

# 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 applicaton
- PE, ELF, etc.

- **Web Proxy**

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

# What's wrong?

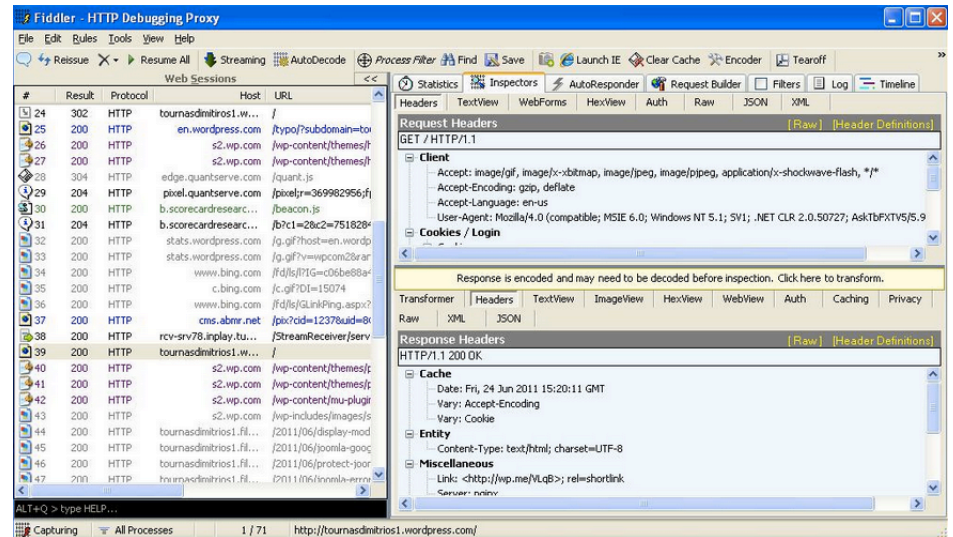
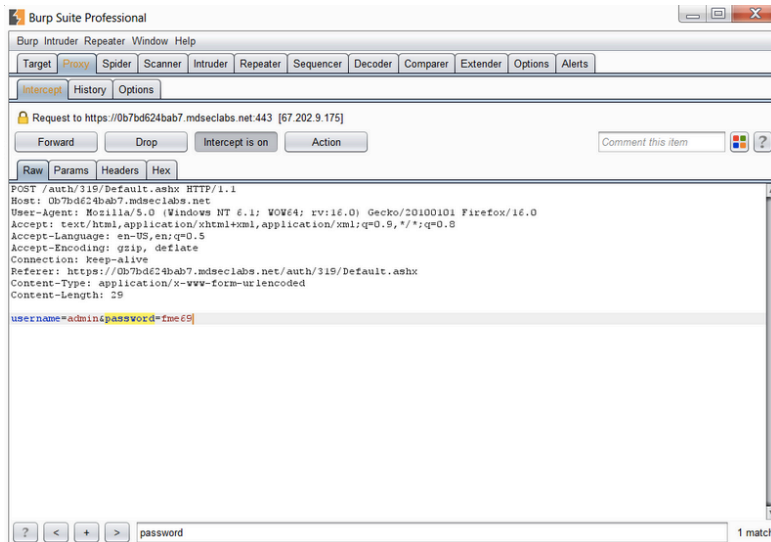
Background



# 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





# Challenges for application analysis

We cannot  
save the earth  
using our resources



Lack of time and manpower

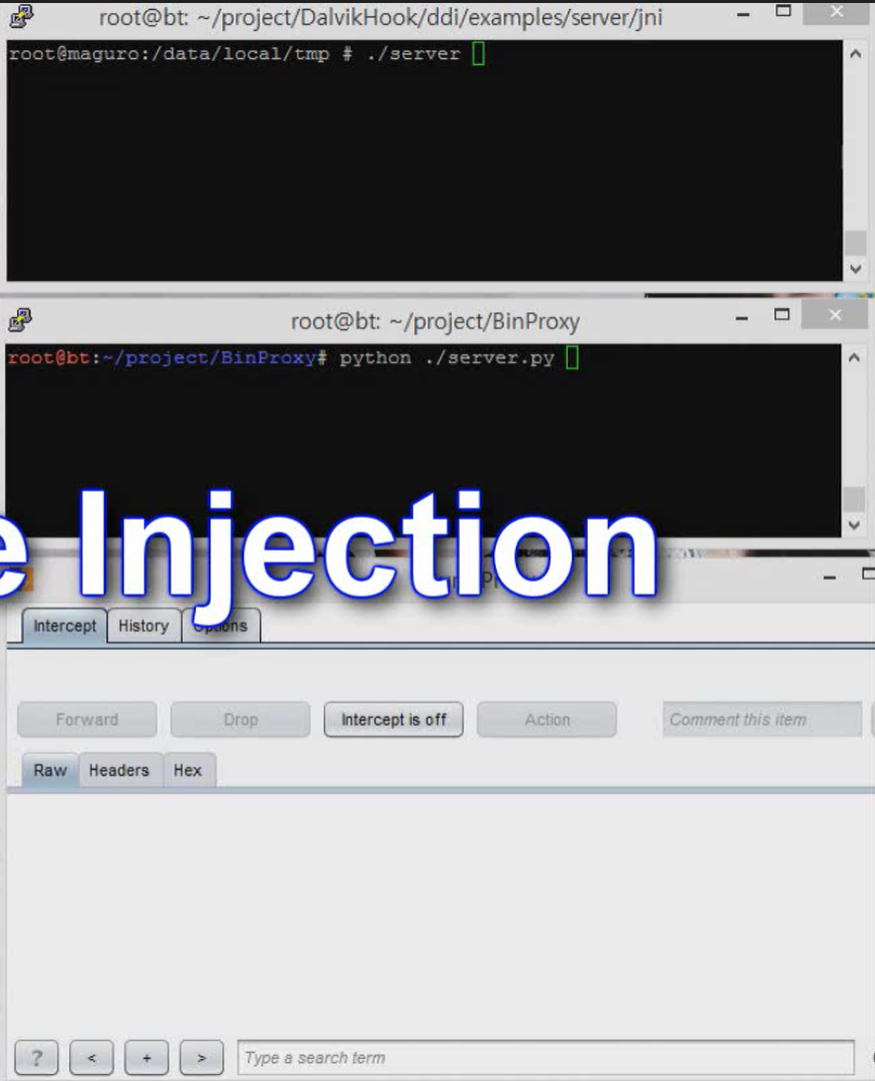
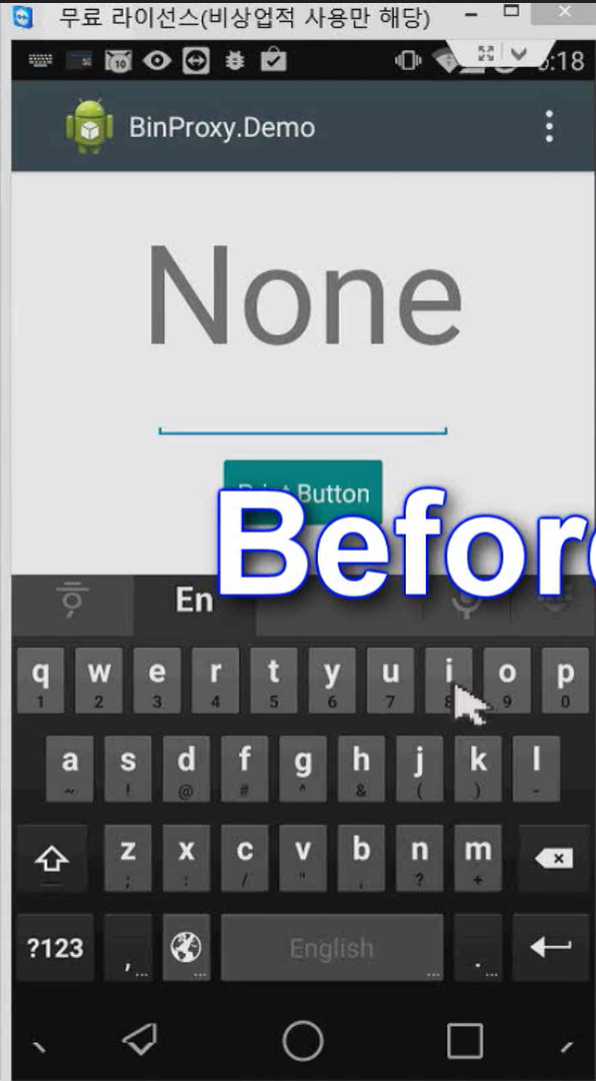
# How to solve a problem?



Need a **EASY** tool

# So what??

BinProxy : A New Paradigm for Binary  
Analysis

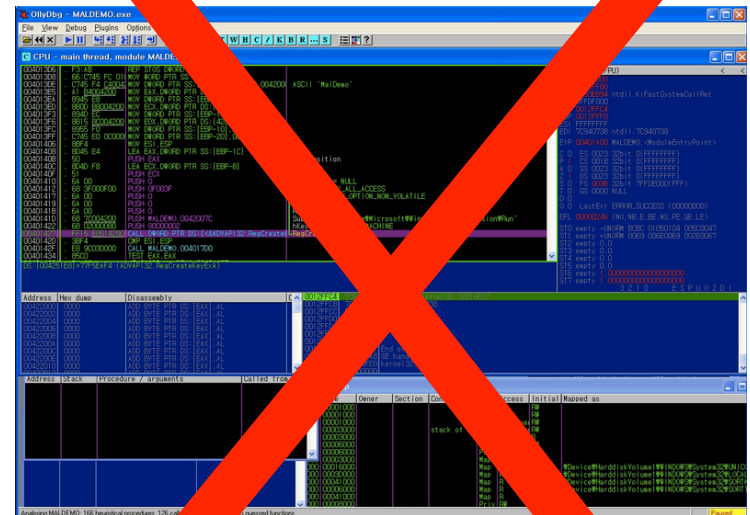


# Before Injection

# Key Features

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

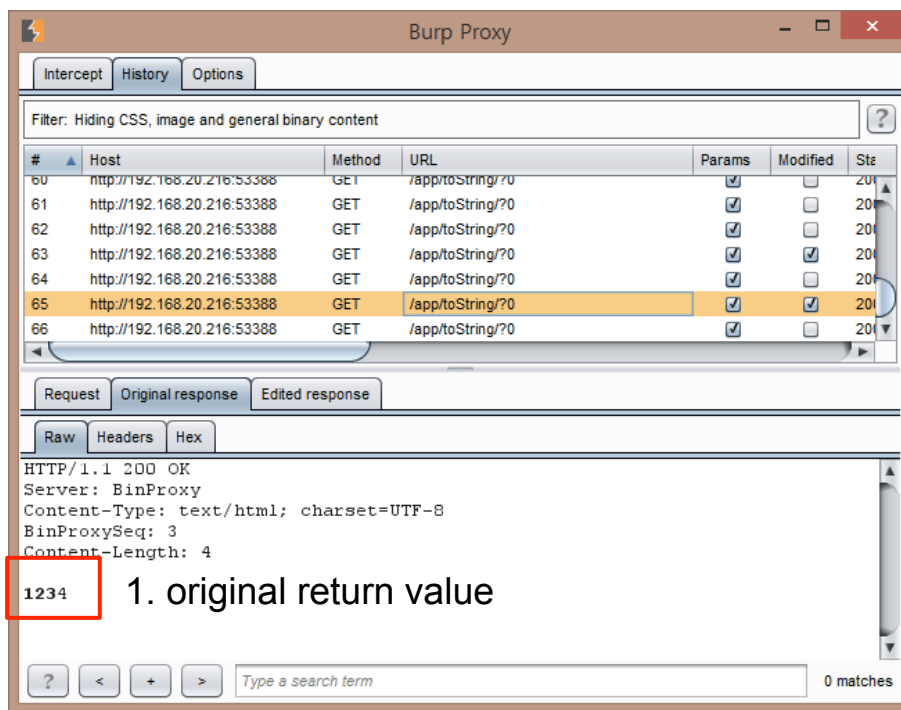
```
(gdb) file lab2
Reading symbols from C:\Users\thomas.schmid\lab2\lab2.exe done.
(gdb) # Invoke debug mode in Cortex-M3 mode
(gdb) target remote :1234
Remote debugging using :1234
arm-none-eabi-sprite: Target memory map ./memory_map.xml
arm-none-eabi-sprite: Target reset
0x6008051c in ?? (<)
Memory access to just regions defined in linker script
(gdb) set mem inaccessible-by-default off
(gdb) # Disable the watchdog
(gdb) #set *0x40006010 = 0x4C6
(gdb) # Specify user application path to file
(gdb) set *0xE000ED08 = 0x2000000
(gdb) # Load the program
(gdb) load
Loading section .text, size 0x98, address 0x20000000
Loading section .data, size 0x4, address 0x20000098
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 (<)
(gdb)
```





# 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**



Intercept History Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Modified	Sta
60	http://192.168.20.216:53388	GET	/app/toString/?u	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200
61	http://192.168.20.216:53388	GET	/app/toString/?0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200
62	http://192.168.20.216:53388	GET	/app/toString/?0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200
63	http://192.168.20.216:53388	GET	/app/toString/?0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	200
64	http://192.168.20.216:53388	GET	/app/toString/?0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200
65	http://192.168.20.216:53388	GET	/app/toString/?0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	200
66	http://192.168.20.216:53388	GET	/app/toString/?0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200

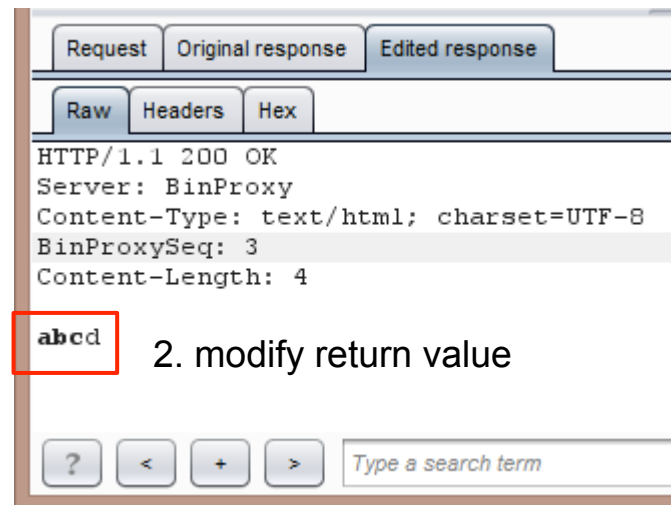
Request Original response Edited response

Raw Headers Hex

```
HTTP/1.1 200 OK
Server: BinProxy
Content-Type: text/html; charset=UTF-8
BinProxySeq: 3
Content-Length: 4
```

1234

1. original return value



Request Original response Edited response

Raw Headers Hex

```
HTTP/1.1 200 OK
Server: BinProxy
Content-Type: text/html; charset=UTF-8
BinProxySeq: 3
Content-Length: 4
```

abcd

2. modify return value

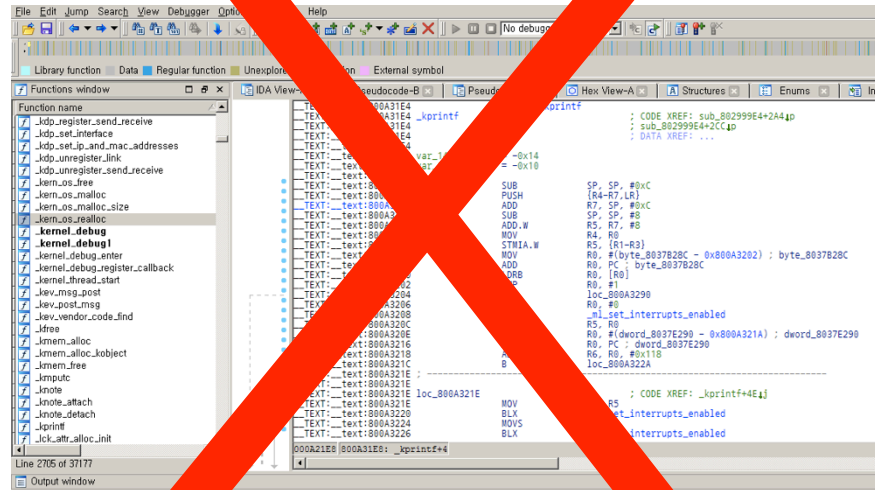
? < + > Type a search term

3. Click Forward Button



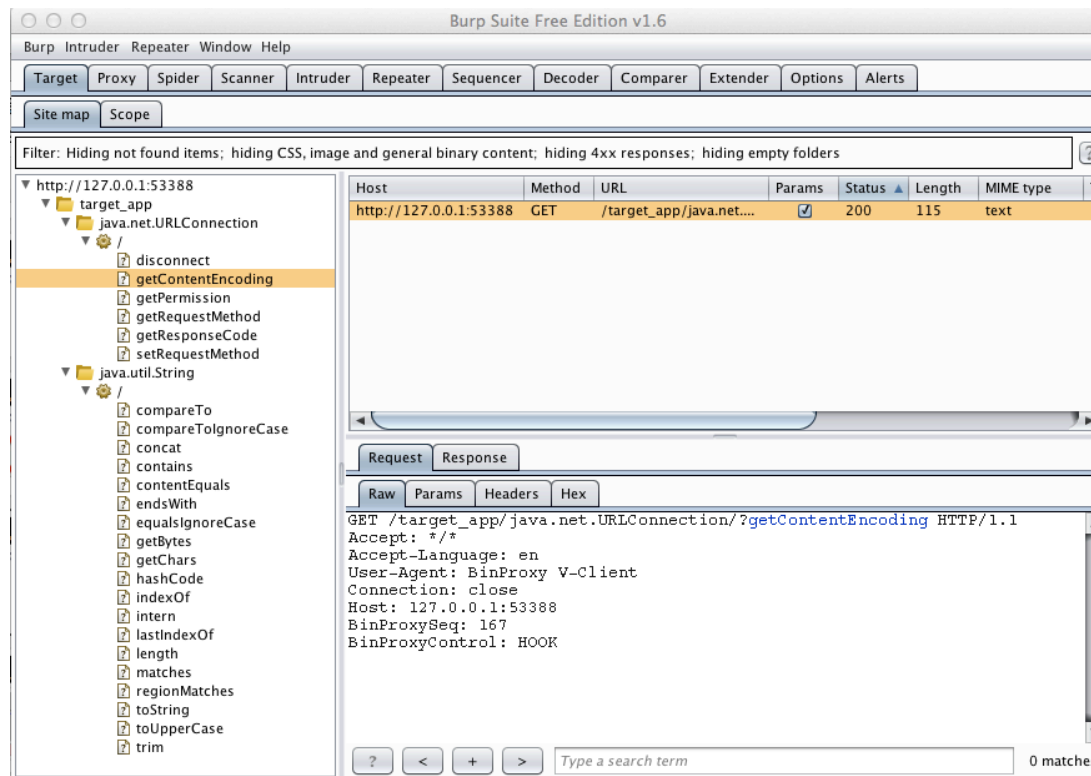
# Key Features (cont'd)

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

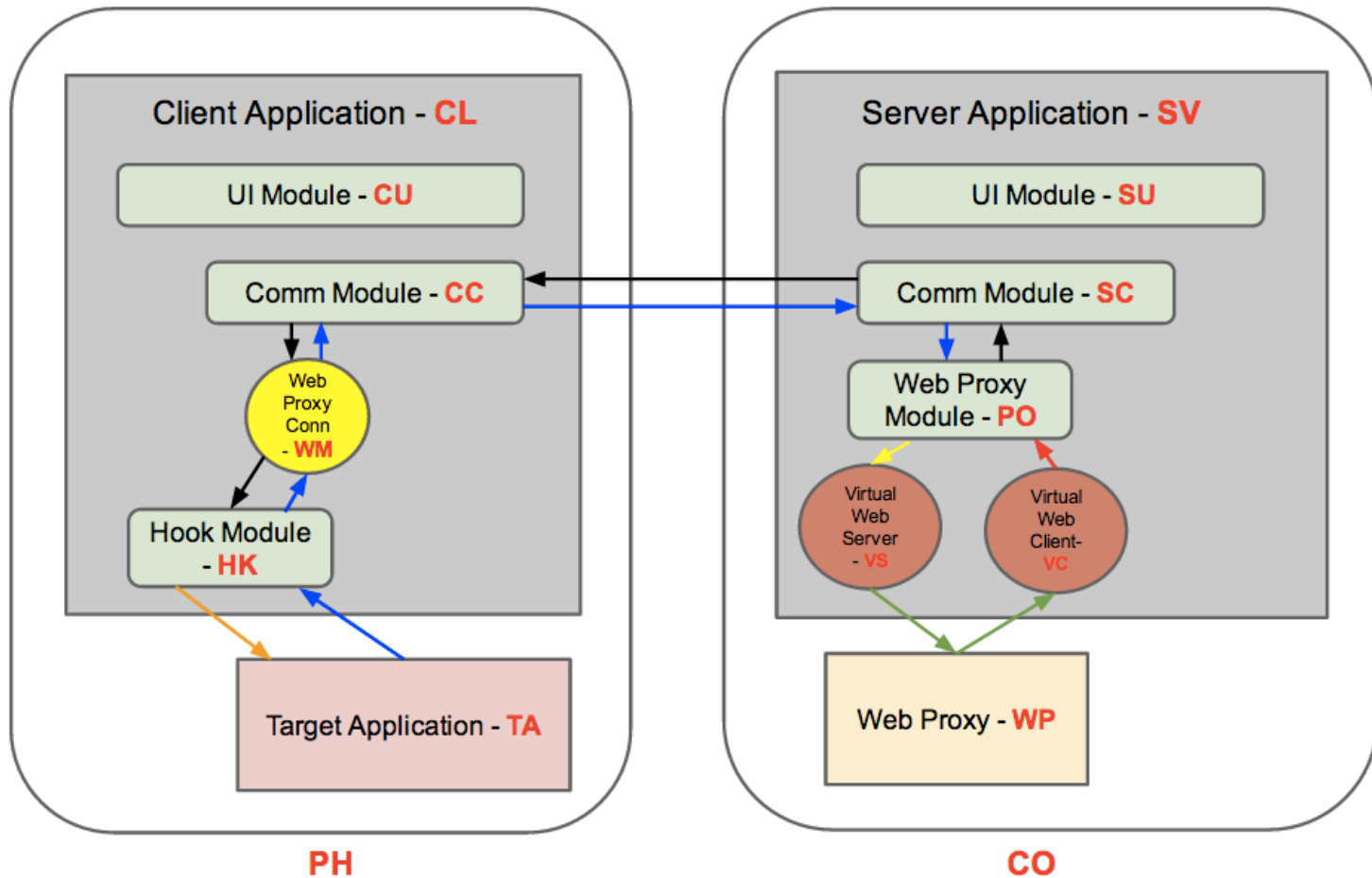


# Key Features (cont'd)

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



# 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

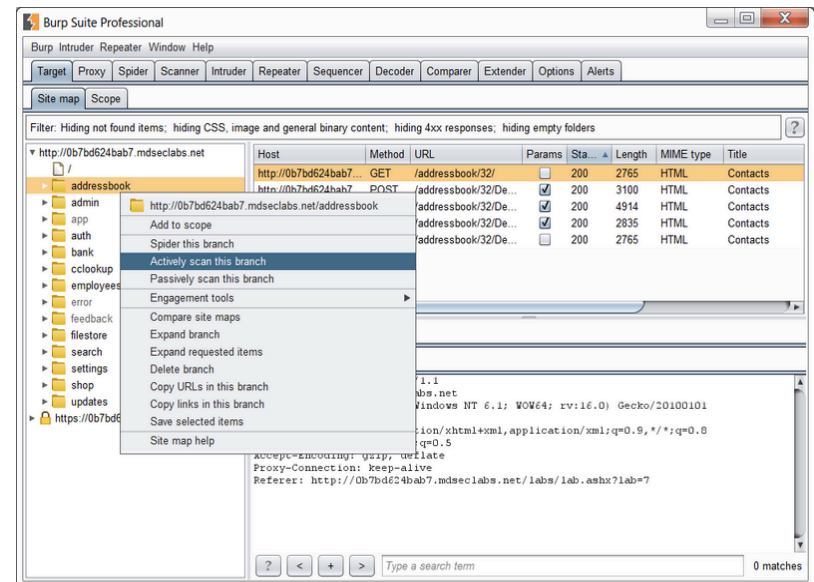
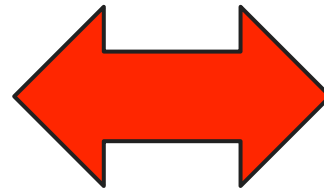
how to control function calls by using web proxy

## Convert Functions

```
haking@live:/ramdisk/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 lib
rary "/lib/tls/libthread_db.so.1".

(gdb) list
1 void fn(char *a) {
2 char buff[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 0x8048382: file stack_1.c, line 3.
(gdb) run AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Starting program: /ramdisk/home/haking/stack_1 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Breakpoint 1, fn (a=0xbffffc2b 'A' (repeats 30 times)) at stack_1.c:3
3 strcpy(buf, a);
(gdb)
```



# 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 / refernce**

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_entropy	8001FA78
 _ml_stack_remaining	8001FA90
 _current_thread	8001FAE0
 _enable_kernel_vfp_context	8001FBB4
 _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
 _strlen	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+Ch] [bp-14h]@1
9     int v13; // [sp+10h] [bp-10h]@1
10    int v14; // [sp+14h] [bp-Ch]@1
11    int v15; // [sp+18h] [bp-8h]@1
```

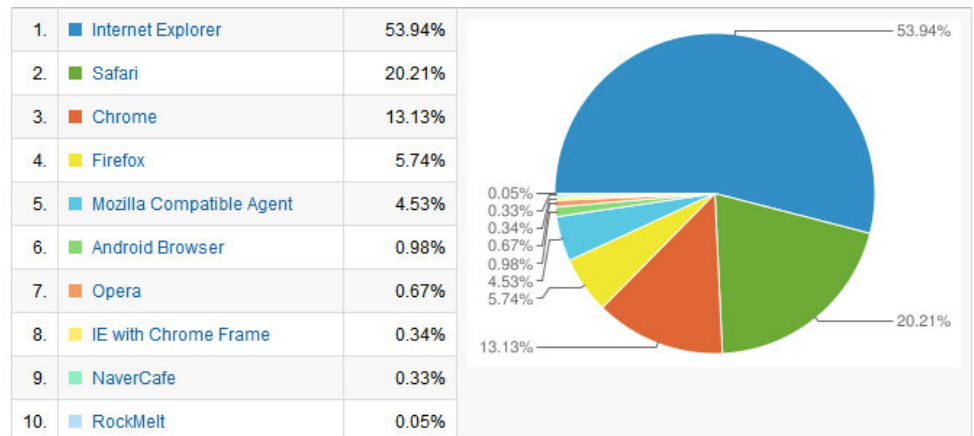
f	sub_805B2E94
f	sub_805B2EA4
f	sub_805B2EB4
f	sub_805B2EC4
f	sub_805B2ED4
f	sub_805B2EE4
f	sub_805B2EF4
f	sub_805B2F04
f	sub_805B2F14
f	sub_805B2F24
f	sub_805B2F34
f	sub_805B2F44
f	sub_805B2F54
f	sub_805B2F64
f	sub_805B2F84
f	sub_805B2F94
f	sub_805B2FA4
f	sub_805B2FB4
f	sub_805B2FE4
f	sub_805B2FF4
f	sub_805B3004
f	sub_805B3014
f	sub_805B3024
f	sub_805B3034
f	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 SeRect(prc: 0x0000000000000000)PDC070: (left=14,right=588,top=237,bottom=259),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x000000000000454E9
5951 18-48:58:660:248,4 SeRect(prc: 0x0000000000000000)PDC0A0: (left=14,right=588,top=231,bottom=253),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x00000000000045916
5952 18-48:58:660:447,5 GeDC(pWind: 0x0000000000000000) 0xFFFFFFFF90111DE explorer.frame.dll + 0x0000000000004A26F
5953 18-48:58:660:544,2 ReleaseDC(pWind: 0x0000000000000000)JDC: 0xFFFFFFFF90111DE 0x0000000000000000 explorer.frame.dll + 0x0000000000004A339
5954 18-48:58:660:640,8 SeRect(prc: 0x0000000000000000)PDC0F0: (left=14,right=588,top=252,bottom=274),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x00000000000045916
5955 18-48:58:660:739,3 SeRect(prc: 0x0000000000000000)PDC070: (left=14,right=588,top=258,bottom=280),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x000000000000454E9
5956 18-48:58:660:837,7 SeRect(prc: 0x0000000000000000)PDC0A0: (left=14,right=588,top=252,bottom=274),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x00000000000045916
5957 18-48:58:660:936,5 GeDC(pWind: 0x0000000000000000) 0xFFFFFFFF90111DE explorer.frame.dll + 0x0000000000004A26F
5958 18-48:58:661:033,8 ReleaseDC(pWind: 0x0000000000000000)JDC: 0xFFFFFFFF90111DE 0x0000000000000000 explorer.frame.dll + 0x0000000000004A339
5959 18-48:58:661:130,4 SeRect(prc: 0x0000000000000000)PDC0F0: (left=14,right=588,top=273,bottom=295),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x00000000000045916
5960 18-48:58:661:246,9 SeRect(prc: 0x0000000000000000)PDC070: (left=14,right=588,top=279,bottom=301),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x000000000000454E9
5961 18-48:58:661:358,0 SeRect(prc: 0x0000000000000000)PDC0A0: (left=14,right=588,top=273,bottom=295),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x00000000000045916
5962 18-48:58:661:463,0 GeDC(pWind: 0x0000000000000000) 0xFFFFFFFF90111DE explorer.frame.dll + 0x0000000000004A26F
5963 18-48:58:661:562,7 ReleaseDC(pWind: 0x0000000000000000)JDC: 0xFFFFFFFF90111DE 0x0000000000000000 explorer.frame.dll + 0x0000000000004A339
5964 18-48:58:661:662,4 SeRect(prc: 0x0000000000000000)PDC0F0: (left=14,right=588,top=294,bottom=316),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x00000000000045916
5965 18-48:58:661:762,0 SeRect(prc: 0x0000000000000000)PDC070: (left=14,right=588,top=300,bottom=322),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x000000000000454E9
5966 18-48:58:661:860,2 SeRect(prc: 0x0000000000000000)PDC0A0: (left=14,right=588,top=294,bottom=316),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x00000000000045916
5967 18-48:58:661:960,4 GeDC(pWind: 0x0000000000000000) 0xFFFFFFFF90111DE explorer.frame.dll + 0x0000000000004A26F
5968 18-48:58:662:055,6 ReleaseDC(pWind: 0x0000000000000000)JDC: 0xFFFFFFFF90111DE 0x0000000000000000 explorer.frame.dll + 0x0000000000004A339
5969 18-48:58:662:152,9 SeRect(prc: 0x0000000000000000)PDC0F0: (left=14,right=588,top=315,bottom=337),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x00000000000045916
5970 18-48:58:662:251,1 SeRect(prc: 0x0000000000000000)PDC070: (left=14,right=588,top=321,bottom=343),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x000000000000454E9
5971 18-48:58:662:353,7 SeRect(prc: 0x0000000000000000)PDC0A0: (left=14,right=588,top=315,bottom=337),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x00000000000045916
5972 18-48:58:662:451,9 GeDC(pWind: 0x0000000000000000) 0xFFFFFFFF90111DE explorer.frame.dll + 0x0000000000004A26F
5973 18-48:58:662:550,0 ReleaseDC(pWind: 0x0000000000000000)JDC: 0xFFFFFFFF90111DE 0x0000000000000000 explorer.frame.dll + 0x0000000000004A339
5974 18-48:58:662:646,4 SeRect(prc: 0x0000000000000000)PDC0F0: (left=14,right=588,top=336,bottom=358),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x00000000000045916
5975 18-48:58:662:744,8 SeRect(prc: 0x0000000000000000)PDC070: (left=14,right=588,top=342,bottom=364),x:1:0x000... 0x0000000000000000 explorer.frame.dll + 0x000000000000454E9
5976 18-48:58:662:844,8 SeRect(prc: 0x0000000000000000)PDC0A0: (left=14,right=588,top=336,bottom=358),x:1:0x000... 0x0000000000000000 explorer.frame.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.

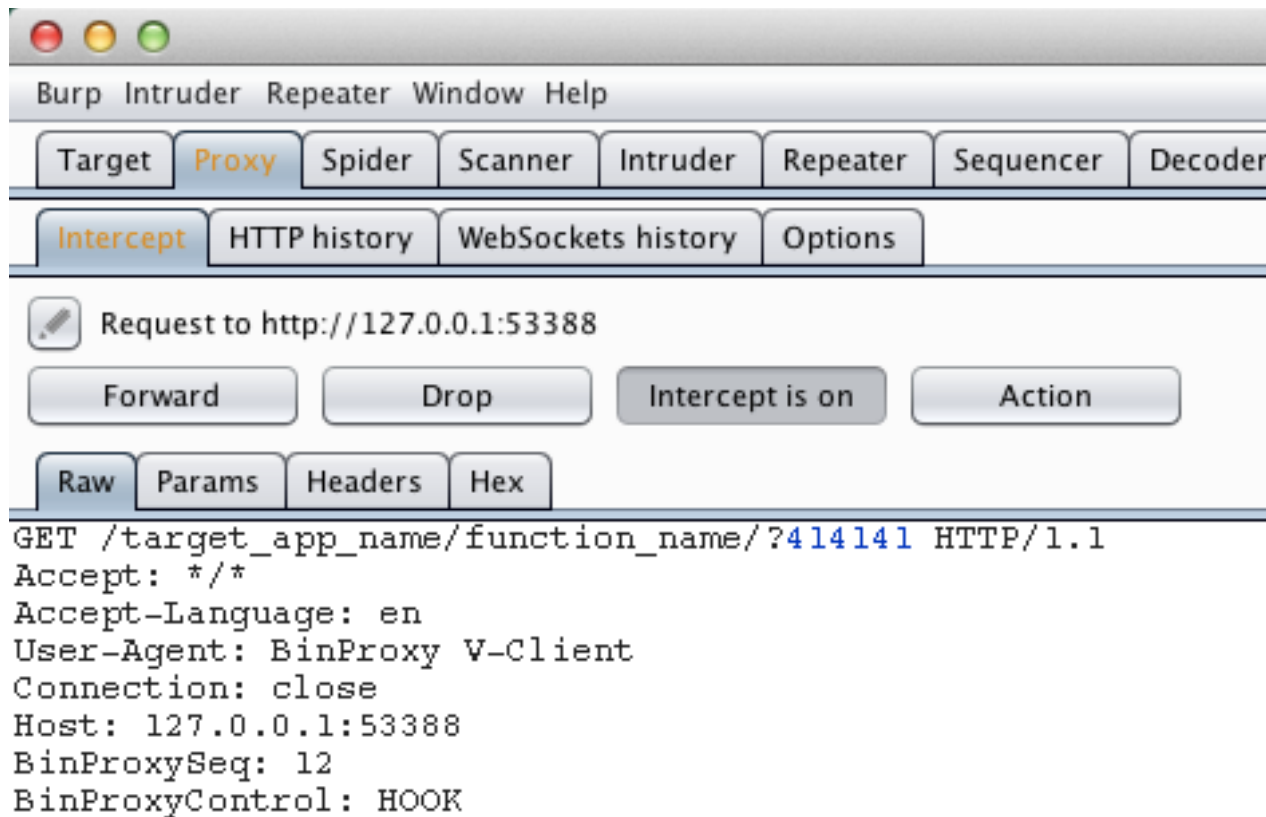
```
POST http://127.0.0.1:53388/function_name
```

```
Host: target_app_name
```

```
User-Agent: BinProxy
```

```
01_414141
```

# How to interwork with a web proxy – Web Proxy





# 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
```

0

# How to interwork with a web proxy – Web Proxy

The screenshot displays the Burp Suite interface. At the top, there is a menu bar with options: Burp, Intruder, Repeater, Window, and Help. Below the menu bar is a toolbar with buttons for Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Options, and Alerts. A second row of buttons includes Intercept, HTTP history, WebSockets history, and Options. A filter bar indicates 'Filter: Hiding CSS, image and general binary content'. The main area contains a table of HTTP history entries:

#	Host	Method	URL	Params	Edited	Status
67	http://127.0.0.1:53388	GET	/target_app_name/function_name/?414141	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200

Below the table, there are tabs for 'Request' and 'Response'. Underneath, there are tabs for 'Raw', 'Headers', and 'Hex'. The 'Raw' tab is selected, showing the following text:

```
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



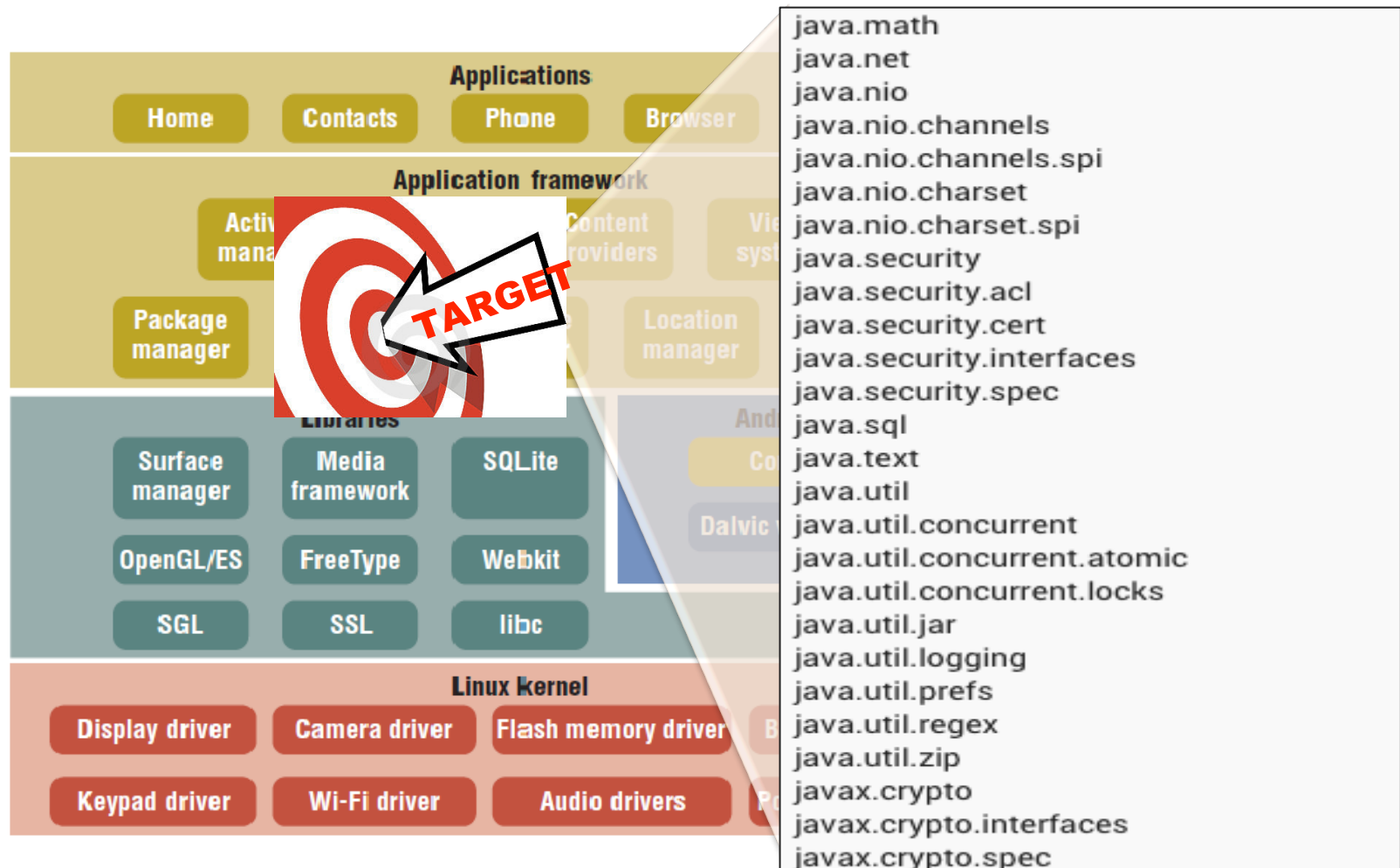
**Function**

How to Hook ???

What & How  
To Extract ???

# Key Requirements

## - What & How To extract ..



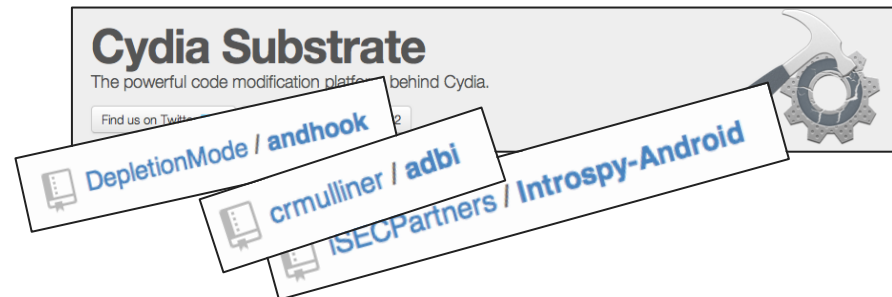
# Key Requirements

## - What & How To extract ..

We can use for hooking in Android :

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

....





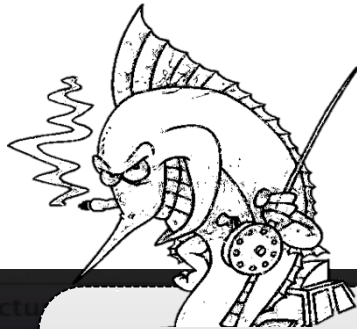
# ADBI

## (Android Binary Instrumentation Toolkit)

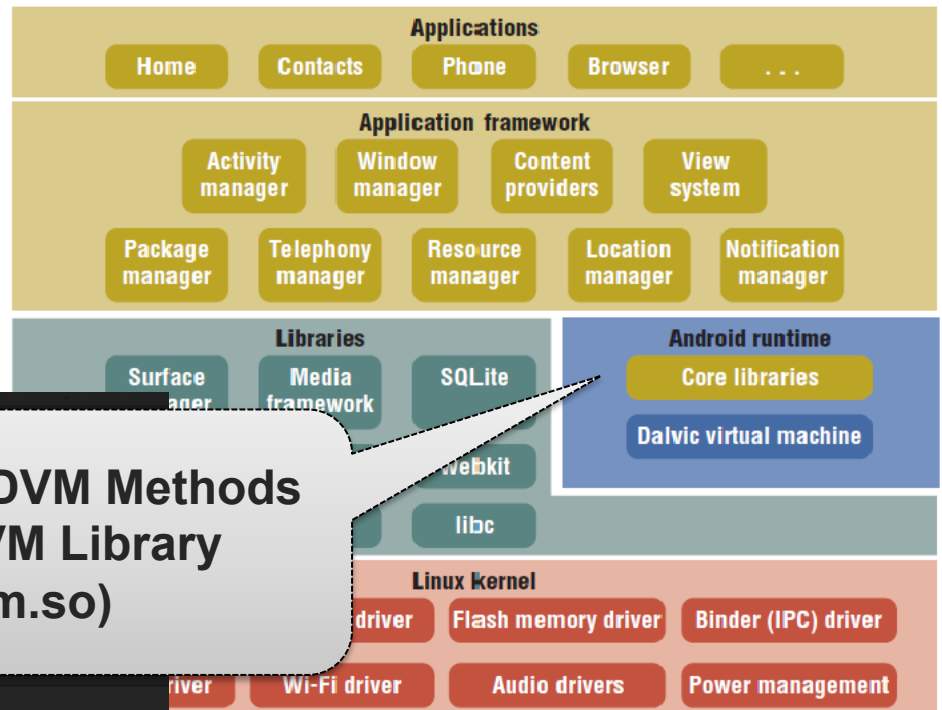
Dynamic Dalvik Instrumentation Framework for Android (old)

- Collin Mulliner, SummerCon 2013.

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

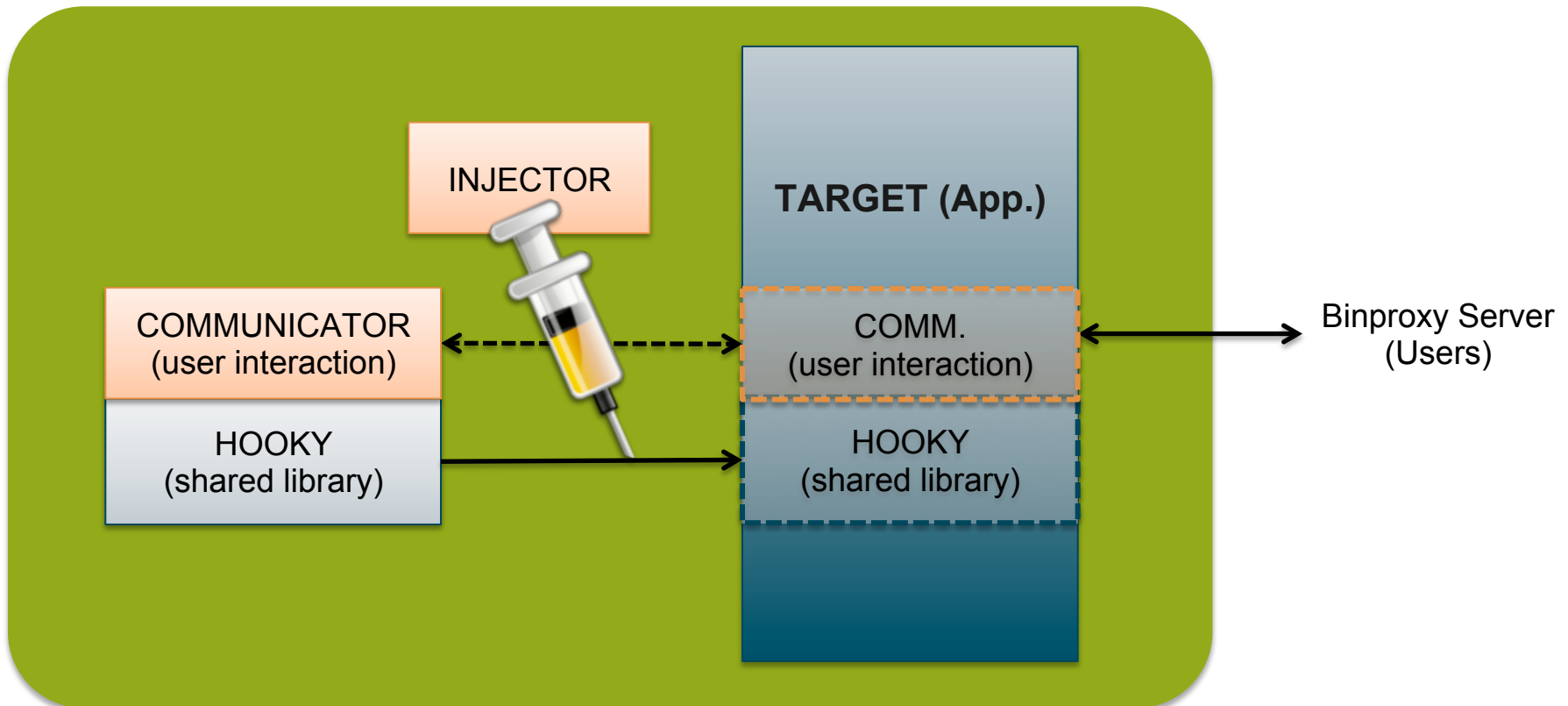


**Intercept/Use DVM Methods  
on Dalvik VM Library  
(libdvm.so)**



```
... Format Picture
dvmUseJNIBridge
dvmFindDirectMe
dvmFindVirtualM
dvmFindLoadedCl
dumpMethods
dvmFindClassByNa
...
```

# Binproxy Client modules for Android

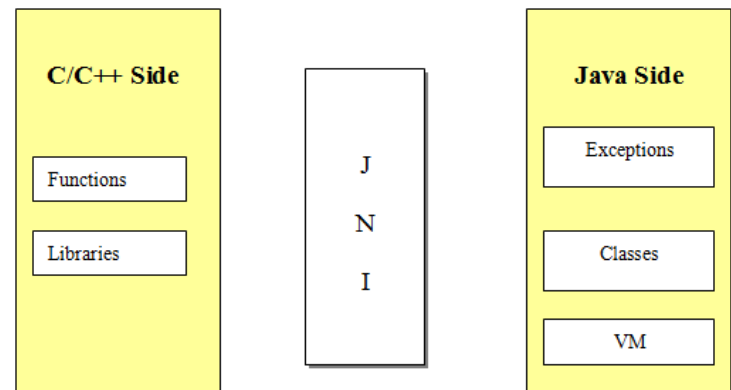


# Binproxy Client modules ... (cont'd)

- 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

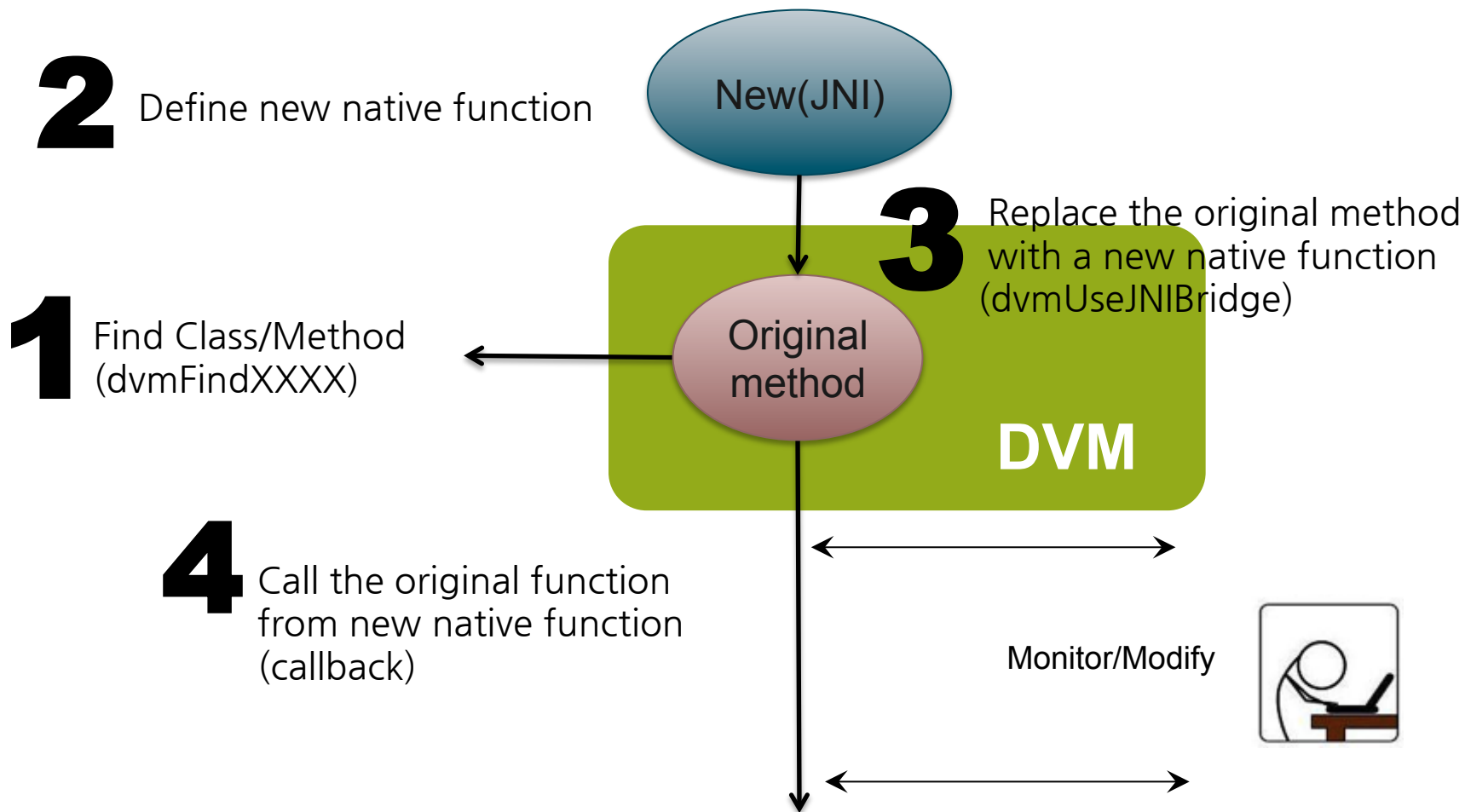
# How to implement Android Client

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_comparetoCase(JNIEnv *env, jobject obj, jobject str)
```

Call <return type> Method

```
    jboolean CallBooleanMethod( JNIEnv *env, jobject obj, jstring str, jbooleanArray args )
    jbyte CallByteMethod( JNIEnv *env, jobject obj, jstring str, jbyteArray args )
    jchar CallCharMethod( JNIEnv *env, jobject obj, jstring str, jcharArray args )
    jdouble CallDoubleMethod( JNIEnv *env, jobject obj, jstring str, jdoubleArray args )
    jfloat CallFloatMethod( JNIEnv *env, jobject obj, jstring str, jfloatArray args )
    jint CallIntMethod( JNIEnv *env, jobject obj, jstring str, jintArray args )
    jlong CallLongMethod( JNIEnv *env, jobject obj, jstring str, jlongArray args )
    jobject CallObjectMethod( JNIEnv *env, jobject obj, jstring str, jobjectArray args )
    jshort CallShortMethod( JNIEnv *env, jobject obj, jstring str, jshortArray args )
    void CallVoidMethod( JNIEnv *env, jobject obj, jstring str, voidArray args )

    Send_Bp("Before", sb.method_name, (*env)->CallIntMethodA(env, obj, str, &args));
    Recv_Bp(&args);

    int res = (*env)->CallIntMethodA(env, obj, str, &args);
    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, jstring str, jbooleanArray args )
jbyte CallByteMethod( JNIEnv *env, jobject obj, jstring str, jbyteArray args )
jchar CallCharMethod( JNIEnv *env, jobject obj, jstring str, jcharArray args )
jdouble CallDoubleMethod( JNIEnv *env, jobject obj, jstring str, jdoubleArray args )
jfloat CallFloatMethod( JNIEnv *env, jobject obj, jstring str, jfloatArray args )
jint CallIntMethod( JNIEnv *env, jobject obj, jstring str, jintArray args )
jlong CallLongMethod( JNIEnv *env, jobject obj, jstring str, jlongArray args )
jobject CallObjectMethod( JNIEnv *env, jobject obj, jstring str, jobjectArray args )
jshort CallShortMethod( JNIEnv *env, jobject obj, jstring str, jshortArray args )
void CallVoidMethod( JNIEnv *env, jobject obj, jstring str, voidArray 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 ..

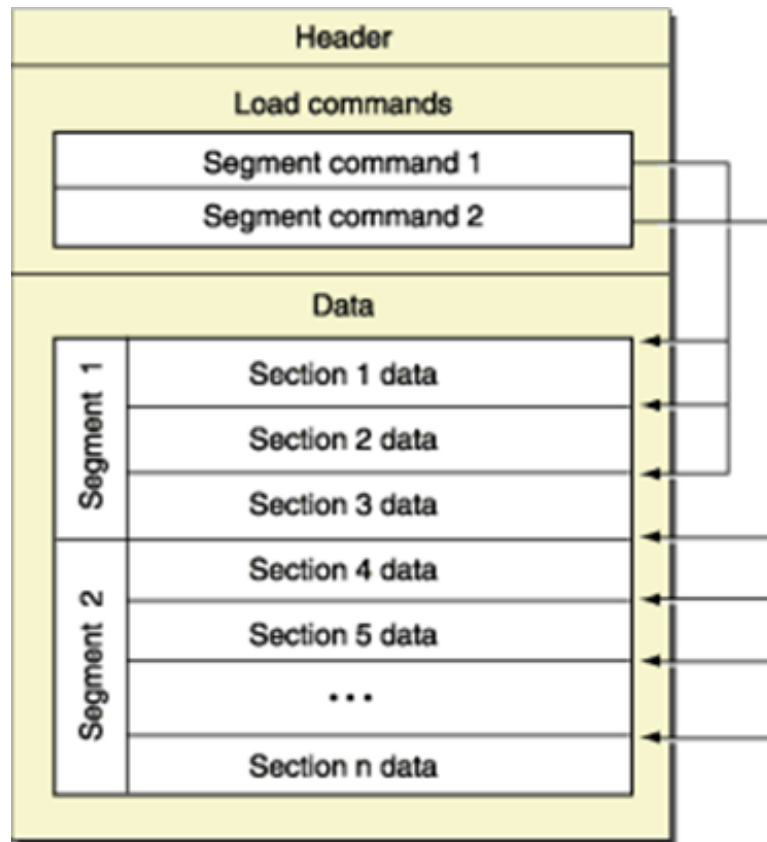
- We can use for hooking in iOS:
  - a. Cydia Substrate for iOS
  - b. fishhook
  - c. Mach-O-Hook

# 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

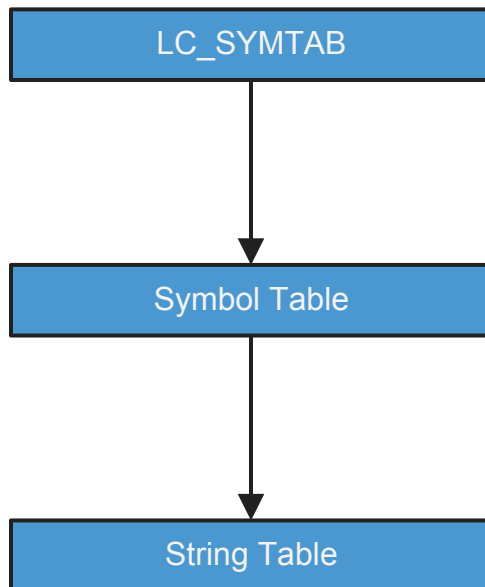
# Key Requirements - What & How extract...

- Mach-O File Format



# Key Requirements - What & How extract...

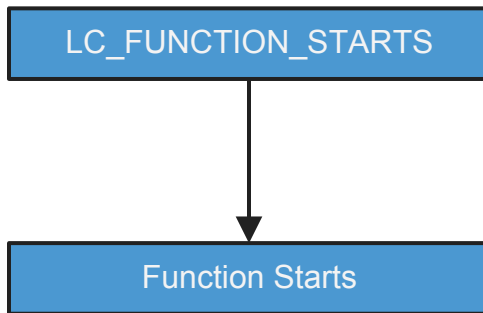
- API



Offset	Data	Description	Value
0000CDCC	00000CA7	String Table Index	_socket
0000CDD0	01	Type	
		00	N_UNDF
		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_UNDF
		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



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	0xAEA1
0000C4F9	30	uleb128	0xAED1
0000C4FA	3C	uleb128	0xAF0D
0000C4FB	60	uleb128	0xAF6D
0000C4FC	30	uleb128	0xAF9D
0000C4FD	30	uleb128	0xAFCD
0000C4FE	30	uleb128	0xAFFD
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:]	...	0000AEA0
f	-[ViewController .cxx_destruct]	...	0000AED0
f	-[AppDelegate application:didFini...	...	0000AF0C
f	-[AppDelegate applicationWillResi...	...	0000AF6C
f	-[AppDelegate applicationDidEnte...	...	0000AF9C
f	-[AppDelegate applicationWillEnte...	...	0000AFCC
f	-[AppDelegate applicationDidBec...	...	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**

```

Python — 50x22
iksui-Mac:Desktop iks$ python SampleApp_server.py
('Server Port : ', 8888)

```



```

bash — 50x22
iksui-Mac:Desktop iks$

```



# 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](mailto:binproxy@0-day.me)