Summary of Critical and Exploitable iOS Vulnerabilities in 2016

阿里聚安全 / 2016-12-29 08:42:35 / 浏览数 2910 安全技术 漏洞分析 顶(0) 踩(0)

Summary of Critical and Exploitable iOS Vulnerabilities in 2016

Author：Min (Spark) Zheng, Cererdlong, Eakerqiu @ Team OverSky

0x00 Introduction

iOS security is far more fragile than you believe. And there are lots of critical and exploitable iOS vulnerabilities in the wild. We summarized these critical iOS vulnerabilities which can be used for remote code execution or jailbreaking in this report. Hopefully, it can bring some help for your mobile security research.

0x01 iOS 10.1.1 Critical and Exploitable Vulnerabilities

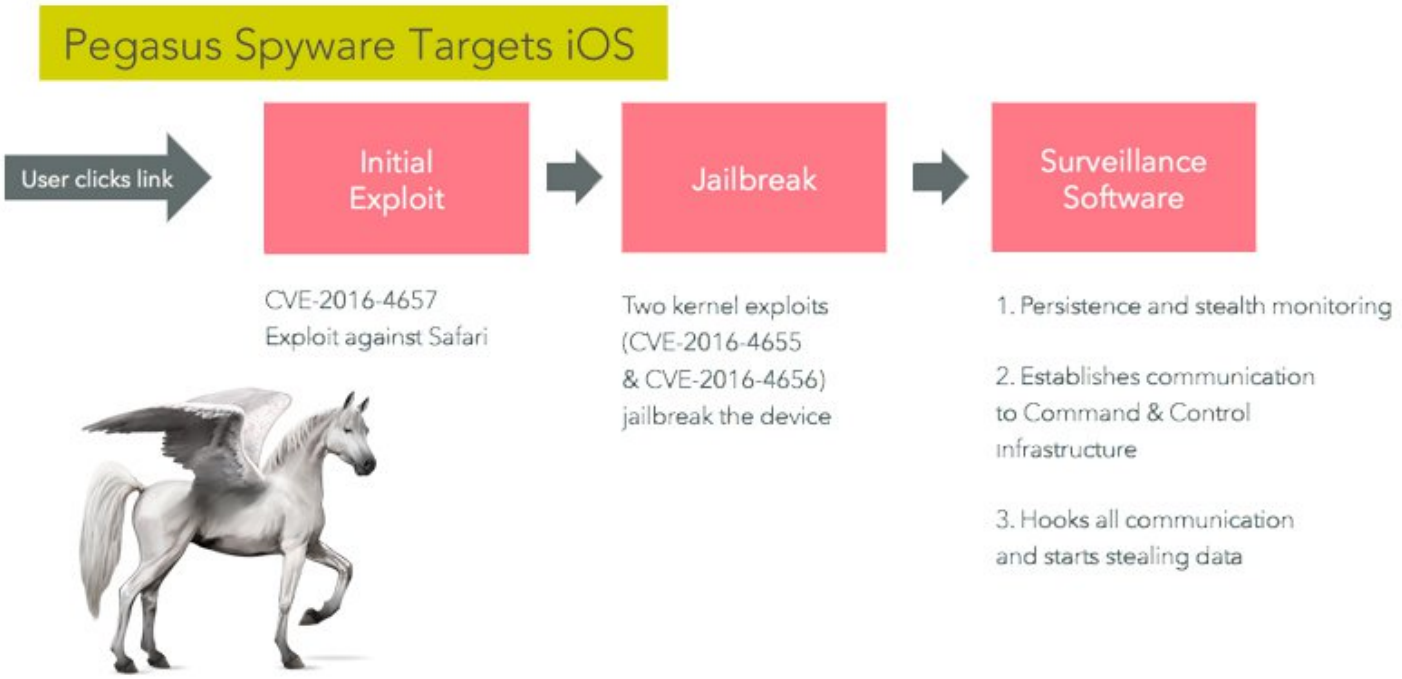
1. Mach_portal exploit chain: The exploit chain was published by Ian Beer of Google Project Zero. The whole exploit chain consists of three vulnerabilities:

CVE-2016-7637: Broken kernel Mach port name uref handling on iOS/MacOS can lead to privileged port name replacement in other processes.
CVE-2016-7661: MacOS/iOS arbitrary port replacement in powerd.
CVE-2016-7644: XNU kernel UaF due to lack of locking in set_dp_control_port.

The attacker first uses CVE-2016-7637 to replace launchd's send right to “com.apple.iohideventsystem” with a send right to a port which belongs to the attacker. The attacker also holds the receive right of that port. Then the attacker uses CVE-2016-7661 to crash the “powerd” daemon which runs as root. Because of the daemon mechanism, the “powerd” will automatically restart but its startup process will look up the “com.apple.iohideventsystem” Mach service and send its own task port to that service. Because the attacker holds the receive right of that port which means the “powerd” actually sends its task port to the attacker. After that, the attacker uses “powerd”'s task port to get the host_priv port which is used to trigger the XNU kernel UaF bug (CVE-2016-7644). Because the kernel forgets to lock the set_dp_control_port when releasing a reference on a port, the attacker can get a send right to the kernel task port. After getting the kernel task port, the attacker can use mach_vm_read() and mach_vm_write() which provided by the XNU system to modify kernel memory.

In 2016.12.22, based on the Beer’s Mach_portal exploit chain, qwertyoruiop added KPP bypass, kernel patch, and Cydia installation on this project. Then he released iOS 10.0./10.1. jailbreak for arm64 devices on yalu.qwertyoruiop.com.

0x02 iOS 9.3.4 Critical and Exploitable Vulnerabilities



1. PEGASUS/Trident exploit chain: The exploit chain was found from an apt issue for a human rights activist. There are three vulnerabilities in the Trident exploit:

CVE-2016-4657: Visiting a maliciously crafted website may lead to arbitrary code execution.
CVE-2016-4655: An application may be able to disclose kernel memory.
CVE-2016-4656: An application may be able to execute arbitrary code with kernel privileges.

For Safari browser, the vulnerability exists within the slowAppend() method of MarkedArgumentBuffer in JavaScriptCore library and can be exploited via the usage of a MarkedArgumentBuffer in the static defineProperties() method. The Pegasus exploit chain triggers this vulnerability by passing a specially crafted sequence of properties to the defineProperties() method and then gets read/write and code execution ability.

For the XNU kernel, the vulnerability exists in the OSUnserializeBinary() method which is used to unserialize the data from the user land input. Because OSUnserializeBinary() doesn't check the length of serialized OSNumber, the attacker can get leaked kernel stack information using io_registry_entry_get_property_bytes(). On the other hand, by using a crafted serialized OSString Object, the attacker can trigger UaF vulnerability in the kernel and then get the read and write ability of the kernel memory.

In addition, by using JavaScriptCore vulnerability, PEGASUS exploit chain can persist after rebooting which means untethered jailbreak. Last but not least, more details about this exploit chain can be referred to our previous article: https://jaq.alibaba.com/community/art/show?articleid=532 and DEMOs:

Youtube: https://www.youtube.com/watch?v=EwRVvUKBSKQ
Youku: http://v.youku.com/v_show/id_XMTg4NzA5OTEwOA==.html

0x03 iOS 9.3.3 Critical and Exploitable Vulnerabilities

1. IOMobileFramebuffer Kernel Heap Overflow: This vulnerability exists in the IOMobileFramebuffer IOKit kernel service. Because IOMobileFramebuffer::swap_submit(IOMFBSwap *) doesn't check the IOMFBSwap data from the user land, the attacker can use a crafted IOMFBSwap data to achieve a heap overflow in the kernel and then translate it into kernel read/write ability. This vulnerability can be triggered in the sandbox (do not need sandbox escapes) and it was used in the Pangu's iOS 9.3.3 jailbreak.

0x04 iOS 9.3.2 Critical and Exploitable Vulnerabilities

WebKit heapPopMin Remote Code Execution: This vulnerability exists in the WebCore ::TimerBase::heapPopMin() and the attacker can use this vulnerability to achieve arability code execution in Safari through a crafted html webpage. Note that the Safari process is sandboxed. So, the attacker needs to do a sandbox escape if he wants to get more user data or attack the kernel.

GasGauge Race Condition: This vulnerability was disclosed by qwertyoruiop. Because GasGauge kernel service doesn't lock the process when it frees the memory, the attacker can use multi-thread to do the race. If the race wins, the vulnerability will cause double free. In addition, the attack can translate it into UaF in any zone and achieve kernel read/write ability. Note that this kernel service cannot be reached in the sandbox. So the attacker needs a sandbox escape before using this vulnerability.

0x05 iOS 9.3.1 Critical and Exploitable Vulnerabilities

1. InpuTbag Heap Overflow: This vulnerability was disclosed by Team OverSky of Alibaba mobile security. The vulnerability exists in the postElementValues() method of IOHIDDevice kernel service. Because the postElementValues() method doesn't check the size of input report, the attacker can use a crafted input report to overflow the kernel heap and then achieve kernel read/write ability. Note that this kernel service cannot be reached in the sandbox and it needs "com.apple.hid.manager.user-access-device" entitlement. So the attack needs a sandbox escape and an entitlement bypass before using this vulnerability.

0x06 iOS 9.1 Critical and Exploitable Vulnerabilities

CVE-2015-7037 Photos Sandbox Escape: The vulnerability exists in the com.apple.PersistentURLTranslator.Gatekeeper XPC service. By using a crafted XPC message, the attacker can achieve arbitrary file read/write ability of “mobile” user outside the sandbox. Combining with the vulnerability of dyld, the attacker can achieve arbitrary code execution outside the sandbox.

CVE-2015-7084 IORegistryIterator Race Condition: The vulnerability exists in the IOKit kernel service. Because the kernel does not lock the process when it frees the IORegistryIterator object, the attacker can use multi-thread to do the race. If the race wins, the vulnerability will cause a double free. Then the attacker can use the vulnerability to achieve kernel read/write ability and jailbreak the iOS devices.

0x07 iOS 9.0 Critical and Exploitable Vulnerabilities

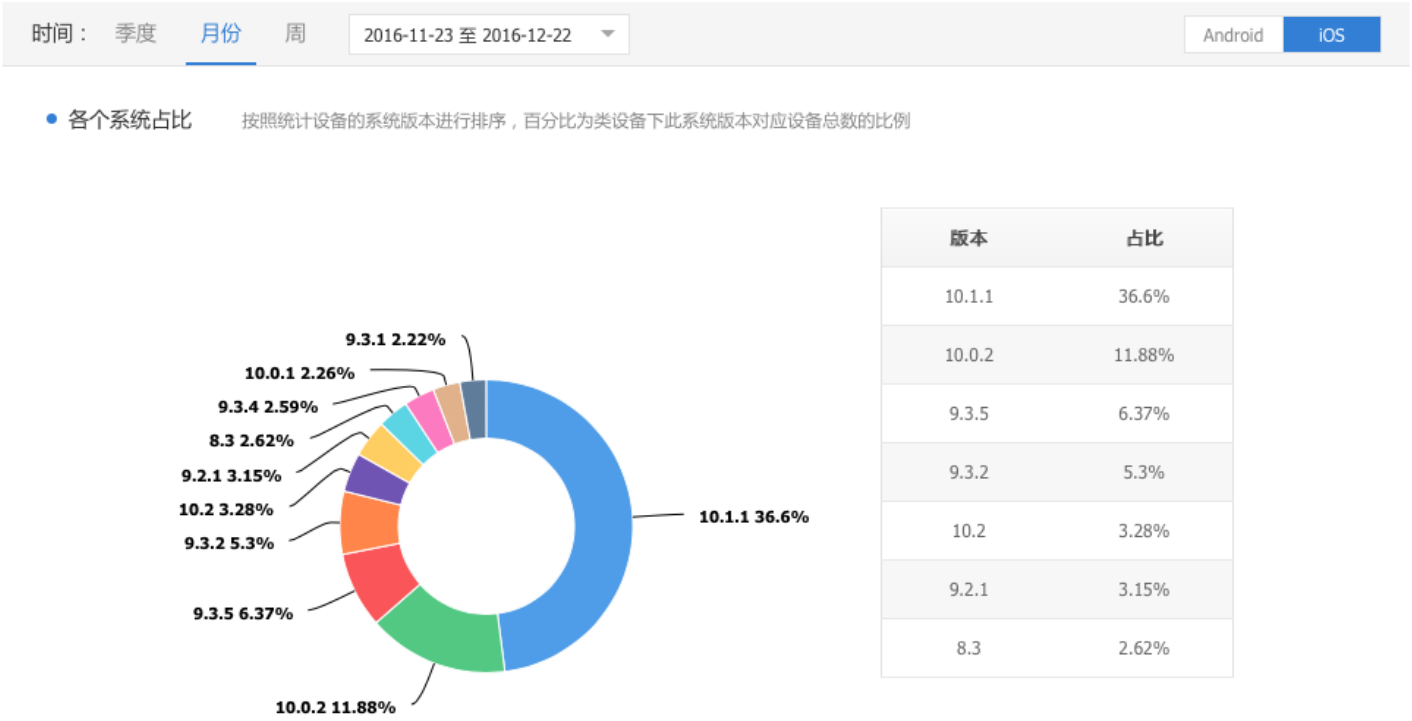
```
//--------------------------------------------------------------------
// IOHIDResourceDeviceUserClient::terminateDevice
//--------------------------------------------------------------------
IOReturn IOHIDResourceDeviceUserClient::terminateDevice()
{
    if (_device) {
        _device->terminate();
    }
    OSSafeRelease(_device);

    return kIOReturnSuccess;
}
```

```
/*! @function OSSafeRelease
 *  @abstract Release an object if not <code>NULL</code>.
 *  @param    inst  Instance of an OSObject, may be <code>NULL</code>.
 */
#define OSSafeRelease(inst)       do { if (inst) (inst)->release(); } while (0)
```

1. CVE-2015-6974 IOHIDFamily UaF: The vulnerability exists in the IOHIDResource kernel service. The kernel service does not set the “device” pointer to NULL after releasing the device in the terminateDevice() method. The attacker can use this vulnerability to trigger UaF in the kernel and then translate into kernel read/write ability. This vulnerability was used in the Pangu’s iOS 9.0 jailbreak. Note that this kernel service cannot be reached in the sandbox. So the attacker needs a sandbox escape before using this vulnerability.

0x08 Summary

We can clearly observe that the number of critical and exploitable vulnerabilities in 2016 is very large. However, lots of iOS devices cannot upgrade to the latest iOS version. In addition, there are minor changes in recent iOS systems. So, more and more people lack interest in upgrading their devices.



According to one professional mobile statistics platform, only 3.28% devices are using the latest iOS 10.2 in December of 2016. It means 96.72% devices can be exploited by Mach_portal exploit chain at that time. Therefore, we kindly remind customers to upgrade their devices and be careful with the potential threats in the future.

Last but not least，you can find iOS jailbreak vulnerabilities and materials related to this article in our Github：https://github.com/zhengmin1989/GreatiOSJailbreakMaterial