Arbitrary Pointer use in distnoted via XPC

Gabe Kirkpatrick - g@be-k.biz

Summary

An issue exists in distnoted which can cause a value supplied by a client application over XPC to be used as a pointer to an xpc_connection_t object. I believe that this could be used as part of an exploit to gain code execution in distnoted, as as such escape an app sandbox. distnoted is accessible from many application sandboxes, and is communicated to by many applications, as such having code execution is distnoted could provide the ability to attack other applications which are it's clients.

Tested on the following software version:

```
ProductName: macOS
ProductVersion:  11.1
BuildVersion:    20C69
```

PoC Details

This bug can be reproed by sending 2 XPC messages to `com.apple.distributed_notifications@Uv3`. I have provided an application that repeatedly sends these messages by calling `addObserver` and `removeObserver` on `NSDistributedNotificationCenter`., as well as a dylib to be injected into the application, which will modify the XPC messages right before they are sent to distnoted.

The dylib which does the actual hooking of the client's xpc_connection_send_message is written using the frida-gum hooking library which is provided. The logic for modifying the XPC message is located at line 87 in

Repro Steps:
Build and run the PoC by running the following from the `distnoted_poc` folder:

```
make
./run_poc.sh
```

This should cause the distnoted process to repeatedly crash with the following exception:

```
Exception Type:        EXC_BAD_ACCESS (SIGSEGV)
Exception Codes:       KERN_INVALID_ADDRESS at 0x0000414141414144
Exception Note:        EXC_CORPSE_NOTIFY
```

Bug Details

The XPC messages normally sent by the `addObserver` & `removeObserver` functions look like the following:

addObserver:

```
msg: <dictionary: 0x…> { count = 7, transaction: 0, voucher = 0x0,
contents =
      "options" => <uint64: 0x…>: 4
      "object" => <string: 0x…> { length = 24, contents =
"kCFNotificationAnyObject" }
      "token" => <uint64: 0x…>: 11115375362048
      "name" => <string: 0x…> { length = 47, contents =
"com.apple.HIToolbox.endMenuTrackingNotification" }
      "pn" => <string: 0x…> { length = 47, contents = "/Users/gabe/
distnoted_poc/bin/NotificationsTest" }
      "method" => <string: 0x…> { length = 8, contents = "register" }
      "version" => <uint64: 0x…>: 1
}
```

removeObserver:

```
msg: <dictionary: 0x…> { count = 3, transaction: 0, voucher = 0x0,
contents =
      "tokens" => <array: 0x…> { count = 1, capacity = 8, contents =
            0: <uint64: 0x…>: 11115375362048
      }
      "method" => <string: 0x…> { length = 10, contents =
"unregister" }
      "version" => <uint64: 0x…>: 1
}
```

The bug is triggered by passing an invalid value for `"options"` in the `addObserver` message, and then sending the `removeObserver` message. Doing this will result in the value provided for `"token"` in the first message & `"tokens"` in the second message to be used as a pointer to an `xpc_connection_t` object.

I have attached a crash file showing the results of running the PoC.

Full attacker control of a pointer in a remote process is very dangerous and likely could be used as part of an exploit to gain code execution in distnoted.