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 pattern[1] 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.

```
__int64 mDNSOffloadUserClient::clientClose(mDNSOffloadUserClient *this)
{
  mDNSOffloadUserClient *v1; // rbx
  __int64 v2; // rdi
  __int64 v3; // rax
  v2 = *((_QWORD *)this + 27);
  if ( v2 ){
    ...
    if ( this->CommandGate ) {
      v3 = (*(__int64 (__cdecl **)(_QWORD))(**((_QWORD **)v1 + 27) + 1672LL))
    (*((_QWORD *)v1 + 27));
      if ( v3 )
        (*(void (__fastcall **)(__int64, _QWORD))(*(_QWORD *)v3 + 328LL))(v3,
       *((_QWORD *)v1 + 28));
      this->CommandGate->release();
      this->CommandGate = NULL;
    }
```

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

```
  }
  *((_QWORD *)v1 + 27) = 0LL;
  return 0LL;
}
```

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.

```
__int64 __fastcall mDNSOffloadUserClient::doRequest(mDNSOffloadUserClient *this, void *a2, void *a3, __int64 a4, unsigned __int64 *a5) {
  __int64 result; // rax
  __int64 v6; // rdi
  __int64 v7; // [rsp+8h] [rbp-8h]

  v7 = a4;
  result = 0xE0000001LL;
  if ( *((_QWORD *)this + 27) ) {
    if ( this->CommandGate )
      result = this->CommandGate->runAction(mDNSOffloadUserClient::doRequestGated, a2, a3, &v7, a5);
  }
  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