

Objective

We have to defeat the enemies
and save the earth!!



Our contributions

- Introduce a **new concept of analysis framework** to use easily
 - perform analysis of normal application by using web proxy
- Introduce methodologies for implementing our concept
 - pros and cons of the methodologies
- Demonstrate use cases

Define Keyword

- **Web Application**

- consist of usually script languages
- operate based on web server/client

- **Normal Application**

- executable binary except for web application
- PE, ELF, etc.

- **Web Proxy**

- a tool for web application analysis
- Burp suite, paros, fiddler, etc.

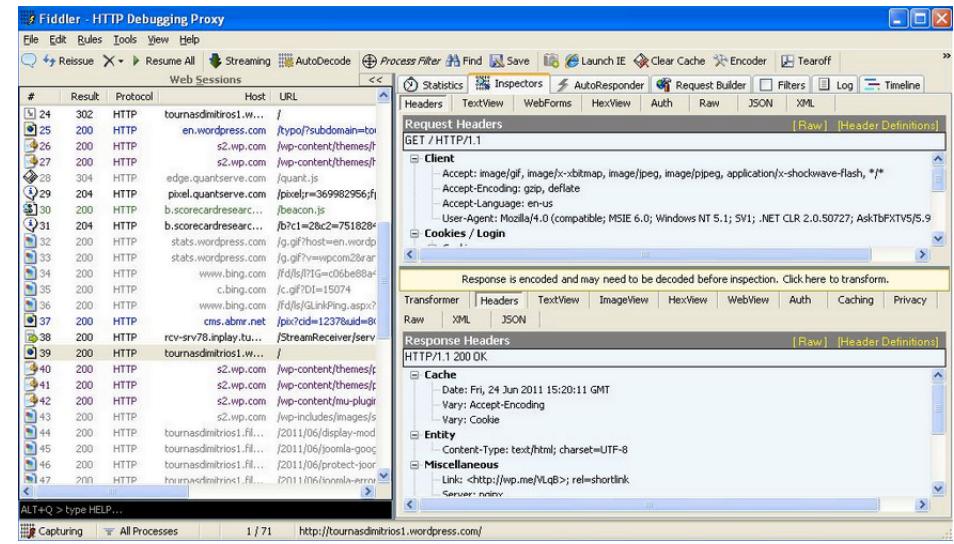
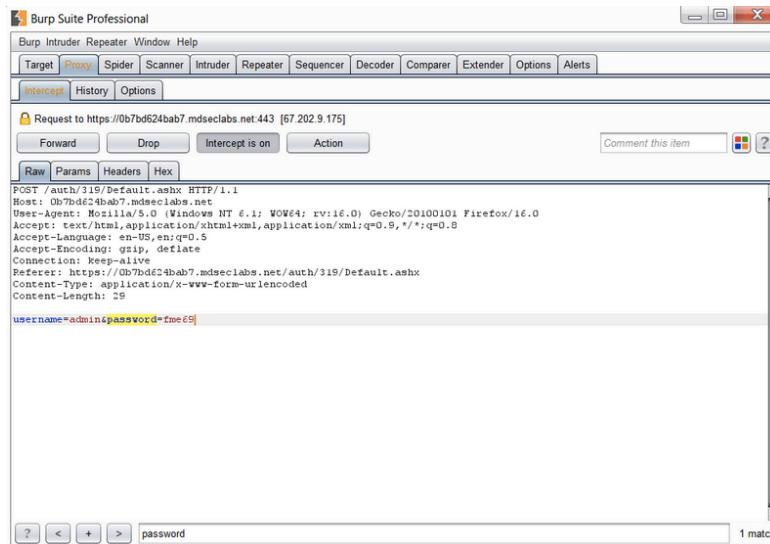
What's wrong?

Background

Existing methodologies/tools for application analysis

Web Application analysis

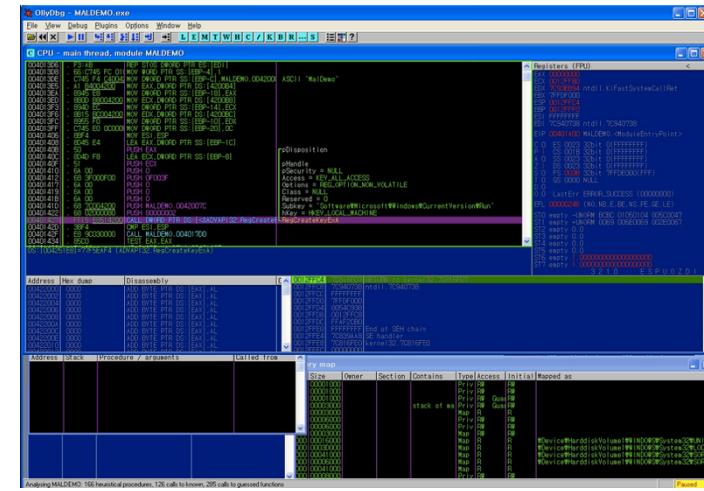
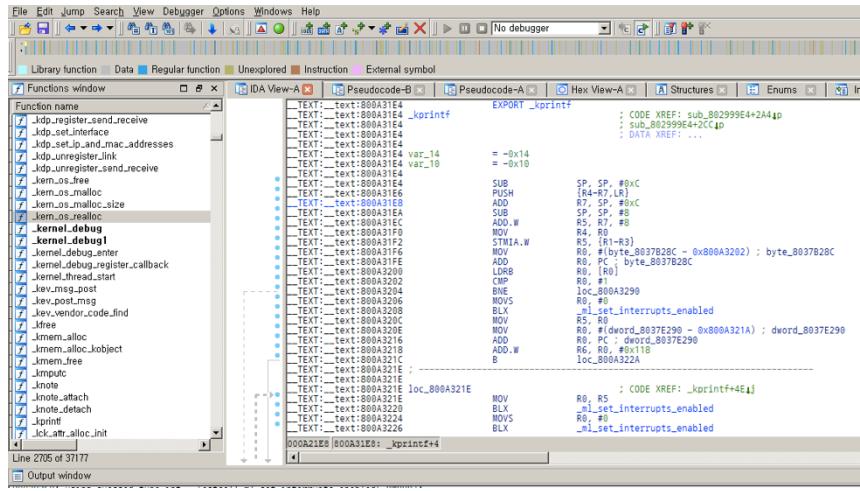
- easy to use and operate using a web proxy (burp, paros, fiddler, etc.)
- monitor and modify the contents without difficulty



Existing methodologies/tools (cont'd)

Normal Application (executables) analysis

- much harder and complex than web application (GDB, IDA, Ollydbg, windbg, etc.)
- In secure channel, how can we check the contents?



Challenges for application analysis

We cannot
save the earth
using our resources



Lack of time and manpower

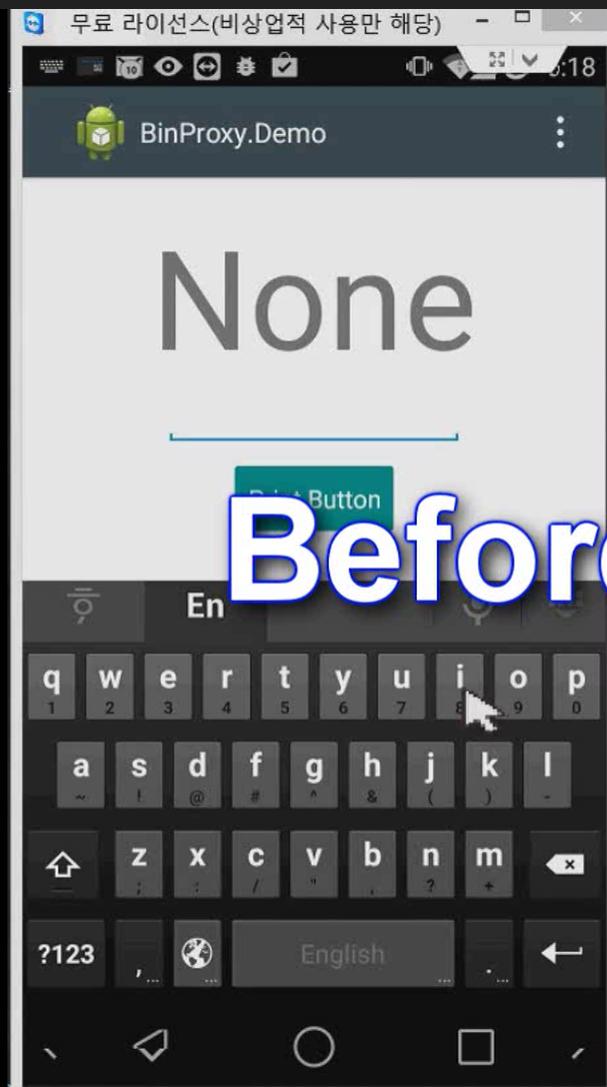
How to solve a problem?



Need a EASY tool

So what??

BinProxy : A New Paradigm for Binary Analysis



```
root@bt: ~/project/DalvikHook/ddi/examples/server/jni
root@maguro:/data/local/tmp # ./server

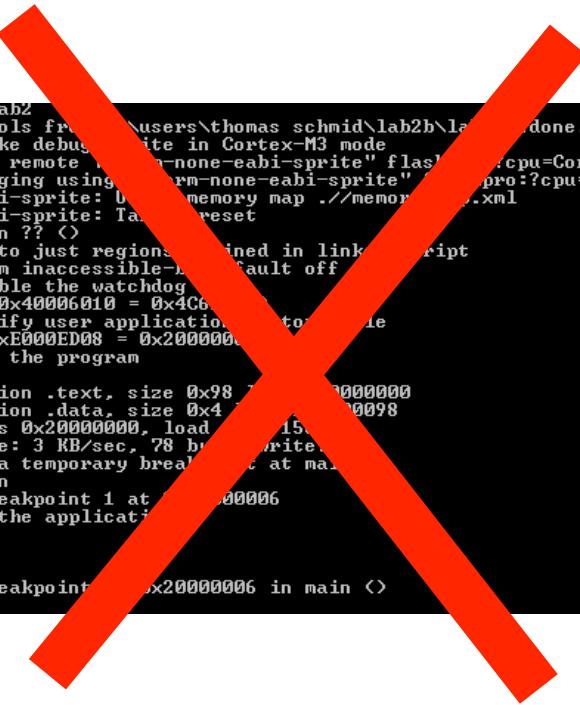
root@bt: ~/project/BinProxy
root@bt:~/project/BinProxy# python ./server.py
```

The image shows two terminal windows. The top window is running the command `./server` in a directory under `DalvikHook`. The bottom window is running `python ./server.py` in a directory under `BinProxy`. Both windows show a blank black screen.

A large blue watermark "Before Injection" is overlaid across the middle of the image.

Key Features

We do not need gdb and ollydbg
to analyze applications any more.



```
(gdb) file lab2
Reading symbols from C:\Users\thomas.schmid\lab2b\lab2... done.
(gdb) # Invoke debug remote in Cortex-M3 mode
(gdb) target remote arm-none-eabi-sprite" flash -cpu=Cortex-M3 "./"
Remote debugging using "arm-none-eabi-sprite" <pro:?cpu=Cortex-M3 "./"
arm-none-eabi-sprite: b /memory map ./memory.map.xml
arm-none-eabi-sprite: Target reset
0x6008051c in ?? ()
Copy access to just regions defined in link script
(gdb) set mem inaccessible-by-default off
(gdb) # Disable the watchdog
(gdb) #set *$040006010 = $04C60000
(gdb) # Specify user application start file
(gdb) set *$0x1000ED08 = $0x2000000
(gdb) # Load the program
(gdb) load
Loading section .text, size 0x98   0x20000000
Loading section .data, size 0x4   0x2000098
Start address 0x20000000, load size 15
Transfer rate: 3 KB/sec, 78 bytes written.
(gdb) # set a temporary breakpoint at main
(gdb) tb main
Temporary breakpoint 1 at 0x20000006
(gdb) # Run the application
(gdb) cont
Continuing.

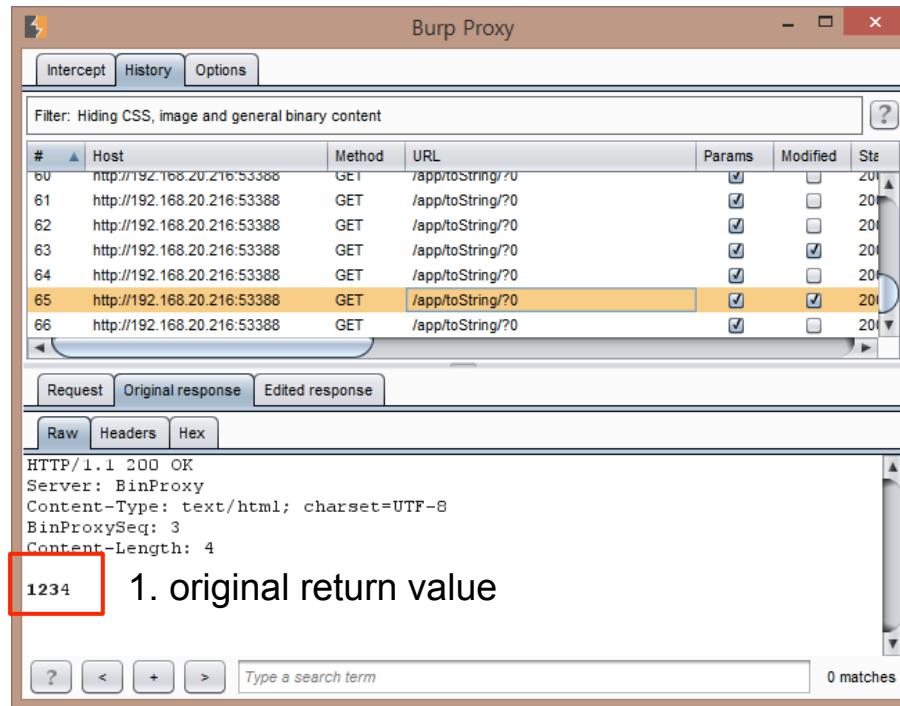
Temporary breakpoint 1 at 0x20000006 in main ()
```



Key Features (cont'd)

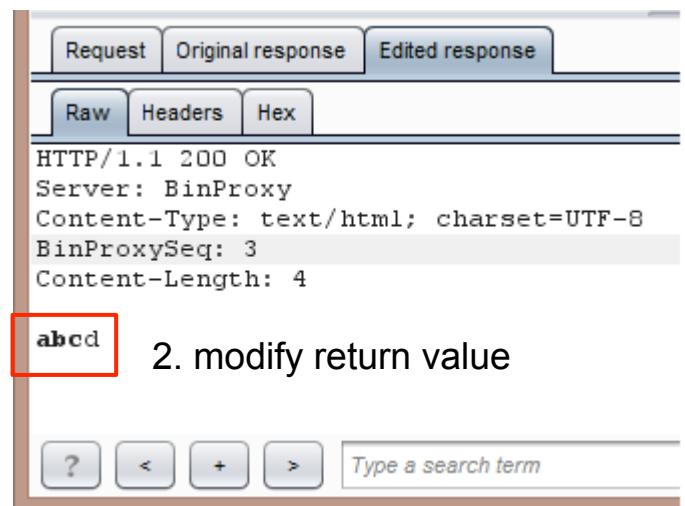
Should we use the difficult tools for **simple analysis**?

You can monitor and control the normal applications
with your favorite web proxy



The screenshot shows the Burp Proxy interface. At the top, there's a toolbar with 'Intercept', 'History', and 'Options' buttons. Below that is a table with columns: #, Host, Method, URL, Params, Modified, and Status. A filter bar above the table says 'Filter: Hiding CSS, image and general binary content'. The table lists several requests, with the 65th request selected. The bottom part of the interface shows tabs for 'Request', 'Original response', and 'Edited response', with 'Raw' selected. The 'Raw' tab displays the HTTP response: 'HTTP/1.1 200 OK', 'Server: BinProxy', 'Content-Type: text/html; charset=UTF-8', 'BinProxySeq: 3', and 'Content-Length: 4'. A red box highlights the number '1234' in the 'Original response' tab. The status bar at the bottom says '0 matches'.

1. original return value



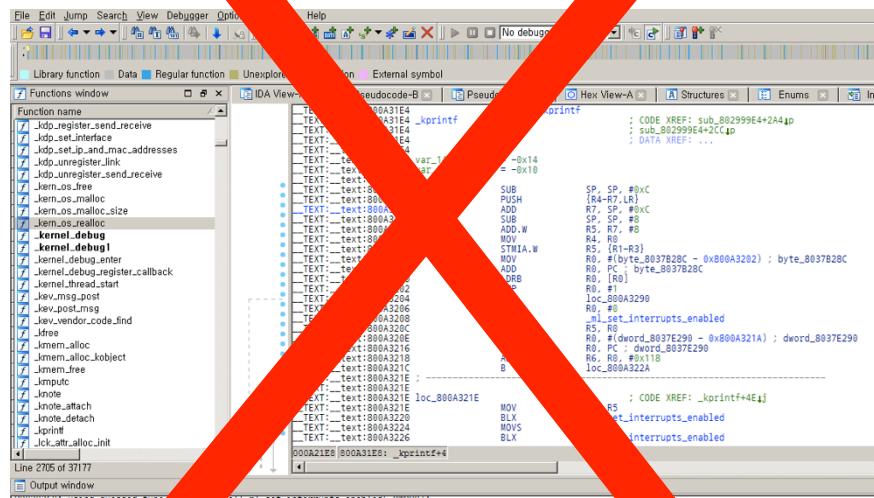
The screenshot shows the Burp Suite interface. At the top, there are tabs for 'Request', 'Original response', and 'Edited response', with 'Original response' selected. Below that is another set of tabs for 'Raw', 'Headers', and 'Hex', with 'Raw' selected. The 'Raw' tab displays the modified HTTP response: 'HTTP/1.1 200 OK', 'Server: BinProxy', 'Content-Type: text/html; charset=UTF-8', 'BinProxySeq: 3', and 'Content-Length: 4'. The word 'abcd' is entered in the search bar at the bottom. A red box highlights the word 'abcd'. The status bar at the bottom says 'Type a search term'.

2. modify return value

3. Click Forward Button

Key Features (cont'd)

We do not want to use difficult IDA tool
to analyze applications any more.



Key Features (cont'd)

You can know what functions are existed in target apps and what functions can be monitored.

The screenshot shows the Burp Suite Free Edition v1.6 interface. The top menu bar includes Burp, Intruder, Repeater, Window, Help, Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Options, and Alerts. Below the menu is a toolbar with Site map and Scope buttons, where Site map is selected. A filter bar at the top of the main pane says "Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders".

The left pane displays a tree view of URLs under "http://127.0.0.1:53388". One node, "target_app/java.net.URLConnection/", is expanded, showing methods like disconnect, getContentEncoding, getPermission, getRequestMethod, getResponseCode, and setRequestMethod. Another node, "java.util.String", is also expanded, showing methods like compareTo, compareToIgnoreCase, concat, contains, contentEquals, endsWith, equalsIgnoreCase, getBytes, getChars, hashCode, indexOf, intern, lastIndexOf, length, matches, regionMatches, toString, toUpperCase, and trim.

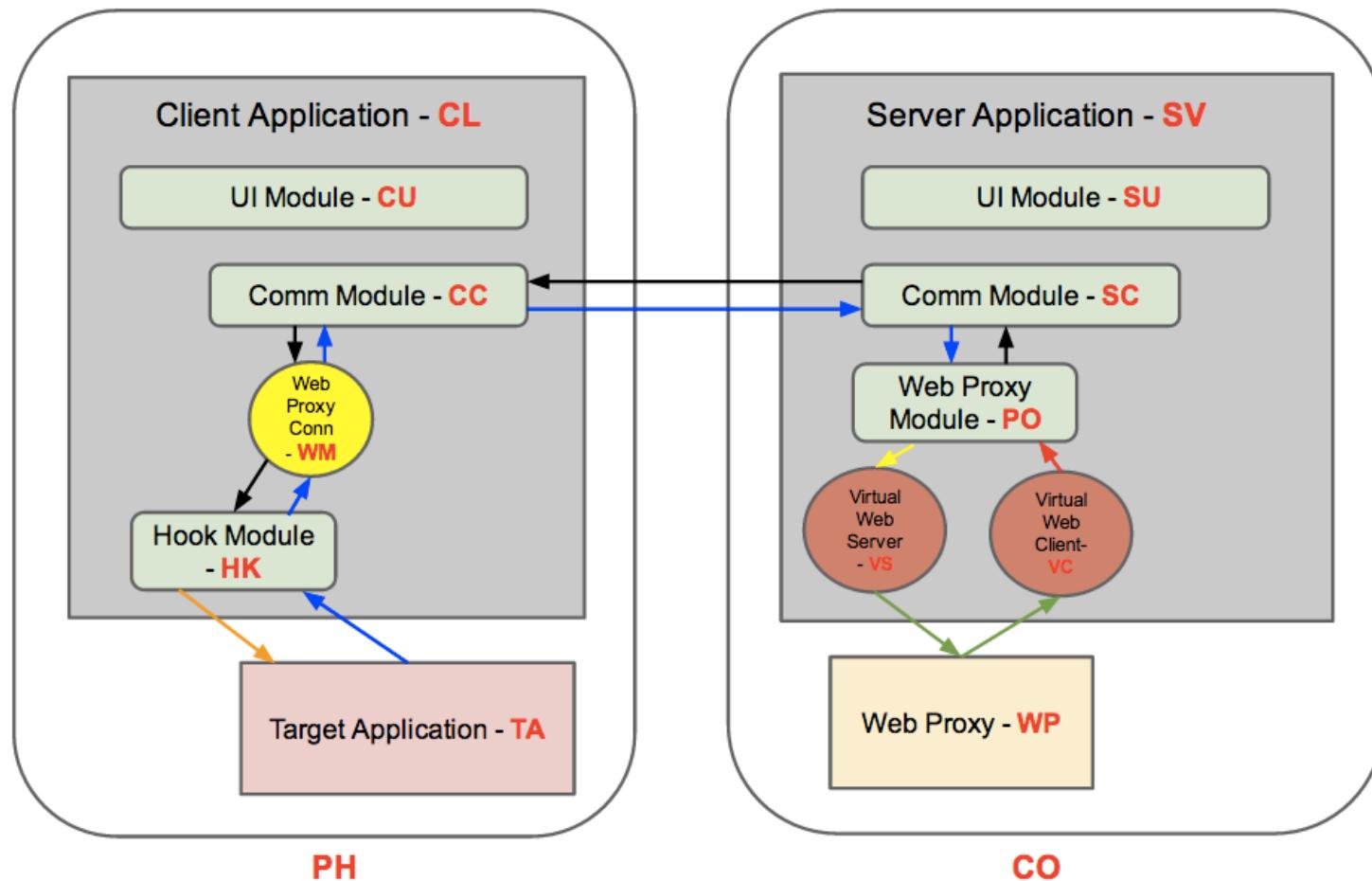
The right pane has tabs for Host, Method, URL, Params, Status, Length, and MIME type. A single row is selected in the table, showing Host: http://127.0.0.1:53388, Method: GET, URL: /target_app/java.net...., Params: (checkbox checked), Status: 200, Length: 115, and MIME type: text.

Below the table are tabs for Request and Response. The Request tab shows the following HTTP request:

```
GET /target_app/java.net.URLConnection/?getContentEncoding HTTP/1.1
Accept: /*
Accept-Language: en
User-Agent: BinProxy V-Client
Connection: close
Host: 127.0.0.1:53388
BinProxySeq: 167
BinProxyControl: HOOK
```

At the bottom of the interface is a search bar with the placeholder "Type a search term" and a note "0 matches".

Overall Architecture



Components

- Target application
 - smart phone apps, executable program based on Windows, OSX and etc.
- Web Proxy
 - A user-friendly proxy to be used for analysis (ex. burp, paros, ...)
- BinProxy Client
 - is Operated in the target application is installed
 - communication module : communicate with BinProxy server
 - hooking module : modify the flow of functions.
- BinProxy Server
 - is Operated in the web proxy is installed
 - communication module : communicate with BinProxy client and web proxy

What You Need

Need things to make BinProxy

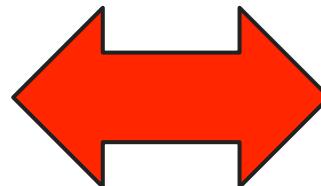
Intercept function call & Forward it to a Web proxy

Main techniques for implementation

how to control function calls by using web proxy

Convert Functions

```
haking@live:~/randisk/home/haking
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "i386-redhat-linux-gnu"...Using host libthread_db library "/lib/tls/libthread_db.so.1".
(gdb) list
1 void fn(char *a) {
2     char buf[10];
3     strcpy(buf, a);
4     printf("the end of fn\n");
5 }
6
7 main (int argc, char *argv[]) {
8     fn(argv[1]);
9     printf("the end\n");
10 }
(gdb) break 3
Breakpoint 1 at 0x0040382: file stack_1.c, line 3.
(gdb) run AAAA...AAAAA
Starting program: /randisk/home/haking/stack_1 AAAA...
Breakpoint 1, fn (a=0xbfffffc2b 'A' <repeats 30 times>) at stack_1.c:3
3     strcpy(buf, a);
(gdb)
```



Burp Suite Professional

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Site map Scope

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

| Host | Method | URL | Params | Status | Length | MIME type | Title |
|-----------------------------------|--------|----------------------|--------|--------|--------|-----------|----------|
| http://0b7bd624bab7.mdseclabs.net | GET | /addressbook/32/ | | 200 | 2765 | HTML | Contacts |
| | POST | /addressbook/32/0... | | 200 | 3100 | HTML | Contacts |
| | POST | /addressbook/32/0... | | 200 | 4914 | HTML | Contacts |
| | POST | /addressbook/32/0... | | 200 | 2835 | HTML | Contacts |
| | POST | /addressbook/32/0... | | 200 | 2765 | HTML | Contacts |

addressbook

- admin
- app
- auth
- bank
- clookup
- employees
- error
- feedback
- filestore
- search
- settings
- shop
- updates

Add to scope
Spider this branch
Actively scan this branch
Passively scan this branch
Engagement tools
Compare site maps
Expand branch
Expand requested items
Delete branch
Copy URLs in this branch
Copy links in this branch
Save selected items
Site map help

1.1
abs.net
Windows NT 6.1; WOW64; rv:16.0 Gecko/20100101
Content-type:application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate
Proxy-Connection: keep-alive
Referer: http://0b7bd624bab7.mdseclabs.net/labs/lab.ashx?lab=?

Type a search term 0 matches

Function monitoring and Function Controlling

Main techniques for implementation (cont'd)

API / User-defined function

Hooking

Function monitoring and Function Controlling

Main techniques for implementation (cont'd)

Dynamic function Hooking

No need a pre-compiled hooking code
Dynamic target function selection

Function monitoring and Function Controlling

Main techniques for implementation (cont'd)

Return value,

primitive / reference

type arguments

Target function selection

Main techniques for implementation (cont'd)

Extraction API lists

| | |
|----------------------------|-----------|
| _ml_at_interrupt_context | 8001F920 |
| _ml_io_map | 8001F940 |
| _ml_get_entropie | 8001FA78 |
| _ml_stack_remaining | 8001FA90 |
| _current_thread | 8001FAE0 |
| _enable_kernel_vfp_context | 8001FBBA4 |
| _OSSynchronizeIO | 8001FBD8 |
| _copyinstr | 8001FDB4 |
| _copyin | 8001FE58 |
| _copyout | 8001FF48 |
| _ml_get_interrupts_enabled | 8002029C |
| _ml_set_interrupts_enabled | 800202C0 |
| _ovbcopy | 80020924 |
| _memmove | 80020930 |
| _memset | 80020C58 |
| _bzero | 80020C70 |
| _strlen | 80020D78 |
| _strnlen | 80020DE8 |

Target function selection (cont'd)

Main techniques for implementation (cont'd)

Extracting user-defined functions
and Finding out Args and Types

```
1 int __fastcall sub_805D2CE0(int a1, int a2, int a3, int a4, int a5, int a6)
2 {
3     int v6; // r4@1
4     int v7; // r0@1
5     int v8; // r1@1
6     int v9; // r0@2
7     int v11; // [sp+8h] [bp-18h]@1
8     int v12; // [sp+C] [bp-14h]@1
9     int v13; // [sp+10h] [bp-10h]@1
10    int v14; // [sp+14h] [bp-C]@1
11    int v15; // [sp+18h] [bp-8h]@1
```

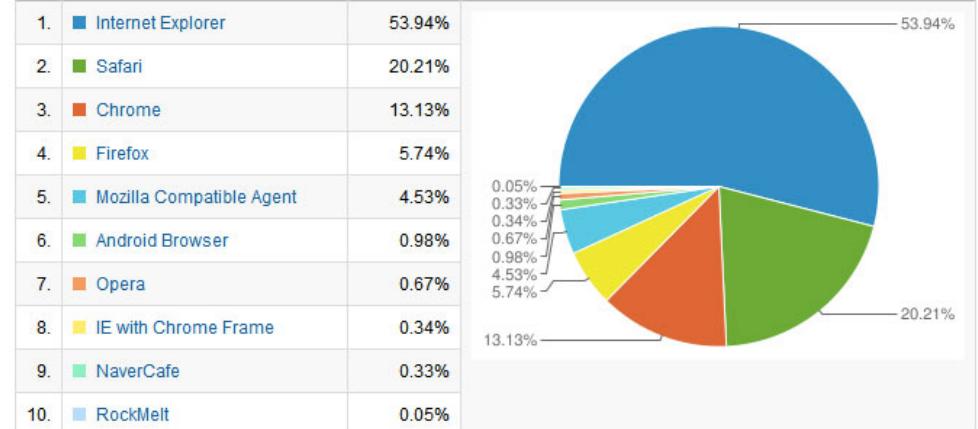
| | |
|--|--------------|
| | sub_805B2E94 |
| | sub_805B2EA4 |
| | sub_805B2EB4 |
| | sub_805B2EC4 |
| | sub_805B2ED4 |
| | sub_805B2EE4 |
| | sub_805B2EF4 |
| | sub_805B2F04 |
| | sub_805B2F14 |
| | sub_805B2F24 |
| | sub_805B2F34 |
| | sub_805B2F44 |
| | sub_805B2F54 |
| | sub_805B2F64 |
| | sub_805B2F84 |
| | sub_805B2F94 |
| | sub_805B2FA4 |
| | sub_805B2FB4 |
| | sub_805B2FE4 |
| | sub_805B2FF4 |
| | sub_805B3004 |
| | sub_805B3014 |
| | sub_805B3024 |
| | sub_805B3034 |
| | sub_805B3044 |

Target function selection (cont'd)

Main techniques for implementation (cont'd)

Monitoring function calls and statistics
-> Selecting target functions easily

| | | |
|------|---------------------|--|
| 5950 | 18:48:58:660:247,6 | SetRect(prc: 0x0000000000) [FD:70] {left=14,right=588,top=237,bottom=259},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045AE9 |
| 5952 | 18:48:58:660:447,5 | GetDC(hWnd:d: 0x0000000000) [FD:80] {left=14,right=588,top=231,bottom=233},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x0000000000004429F |
| 5953 | 18:48:58:660:544,2 | ReleaseDC(hWnd:d: 0x0000000000) [FD:90] {0xFFFFFFF90111DB} |
| 5954 | 18:48:58:660:640,8 | SetRect(prc: 0x0000000000) [FD:CF0] {left=14,right=588,top=252,bottom=274},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000044339 |
| 5955 | 18:48:58:660:739,3 | SetRect(prc: 0x0000000000) [FD:70] {left=14,right=588,top=258,bottom=280},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045916 |
| 5956 | 18:48:58:660:837,7 | SetRect(prc: 0x0000000000) [FD:80] {left=14,right=588,top=232,bottom=274},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045916 |
| 5957 | 18:48:58:660:936,7 | GetDC(hWnd:d: 0x0000000000) [FD:90] {0xFFFFFFF90111DB} |
| 5958 | 18:48:58:661:133,8 | ReleaseDC(hWnd:d: 0x0000000000) [FD:90] {0xFFFFFFF90111DB} |
| 5959 | 18:48:58:661:130,4 | SetRect(prc: 0x0000000000) [FD:CF0] {left=14,right=588,top=273,bottom=295},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000044339 |
| 5960 | 18:48:58:661:1246,9 | SetRect(prc: 0x0000000000) [FD:70] {left=14,right=588,top=279,bottom=301},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045AE9 |
| 5961 | 18:48:58:661:1358,0 | SetRect(prc: 0x0000000000) [FD:80] {left=14,right=588,top=273,bottom=295},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045916 |
| 5962 | 18:48:58:661:1463,0 | GetDC(hWnd:d: 0x0000000000) [FD:90] {0xFFFFFFF90111DB} |
| 5963 | 18:48:58:661:1562,7 | ReleaseDC(hWnd:d: 0x0000000000) [FD:90] {0xFFFFFFF90111DB} |
| 5964 | 18:48:58:661:1662,4 | SetRect(prc: 0x0000000000) [FD:CF0] {left=14,right=588,top=294,bottom=316},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045916 |
| 5965 | 18:48:58:661:1762,0 | SetRect(prc: 0x0000000000) [FD:70] {left=14,right=588,top=300,bottom=322},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045AE9 |
| 5966 | 18:48:58:661:1860,2 | SetRect(prc: 0x0000000000) [FD:80] {left=14,right=588,top=294,bottom=316},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045916 |
| 5967 | 18:48:58:661:1960,4 | GetDC(hWnd:d: 0x0000000000) [FD:90] {0xFFFFFFF90111DB} |
| 5968 | 18:48:58:662:055,6 | ReleaseDC(hWnd:d: 0x0000000000) [FD:90] {0xFFFFFFF90111DB} |
| 5969 | 18:48:58:662:152,9 | SetRect(prc: 0x0000000000) [FD:CF0] {left=14,right=588,top=315,bottom=337},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045916 |
| 5970 | 18:48:58:662:251,1 | SetRect(prc: 0x0000000000) [FD:70] {left=14,right=588,top=321,bottom=343},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045AE9 |
| 5971 | 18:48:58:662:353,7 | SetRect(prc: 0x0000000000) [FD:80] {left=14,right=588,top=315,bottom=337},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045916 |
| 5972 | 18:48:58:662:451,9 | GetDC(hWnd:d: 0x0000000000) [FD:90] {0xFFFFFFF90111DB} |
| 5973 | 18:48:58:662:550,0 | ReleaseDC(hWnd:d: 0x0000000000) [FD:90] {0xFFFFFFF90111DB} |
| 5974 | 18:48:58:662:646,4 | SetRect(prc: 0x0000000000) [FD:CF0] {left=14,right=588,top=336,bottom=358},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045916 |
| 5975 | 18:48:58:662:744,8 | SetRect(prc: 0x0000000000) [FD:70] {left=14,right=588,top=342,bottom=364},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045AE9 |
| 5976 | 18:48:58:662:844,8 | SetRect(prc: 0x0000000000) [FD:80] {left=14,right=588,top=336,bottom=358},x1:0x0000... 0x00000000000000000000000000000000 explorerframe.dll + 0x00000000000045916 |



How to make?

the way of building BinProxy

How to interwork with a web proxy - BinProxy Client

```
int hooked_func(arg1, arg2, ...)  
{  
    arg_string = make_arg_string(original_arg1, original_arg2, ...);  
    new_arg_string = send_to_server_and_wait("  
        HOOK_INFO^^before_call^#{original_function_name}^^#{arg_string}");  
}
```

- * hooked_func send before_call message to BinProxy Server through communication module.
- * before_call message = function name + the value of arguments
- * After sending a before_call message, the hooked_func will be blocked until getting response from BinProxy Server.

```
if( new_ret_string == ret_string )  
    return ret;  
else  
    return parse(new_ret_string, 1);  
}
```

How to interwork with a web proxy - BinProxy Server

BinProxy Server convert a `before_call` message into HTTP request format for delivering the message to Web Proxy.

```
POST http://127.0.0.1:53388/function_name
Host: target_app_name
User-Agent: BinProxy
```

01_414141

How to interwork with a web proxy - Web Proxy

The screenshot shows the Burp Suite interface, specifically the Repeater tab. The title bar includes 'Burp' and 'Intruder Repeater Window Help'. The menu bar has 'Proxy' selected. Below the menu is a toolbar with 'Intercept' (highlighted), 'HTTP history', 'WebSockets history', and 'Options'. A main panel displays a network request to 'http://127.0.0.1:53388'. The request details are as follows:

```
GET /target_app_name/function_name/?414141 HTTP/1.1
Accept: */*
Accept-Language: en
User-Agent: BinProxy V-Client
Connection: close
Host: 127.0.0.1:53388
BinProxySeq: 12
BinProxyControl: HOOK
```

Below the request details are buttons for 'Forward', 'Drop', 'Intercept is on' (disabled), and 'Action'. At the bottom are tabs for 'Raw', 'Params', 'Headers', and 'Hex'.

How to interwork with a web proxy - BinProxy Client

```
int hooked_func(arg1, arg2, ...)  
{  
    arg_string = make_arg_string(orig  
    new_arg_string = send_to_server_a  
        HOOK_INFO^^before_call^^#{original_...  
    if( new_arg_string == arg_string )  
        ret = original_func(original_arg1, original_arg2, ...);  
    else  
        ret = original_func(parse(new_arg_string, 1), parse(new_arg_string, 2  
            ), ...);  
  
    ret_string = make_ret_string(ret);  
  
    new_ret_string = send_to_server_and_wait("  
        HOOK_INFO^^after_call^^#{original_function_name}^^#{ret_string}");  
}
```

execute an original function

After sending an after_call message,
hooked_func will be blocked until getting response from BinProxy Server.

How to interwork with a web proxy - BinProxy Server

BinProxy Server convert a `after_call` message into HTTP response format for delivering the message to Web Proxy.

```
HTTP/1.1 200 OK
Date: Mon, 04 Aug 2014 17:22:59 GMT
Server: BinProxy
Content-Length: 1
Connection: close
Content-Type: application/return
```

How to interwork with a web proxy - Web Proxy

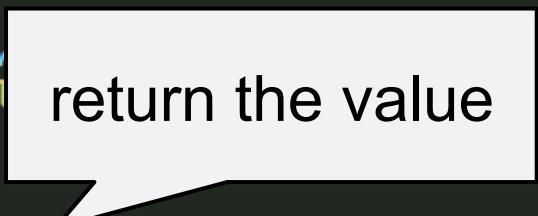
The screenshot shows the Burp Suite interface. The top menu bar includes Burp, Intruder, Repeater, Window, and Help. Below the menu is a toolbar with tabs: Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Options, and Alerts. The 'Proxy' tab is selected. Underneath the toolbar are sub-tabs: Intercept, HTTP history, WebSockets history, and Options; 'Intercept' is selected. A filter bar below the tabs says 'Filter: Hiding CSS, image and general binary content'. The main area displays a table of network requests. The columns are: #, Host, Method, URL, Params, Edited, and Status. A row for request 67 is highlighted in orange. The host is 'http://127.0.0.1:53388', method is 'GET', URL is '/target_app_name/function_name/?414141', Params has a checked checkbox, Edited has an unchecked checkbox, and Status is '200'. Below the table are tabs for Request and Response, with 'Request' selected. At the bottom are Raw, Headers, and Hex tabs, with 'Raw' selected. The raw request data is shown as:

```
HTTP/1.1 200 OK
Server: BinProxy
Content-Type: text/html; charset=UTF-8
BinProxySeq: 14
Content-Length: 1
```

0

How to interwork with a web proxy - BinProxy Client

```
int hooked_func(arg1, arg2, ...)  
{  
    arg_string = make_arg_string(original_arg1, original_arg2, ...);  
  
    new_arg_string = send_to_server_and_wait("HOOK_INFO^^before_call^#{original_function_name}^#{arg_string}");  
  
    if( new_arg_string == arg_string )  
        ret = original_func(original_arg1, original_arg2, ...);  
    else  
        ret = original_func(parse(new_arg_string, 1), parse(new_arg_string, 2  
            ), ...);  
  
    ret_string = make_ret_string(ret);  
  
    new_ret_string = send_to_server_and_wait("HOOK_INFO^^after_call^#{original_function_name}^#{ret_string}");  
  
    if( new_ret_string == ret_string )  
        return ret;  
    else  
        return parse(new_ret_string, 1);  
}
```

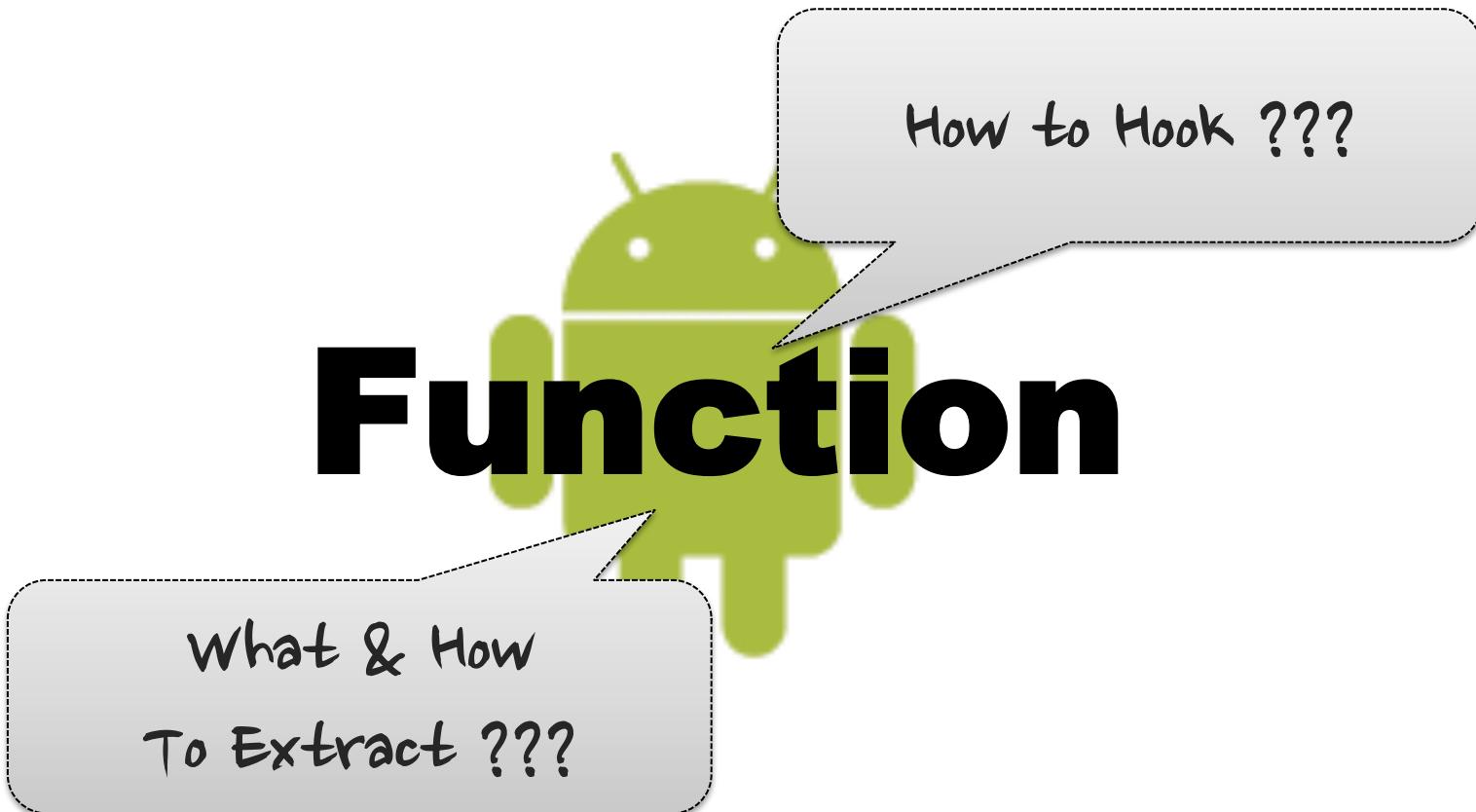


return the value

How to make?

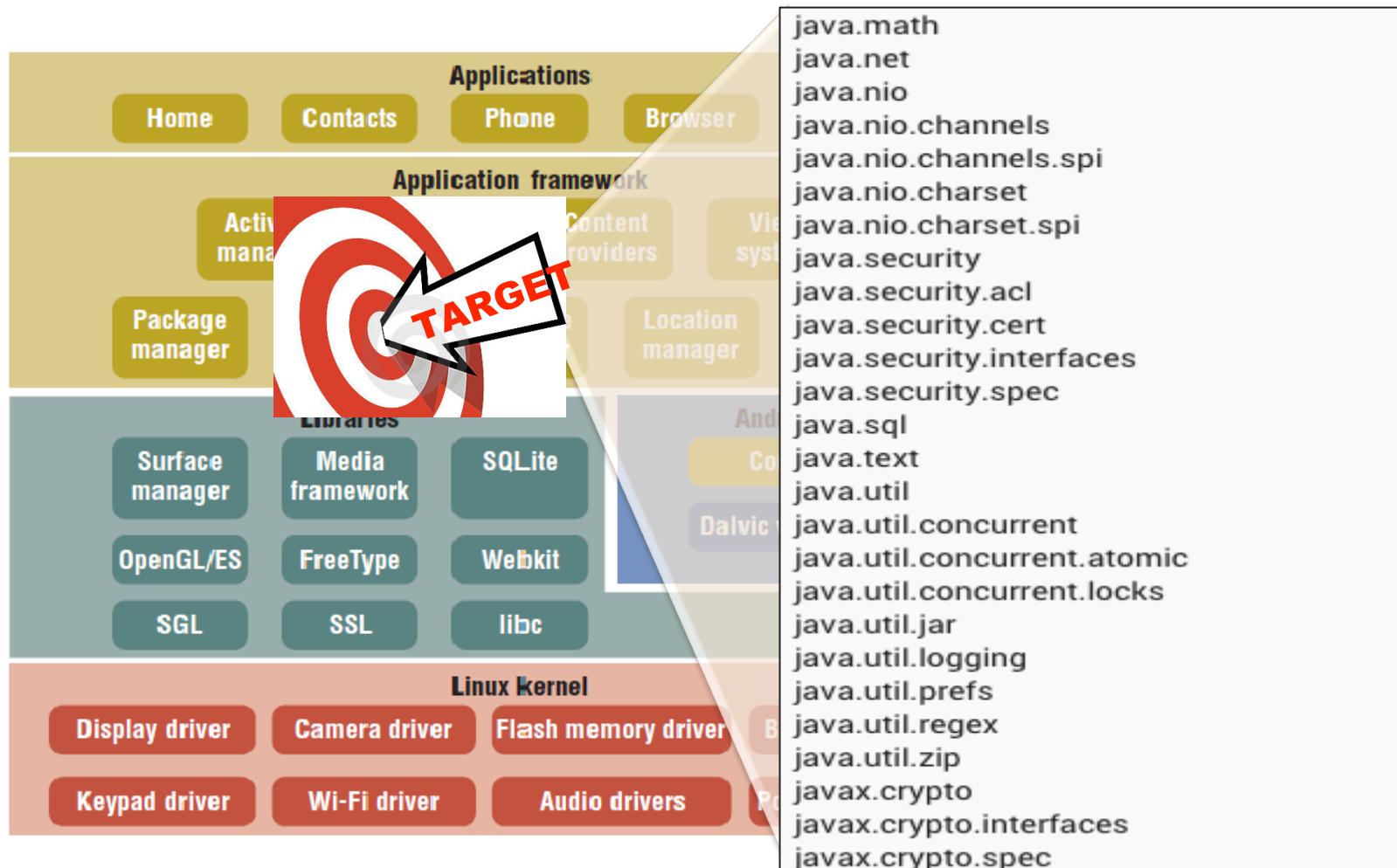
Ways of build android client & PoC

Key Requirements



Key Requirements

- What & How To extract ..



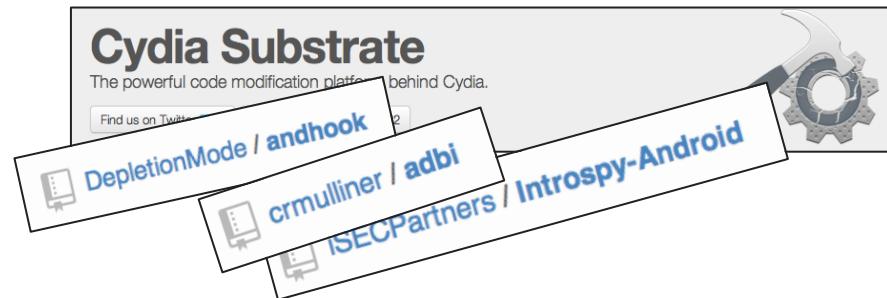
Key Requirements

- What & How To extract ..

We can use for hooking in Android :

- Cydia substrate for Android
- Introspy-Android (GUI Interface + Cydia Substrate)
- AndHook(Android Hooking Framework)
- ADBI(Android Dynamic Binary Instrumentation Toolkit)
- [Paper] Hooking on Android -2014 CodeEngn Conference

....



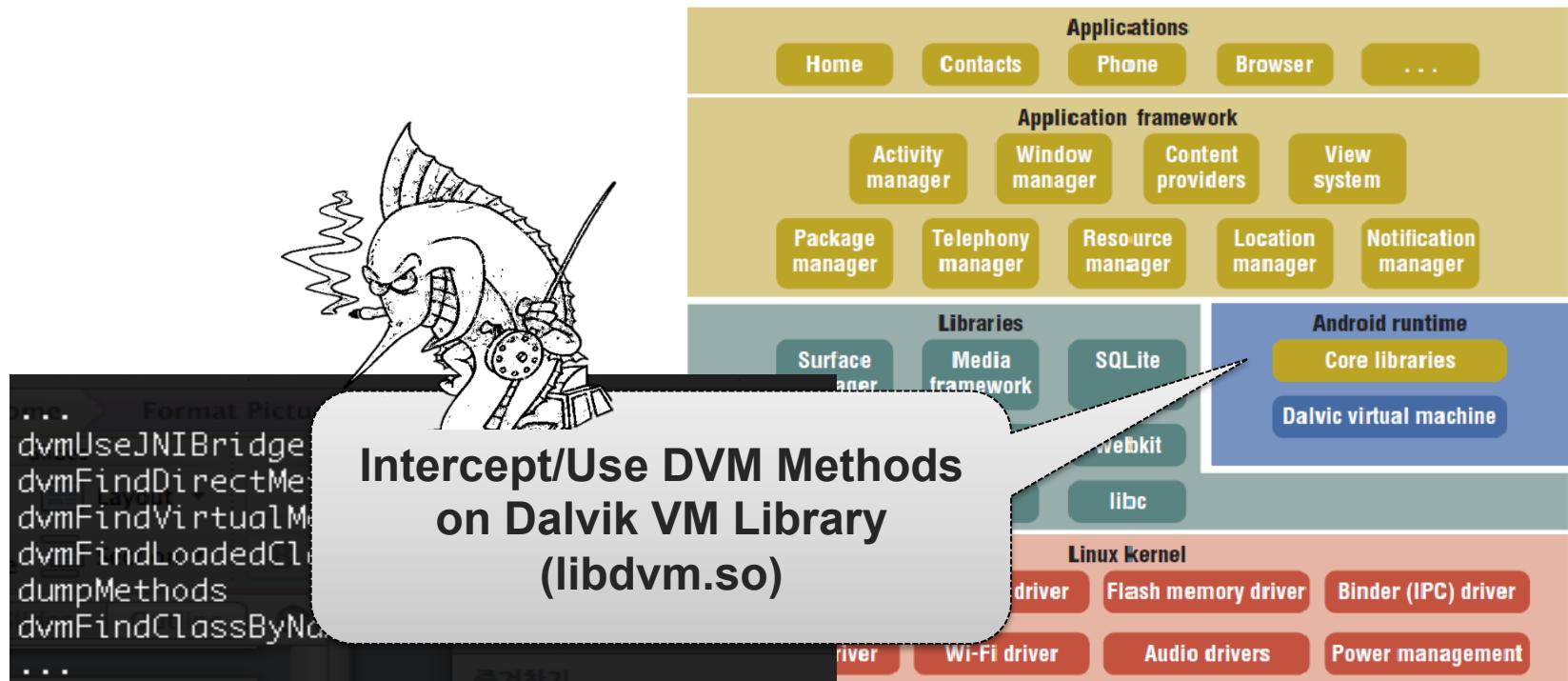
ADBI

(Android Binary Instrumentation Toolkit)

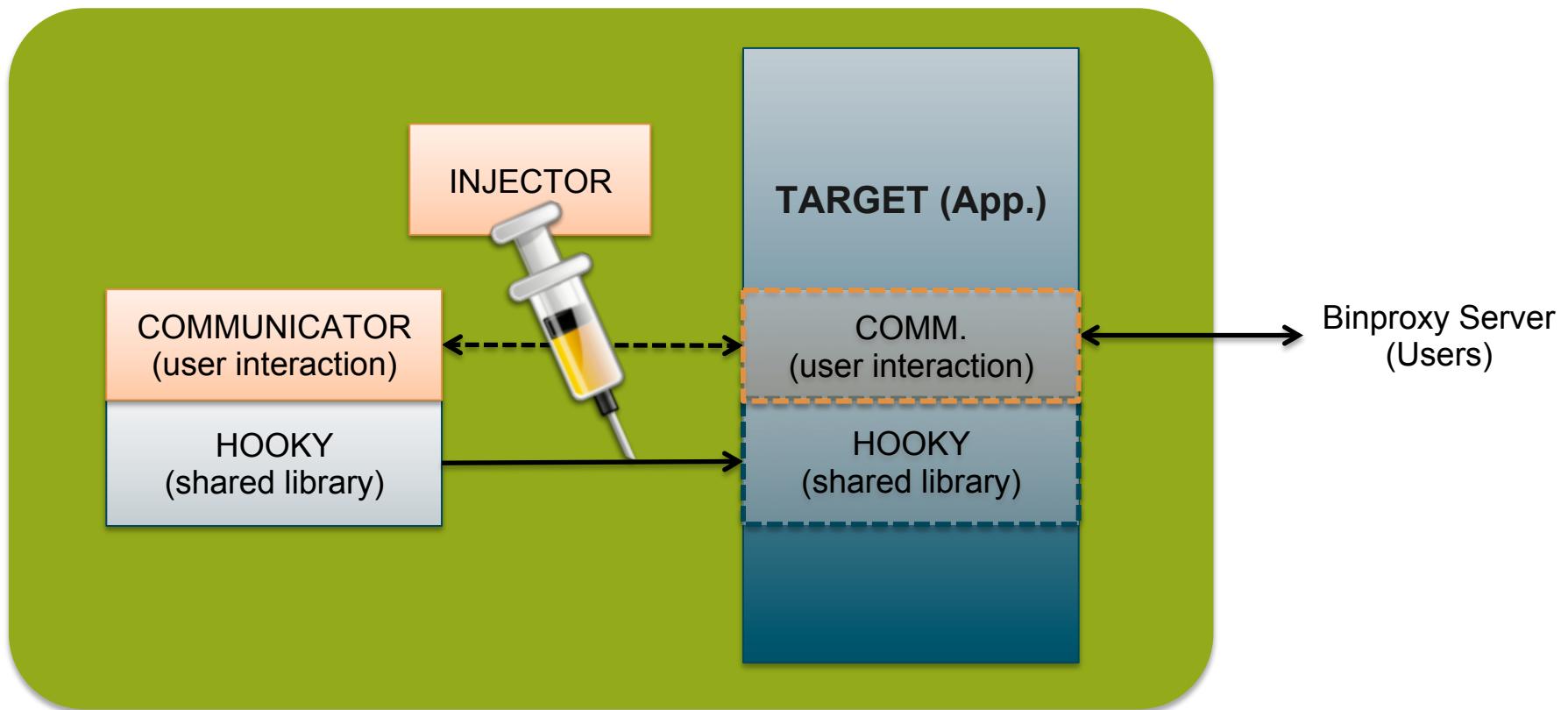
Dynamic Dalvik Instrumentation Framework for Android (old)

- Collin Mulliner, SummerCon 2013.

<https://github.com/crmulliner/adbi> (current)



Binproxy Client modules for Android

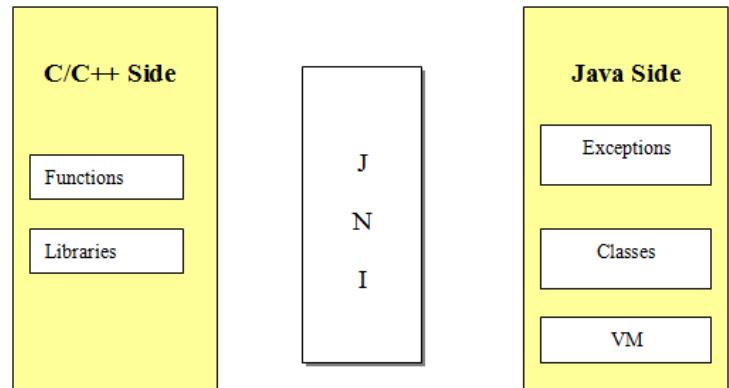


Binproxy Client modules ... (cont'd)

- INJECTOR
 - : Inject the HOOKER(.so) into Target App. (running process)
- HOOKER
 - : Hook the java/Android standard API for analysis.
 - : loaded as the shared library(so) developed using JNI
- COMMUNICATOR
 - : Interactive interface for communication with user
 - : Send/receive values for Hooking, Monitoring, Modifying

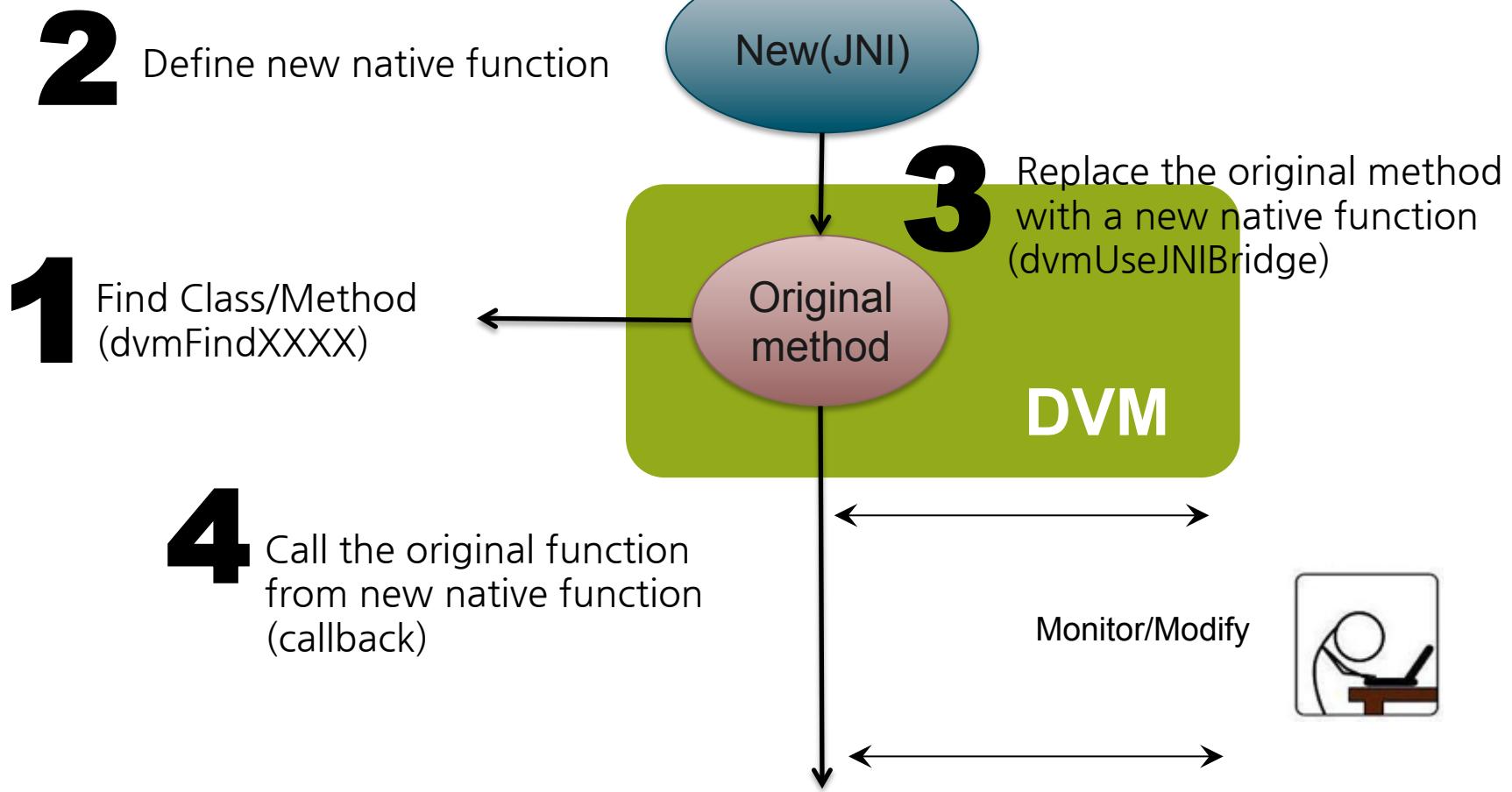
How to implement Android Client

Implemented using **JNI (Java Native Interface)**



- Get the method Information loaded
- Define/Prototype new function(native) for target function(method)
- Call Original Method from new function.
- Monitor/Modify a argument/return value

How to implement ... (Cont'd)



How to implement .. (Cont'd)

```
static void* bp_sb_compareretcase(JNIEnv *env, jobject obj, jobject str)
```

Call <return type> Method

```
    jstring args;
    dalvik_postcall(&d, &sb, env);
    Send_Bp("Before", sb.method_name, (*env)->CallObjectMethodA(env, obj));
    Recv_Bp(&args);

    int res = (*env)->CallIntMethodA(env, obj);
    dalvik_postcall(&d, &sb);

    Send_Bp("After", sb.method_name, res);
    Recv_Bp(&res);

    close(c_sock);

    return res;
}
```

```
jboolean CallBooleanMethod( JNIEnv *env, jobject obj, jbyte value, jbyte args )
jbyte CallByteMethod( JNIEnv *env, jobject obj, jbyte value, jbyte args )
jchar CallCharMethod( JNIEnv *env, jobject obj, jchar value, jchar args )
jdouble CallDoubleMethod( JNIEnv *env, jobject obj, jdouble value, jdouble args )
jfloat CallFloatMethod( JNIEnv *env, jobject obj, jfloat value, jfloat args )
jint CallIntMethod( JNIEnv *env, jobject obj, jint value, jint args )
jlong CallLongMethod( JNIEnv *env, jobject obj, jlong value, jlong args )
jobject CallObjectMethod( JNIEnv *env, jobject obj, jobject value, jobject args )
jshort CallShortMethod( JNIEnv *env, jobject obj, jshort value, jshort args )
void CallVoidMethod( JNIEnv *env, jobject obj, void value, void args )
```

DEMO - PoC for Android App

DEMO

DEMO

Let's Crack
Password-Protection
: Modify RETURN

DEMO

Let's Crack Application
: Monitor ARG. & RETURN

How to make?

Ways of build iOS client & PoC

Key Requirements - How To hook ..

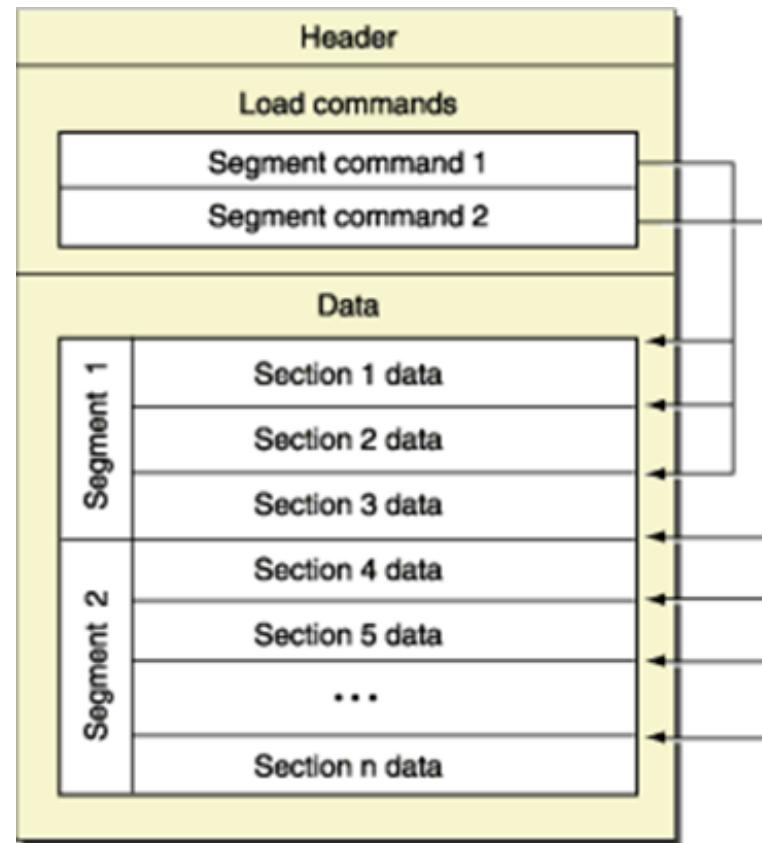
- We can use for hooking in iOS:
 - a. Cydia Substrate for iOS
 - b. fishhook
 - c. Mach-O-Hook

How to implement iOS client

- Use a CydiaSubstrate
 - a. Why CydiaSubstrate?
-> verified stability
- Most of Apps in Cydia are use a CydiaSubstrate!
- Component of CydiaSubstrate
 - a. MobileHooker
 - b. MobileLoader
 - c. Safe Mode

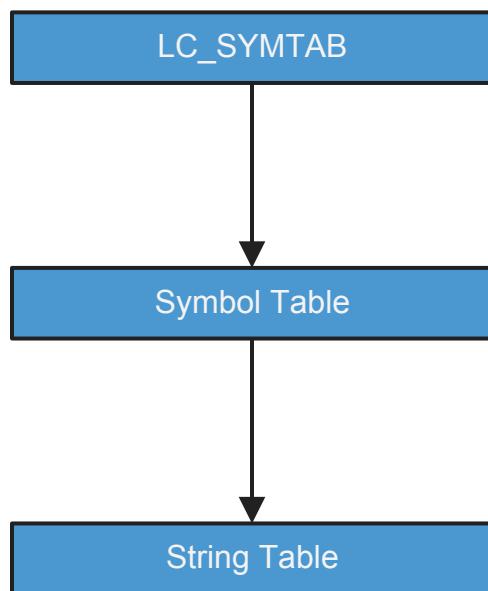
Key Requirements - What & How extract...

- Mach-O File Format



Key Requirements - What & How extract...

- API



| Offset | Data | Description | Value |
|----------|----------|--------------------|-----------------------------------|
| 0000CDCC | 00000CA7 | String Table Index | _socket |
| 0000CDD0 | 01 | Type | |
| | 00 | | N_UNDEF |
| | 01 | | N_EXT |
| 0000CDD1 | 00 | Section Index | NO_SECT |
| 0000CDD2 | 0500 | Description | |
| | 0 | | REFERENCE_FLAG_UNDEFINED_NON_LAZY |
| | | Library Ordinal | 5 (libSystem.B.dylib) |
| 0000CDD4 | 00000000 | Value | 0 |
| 0000CDD8 | 00000CAF | String Table Index | _strcmp |
| 0000CDDC | 01 | Type | |
| | 00 | | N_UNDEF |
| | 01 | | N_EXT |
| 0000CDDD | 00 | Section Index | NO_SECT |
| 0000CDDE | 0500 | Description | |
| | 0 | | REFERENCE_FLAG_UNDEFINED_NON_LAZY |
| | | Library Ordinal | 5 (libSystem.B.dylib) |
| 0000CDE0 | 00000000 | Value | 0 |

Key Requirements - What & How extract...

- Objective-C and User Function Address

The diagram illustrates a flow from the `LC_FUNCTION_STARTS` section header to the `Function Starts`. An arrow points downwards from the `LC_FUNCTION_STARTS` box to the `Function Starts` box.

| Offset | Data | Description | Value |
|----------|--------|-------------|--------|
| 0000C4EC | 95D301 | uleb128 | 0xA995 |
| 0000C4EF | BC03 | uleb128 | 0xAB51 |
| 0000C4F1 | 44 | uleb128 | 0xAB95 |
| 0000C4F2 | 44 | uleb128 | 0xABD9 |
| 0000C4F3 | 74 | uleb128 | 0xAC4D |
| 0000C4F4 | D403 | uleb128 | 0xAE21 |
| 0000C4F6 | 28 | uleb128 | 0xAE49 |
| 0000C4F7 | 30 | uleb128 | 0xAE79 |
| 0000C4F8 | 28 | uleb128 | 0AEA1 |
| 0000C4F9 | 30 | uleb128 | 0xAEA1 |
| 0000C4FA | 3C | uleb128 | 0xAF0D |
| 0000C4FB | 60 | uleb128 | 0xAF6D |
| 0000C4FC | 30 | uleb128 | 0xAF9D |
| 0000C4FD | 30 | uleb128 | 0xAFCD |
| 0000C4FE | 30 | uleb128 | 0AFFD |
| 0000C4FF | 30 | uleb128 | 0xB02D |
| 0000C500 | 30 | uleb128 | 0xB05D |
| 0000C501 | 1C | uleb128 | 0xB079 |
| 0000C502 | 2C | uleb128 | 0xB0A5 |
| 0000C503 | 2C | uleb128 | 0xB0D1 |
| 0000C504 | 9001 | uleb128 | 0xB161 |

| | | | |
|---|---------------------------------------|-----|----------|
| f | _SendLoginCheck | ... | 0000A994 |
| f | -[ViewController viewDidLoad] | ... | 0000AB50 |
| f | -[ViewController didReceiveMemo...] | ... | 0000AB94 |
| f | -[ViewController textFieldShouldR...] | ... | 0000ABD8 |
| f | -[ViewController LoginButton:] | ... | 0000AC4C |
| f | -[ViewController loginPassword] | ... | 0000AE20 |
| f | -[ViewController setLoginPassword:] | ... | 0000AE48 |
| f | -[ViewController infoMsg] | ... | 0000AE78 |
| f | -[ViewController setInfoMsg:] | ... | 0000AEAO |
| f | -[ViewController .cxx_destruct] | ... | 0000AED0 |
| f | -[AppDelegate application:didFinis... | ... | 0000AF0C |
| f | -[AppDelegate applicationWillResi... | ... | 0000AF6C |
| f | -[AppDelegate applicationWillEnte... | ... | 0000AF9C |
| f | -[AppDelegate applicationWillEnte... | ... | 0000AFCC |
| f | -[AppDelegate applicationWillBeC... | ... | 0000AFFC |
| f | -[AppDelegate applicationWillTer... | ... | 0000B02C |
| f | -[AppDelegate window] | ... | 0000B05C |
| f | -[AppDelegate setWindow:] | ... | 0000B078 |
| f | -[AppDelegate .cxx_destruct] | ... | 0000B0A4 |
| f | _main | ... | 0000B0D0 |
| f | _objc_autoreleaseReturnValue\$shim | ... | 0000B160 |

How to implement iOS client

- Target API and method selection
 - a. Extracting Objective C classes & methods
 - b. Extracting API lists
 - c. Finding out user-defined function's args and types
- Monitoring an entire method and API by using hooking (Logging?) (Logging?)

DEMO - PoC for iOS App

DEMO



Anything else?

Future Works

Implementation Methods

- How to obtain a target application's function list and detail informations of the function
- How to utilize database information to distinct functions

Additional Functions

- arbitrary function execution
- arbitrary code execution
- memory scan and patch
- function control based on script languages
- disassemble and decompilation

And...

- Performance Improvement
- Additional OS Support

谢谢

Any Other Questions or Comments?

email : binproxy@0-day.me