

Use After Free in mDNSOffloadUserClient.kext

Zhuo Liang
Qihoo 360 Nirvan Team

September 30, 2018

Background

IOKit UserClient classes usually override the method `IOUserClient::clientClose` which can be triggered by `IOServiceClose` from user space. It is just the way of closing handle of `IOUserClient` used by IOKit and is not responsible for resources management. The resources acquired before should be released in the asynchronous method `::free` not rather `::clientClose`. Ian Beer made a clear explanation about this pattern¹ and the root cause was described as follow:

`IOUserClient::clientClose` is not a destructor and plays no role in the lifetime management of an IOKit object.

Vulnerability

It appears that `mDNSOffloadUserClient` in `mDNSOffloadUserClient.kext` does not obey this programming rule.

```
1 __int64 mDNSOffloadUserClient::clientClose(mDNSOffloadUserClient *this)
2 {
3     mDNSOffloadUserClient *v1; // rbx
4     __int64 v2; // rdi
5     __int64 v3; // rax
6     v2 = *((_QWORD *)this + 27);
7     if ( v2 ){
8         ...
9         if ( this->CommandGate ) {
10             v3 = (*(__int64 (__cdecl **)(__QWORD)))(**((_QWORD **)v1 + 27) + 1672LL))
11             (*( (_QWORD *)v1 + 27));
12             if ( v3 )
13                 (*(void (__fastcall **)(__int64, _QWORD))(*(_QWORD *)v3 + 328LL))(v3,
14                 *((_QWORD *)v1 + 28));
15             this->CommandGate->release();
16             this->CommandGate = NULL;
17         }
18     }
```

¹<https://bugs.chromium.org/p/project-zero/issues/detail?id=1377>

```

16 }
17 *((_QWORD *)v1 + 27) = 0LL;
18 return 0LL;
19 }

```

Listing 1: `mDNSOffloadUserClient::clientClose` method

The code of method `mDNSOffloadUserClient::clientClose` is shown in Listing 1. In this method, the `CommandGate` object is released and it will also be freed at once. Notice that we can also trigger another method `mDNSOffloadUserClient::doRequest` (Shown in Listing 2) in which `CommandGate` object is used for synchronization through `IOConnectCallMethod` in another thread.

```

1 __int64 __fastcall mDNSOffloadUserClient::doRequest(mDNSOffloadUserClient *
   this, void *a2, void *a3, __int64 a4, unsigned __int64 *a5) {
2     __int64 result; // rax
3     __int64 v6; // rdi
4     __int64 v7; // [rsp+8h] [rbp-8h]
5
6     v7 = a4;
7     result = 0xE0000001LL;
8     if ( *((_QWORD *)this + 27) ) {
9         if ( this->CommandGate )
10             result = this->CommandGate->runAction(mDNSOffloadUserClient::
               doRequestGated, a2, a3, &v7, a5);
11     }
12     return result;

```

Listing 2: `mDNSOffloadUserClient::doRequest` method

The proof of concept can be found [here](#) and the reproduction steps are as follow:

1. `clang mDNSOffloadUserClientUaF.c -o mdns_uaf -framework IOKit`
2. `while true; do ./mdns_uaf; done`