
NOP
LDR     R0, [SP, #8]
NOP
LDR     R1, [PC](*)
B       PC+2(**)
dword_75c1d004-origin_PC-2 (4 bytes)(*)
PUSH    {R7}(**)
LDR     R7, PC+6(*)
ADD     R1, R7
POP     {R7}
B       PC+4(**)
NOP
PC(4 bytes)(*)
LDR.W   R12, [R1] (4 bytes)(**)
LDR     PC, _hook_addr+_backup_length
_hook_addr+_backup_length
  
```

1. InitThumbHookInfo

```

Thumb mode
STR     R0, [SP, #0x20+var_10]
STR     R1, [SP, #0x20+var_14]
MOVS    R0, #1
STR     R0, [SP, #8]
LDR     R0, [SP, #8]
LDR     R1, =(dword_75c1d004-PC-2)
ADD     R1, PC
LDR.W   R12, [R1] (4 bytes)
ADD     R0, R12
STR     R0, [R1]
  
```

注：由于PC的CPU三级流水线问题，因此图中标出相同颜色与数量的(*)或(**)代表同一个地址，方便读者查看。如LDR R7, PC+6(*)的PC+6指向PC(4 bytes)(*)。