



# Fuzzing For Fun

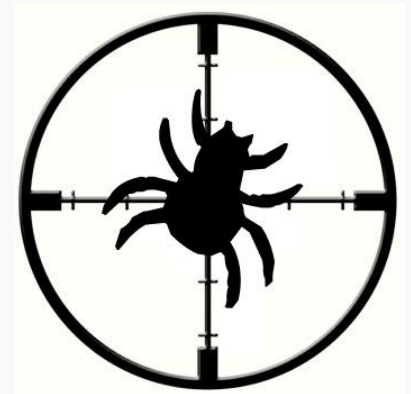
---

현성원 (sweetchip)



# Whoami

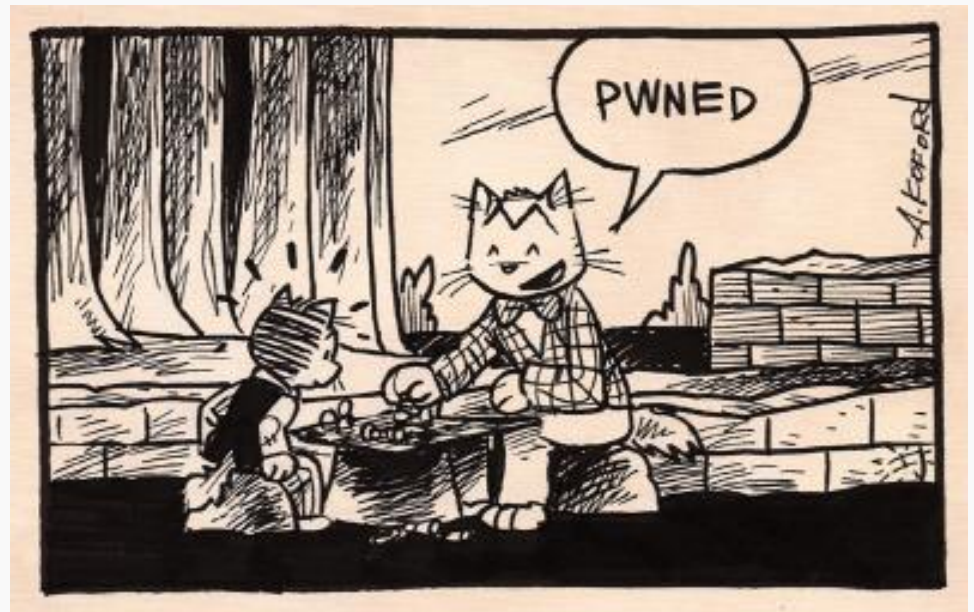
- 현성원 (sweetchip)
- 20 (고3 탈출!)
- 세종대학교 SSG | BOB 2<sup>nd</sup> | HackerSchool – WiseGuys
- 블로그 : <http://blog.sweetchip.kr>
- 페이스북 : <http://fb.me/sweetchipp>
- 윈도우 어플리케이션 취약점 연구 (since 2013)
  - Word Processor
  - Media Player
  - Web Browser
  - ETC
- 삽질 2년차





# Agenda

- Whoami
- Base
  - What is software vulnerability?
  - How to find vulnerability
- Fuzzing
  - Fuzzing For Fun
  - About Fuzzing
  - Let's fuzz
  - Analysis & Exploit
- Bug-Bounty
  - About Bug Bounty
- Tips For Beginner



# Agenda

- PPT는 발표 이후 아래 링크에서 다운로드 받으실 수 있습니다.
- <http://blog.sweetchip.kr/>

[illegible]

- 취약점 (Vulnerability)
  - 개발자의 실수, 프로그램의 결함을 이용하여 프로그램의 실행 흐름을 강제로 바꿀 수 있도록 하는 요소
- 익스플로잇 (exploit)
  - 취약점을 이용하여 실행 흐름을 바꿀 수 있도록 하는 과정 또는 코드





# What is software vulnerability?

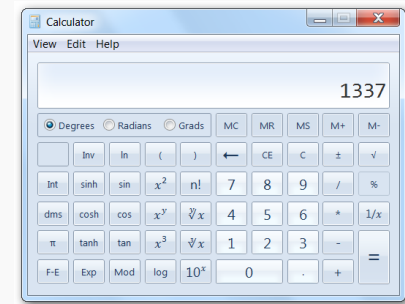
- 취약점의 특이점



Malicious mp3



Media Player



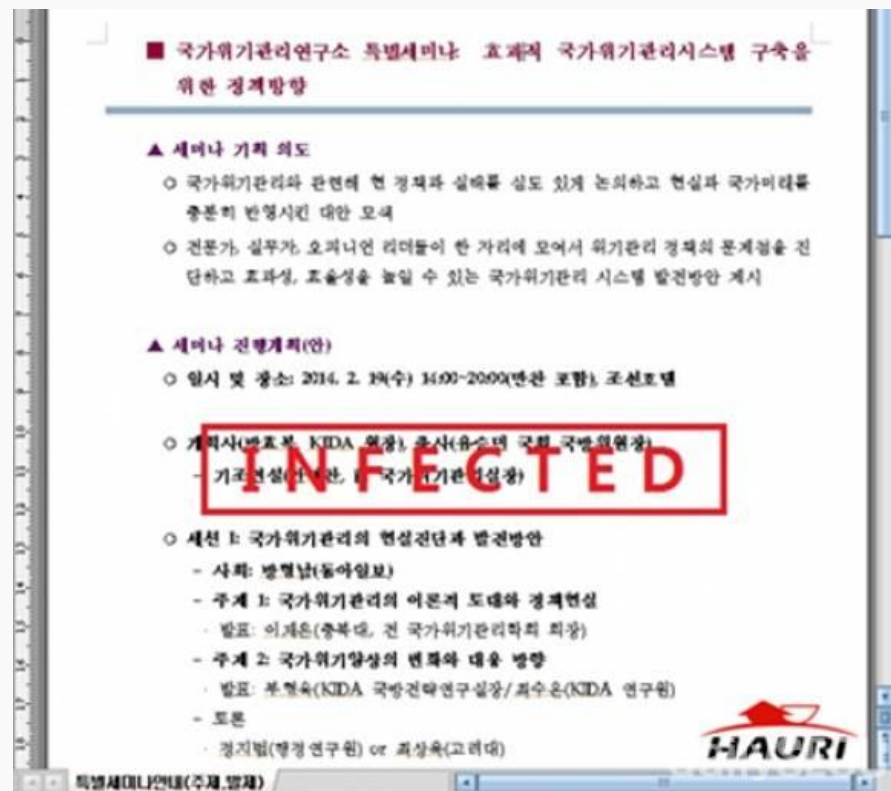
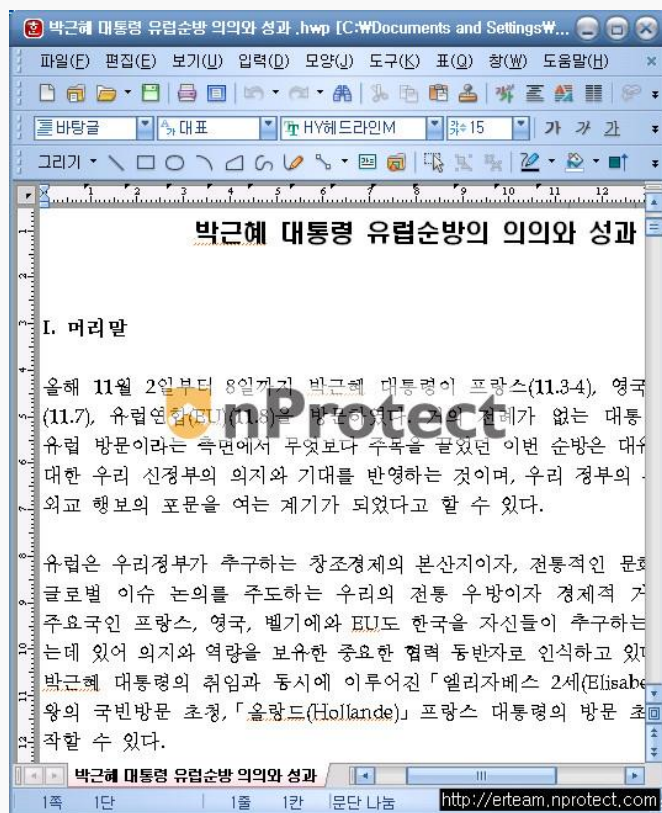
Pwned!

- 다양한 OS와 다양한 어플리케이션에서 다양한 형태로 나타날 수 있음.
- 프로그램의 취약점을 이용 할 경우 EXE 파일이 아닌 HTML, DOC, HWP, MP3 파일로 공격이 가능.



# What is software vulnerability?

- APT Attack





# What is software vulnerability?

- Malware

```
index.html - Notepad
File Edit Format View Help

<META NAME="Audience" CONTENT="General"> <META NAME="Date"
CONTENT="08/01/2002"> <META NAME="Doctype" CONTENT="Menupage : Gateway">
<META NAME="Subject"
CONTENT="Labor relations : Unions : Elections, Labor relations : Unions : Finances : Bonding, other emplo

<STYLE type="text/css">
OPTION.Red{background-color:#990000; color:white}
OPTION.white{background-color:#ffffff; color:black}
SPAN.nobreak{white-space: nowrap;}
</STYLE>

</HEAD>
<BODY>
<center>

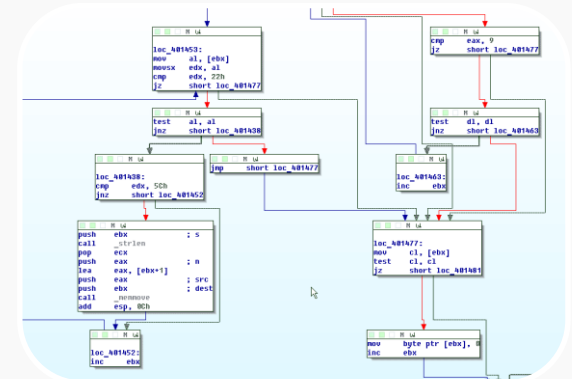
<link rel="stylesheet" href="/style.css" type="text/css" />
<link rel="stylesheet" href="http://www.dol.gov/agency.css" type="text/css" />
<link rel="stylesheet" href="http://www.dol.gov/print.css" type="text/css" media="print" />
<script language="javascript" src="/scripts/textsize.js" ></script>

<div id="Main">
  <!--Header Start -->
  <div id="Header">
    <a href="/"><div id="Banner"></div></a>
    <div id="RightColumn"><table><tr><td valign="bottom">
      <div id="Subscribe">
        <div id="Subscribelabel">Subscribe to <a href="http://www.dol.gov/dol/email.htm">E-mail u
        <div id="SubscribeForm">
          <form action="https://service.govdelivery.com/service/multi_subscribe.html" method
            <input type="hidden" name="code" value="USDOL" />
            <input type="hidden" name="origin" value="http://www.dol.gov" />
            <input type="text" name="login" value="Enter E-mail Address" onClick="this.value=''
            <input type="image" src="/images/SubscribeButton.gif" class="subscribeButton" ALT=
          </form>
        </div>
      </div>
    </td>
  </tr>
</table>
</div>
```





```
Iteration #1456
Iteration #1457
Iteration #1458
Iteration #1459
Iteration #1460
Iteration #1461
Iteration #1462
Iteration #1463
Iteration #1464
Iteration #1465
Iteration #1466
Iteration #1467
Iteration #1468
Iteration #1469
Iteration #1470
Iteration #1471
Iteration #1472
Iteration #1473
```

[illegible]

# Fuzzing

# Source Code Auditing

# Reverse Engineering

- Taint Analysis
- Symbolic Execution
- ETC

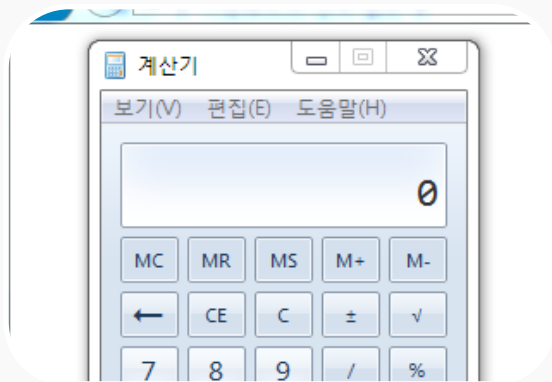


# Fuzzing

- 소프트웨어의 보안을 점검 하는 방법들 중 하나.
- 대상이 오류를 발생할 만한 값을 대입하는 방법.
- 간단한 방법으로 테스트가 가능하여 인기가 많은 편.
- 주로 Memory Corruption Bug를 찾기 위해 사용.
- 대부분의 취약점이 Fuzzing 을 통해 발견.

# Fuzzing

- Fuzzing For



Fun

[\$1000] [106484] High CVE-2011-3928: Use-after-free  
Arthur Gerkis.  
[\$3133.7] [107182] Critical CVE-2011-3928: Use-after-free  
Credit to Chamal de Silva. \*  
[108461] High CVE-2011-3928: Use-after-free  
team509 reported through ZDI (ZDI-CAN-2011-001)  
[\$1000] [108605] High CVE-2011-3927: Use-after-free  
[\$1000] [109556] High CVE-2011-3926: Use-after-free  
Arthur Gerkis.

Credit



Prize

# Fuzzing

- Dumb fuzzing
  - Fuzzing 대상에 대한 정보가 없이 무작위 대입
  - Dumb Fuzzing으로도 많은 취약점 발견.
- Smart Fuzzing
  - Dumb Fuzzing이 아닌 Fuzzing 방법
  - 현재 많은 연구가 이뤄지고 있음.



# Fuzzing

- Example (Buffer Overflow)
- Source code

```
...  
// some codes...  
        strcpy(buf_256, untrusted_input);  
// some codes...  
...
```

- Input
  - Case 1 : %s%x%s%x%n%n ← result : %s%x%s%x%n%n
  - Case 2 : AAAA... (128 bytes) ← result : AAAA... (128 bytes)
  - Case 3 : SNV(\$HNA... <- result : SNV(\$HNA...
  - Case 4 : AAAA... (512 bytes) ← Access Violation



# Fuzzing

- Example (Information Disclosure)
- Source code

```
...  
Char buf[256];  
Char secret[10];  
// some codes ...  
memcpy(buf, untrusted_input, strlen(untrusted_input));  
printf("%s", buf);  
...
```

- Input
  - Case 1 : %s%x%s%x%n%n ← result : %s%x%s%x%n%n
  - Case 2 : AAAA... (128 bytes) ← result : AAAA... (128 bytes)
  - Case 3 : AAAA... (256 bytes) ← result : AA...AAAinc0gnito

# Let's Fuzz

- Fuzzer?
  - 소프트웨어의 취약점을 발견하기 위해 Fuzzing을 수행하는 자동화, 반자동화 Tool
- 소프트웨어의 취약점을 테스트 하기 위한 Fuzzer 는 이미 많이 등장한 상태





# Let's Fuzz


- 수많은 Fuzzer들!

file fuzzer

Search

We've found 31 repository results

Sort: Best match ▾

 **Cr4sh/MsFontsFuzz** C++ ★ 16 🔒 8

OpenType font **file** format **fuzzer** for Windows

Last updated on 9 Jun 2013

- 공개된 Fuzzer는 참고 용도로 쓰는 것을 권장.
- 자신만의 Fuzzer를 만드는 것이 중요.







# Let's Fuzz

- Fuzzer 제작 시작!
- Language
  - Python
    - 간단한 문법
    - 다양한 라이브러리
      - Pydbg – Debugger Module, distorm3 – Disassemble Module
    - ...
- 기타
  - 본인이 가장 잘할 수 있는 언어
  - 비교적 간단한 “인터프리터” 언어 추천.





# Let's Fuzz

- 기본적인 Fuzzer 설계
  - Fuzzer
    - Fuzzing 대상을 실행 하는 모듈
    - Untrusted data 를 생성하는 모듈
    - 오류 발생시 정보 수집용 Debugger
  - Advanced Fuzzer
    - Crash Analyzer Script
    - Your idea.

# Let's Fuzz

- Fuzzing 대상에 대한 정보 수집

자료형	길이(바이트)	설명
WORD array[7]	14	언어별 글꼴 ID(FaceID) 참조 값(표 29 참조)
UINT8 array[7]	7	언어별 장평, 50% ~ 200%(표 29 참조)
INT8 array[7]	7	언어별 자간, -50% ~ 50%(표 29 참조)
UINT8 array[7]	7	언어별 상대 크기, 10% ~ 250%(표 29 참조)
INT8 array[7]	7	언어별 글자 위치, -100% ~ 100%(표 29 참조)
INT32	4	기준 크기, 0pt ~ 4096pt
UINT32	4	속성(표 30 참조)
INT8	1	그림자 간격, -100% ~ 100%
INT8	1	그림자 간격, -100% ~ 100%
COLORREF	4	글자 색
COLORREF	4	밑줄 색
COLORREF	4	음영 색
COLORREF	4	그림자 색
UINT16	2	글자 테두리/배경 ID(CharShapeBorderFill ID) 참조 값
COLORREF	4	취소선 색
전체 길이	72	



# Let's Fuzz

- Mutation Example
- Case 1
  - AAAABBBB => AAAABBBBCCCC...
- Case 2
  - AAAABBBB => ABAABABB...
- Case 3
  - AAAABBBB => AACCCCB B...
- Case 4
  - AAAABBBB => DFEGACKZ...

Original, Original , Abnormal



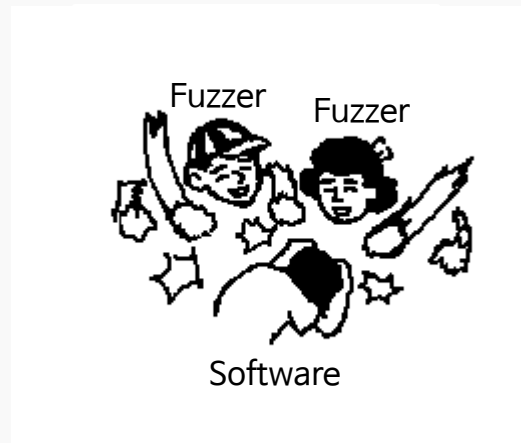


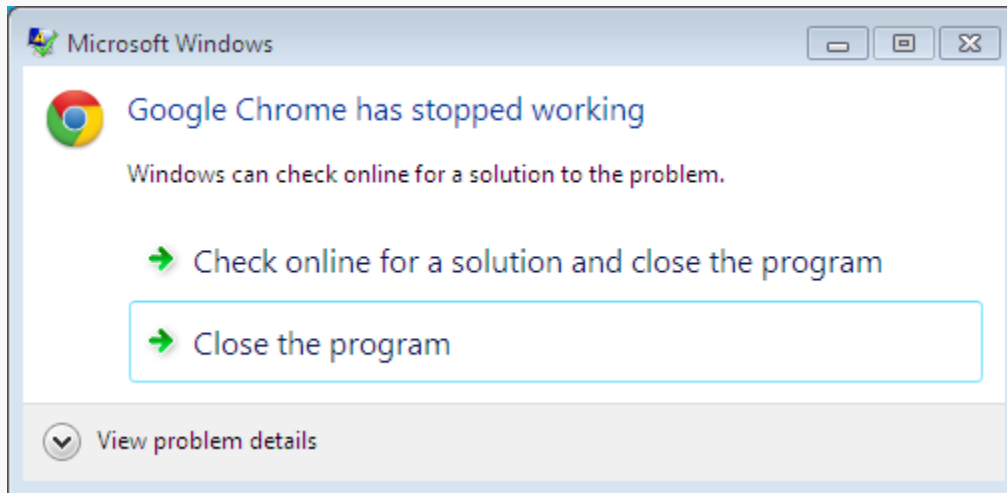
# Let's Fuzz

- Debugger
  - Fuzzing 과정 중 Crash 결과를 받기 위한 용도
  - Python – Pydbg
  - <http://blog.sweetchip.kr/317> (pydbg)
  - 사용이 매우 간편함
  - 예제가 많음

```
CONTEXT DUMP
EIP: 1e032272 test dword [eax+0x541.0x4000
EAX: 41414141 <1094795585> -> N/A
EBX: 00a930c0 < 11088064> -> 0H;kp p FH H 5s``>0000 0H P;x>8!B 0>1$>JX$h>HI<
<heap>
ECX: 00b0a0e0 < 11575520> -> AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA <heap>
EDX: 00001fff < 8191> -> N/A
EDI: 00b0a0e0 < 11575520> -> AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA <heap>
ESI: 00b0a0e0 < 11575520> -> AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA <heap>
EBP: 0021fe9c < 2227868> -> !T0j">0?kTx`pKlkTx!;!0t3>?t[xFjx0.ix0ghx0"Tx!`>
/<^ <stack>
ESP: 0021fe80 < 2227840> -> &0!T0j">0?kTx`pKlkTx!;!0t3>?t[xFjx0.ix0ghx0"Tx!
<stack>
+00: 000000fa < 2298> -> N/A
+04: 1e0326e3 < 503523043> -> N/A
+08: 00b0a0e0 < 11575520> -> AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA <heap>
+0c: 00000000 < 0> -> N/A
+10: 1e1e9290 < 505320080> -> enable() -> NoneEnable automatic garbage collect
ion.isenabled() -> statusReturns true if automatic garbage collection is enabled
.collect([generation]) -> nWith no arguments, run a full collection. The option
al argu <python26.dll.data>
```

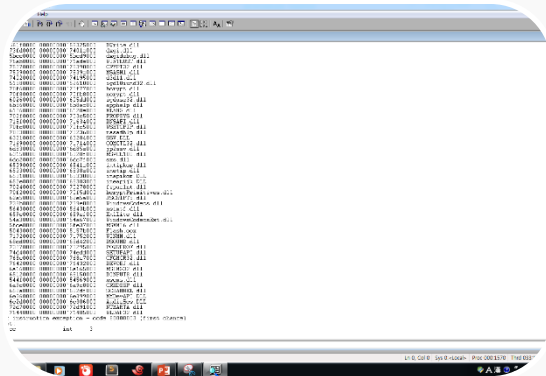
# Fuzz Fuzz Fuzz!



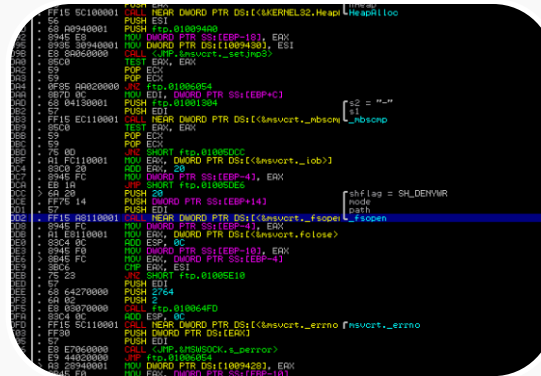




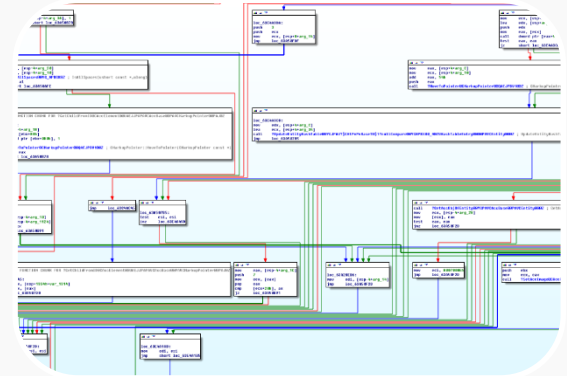
# Analysis



WINDBG



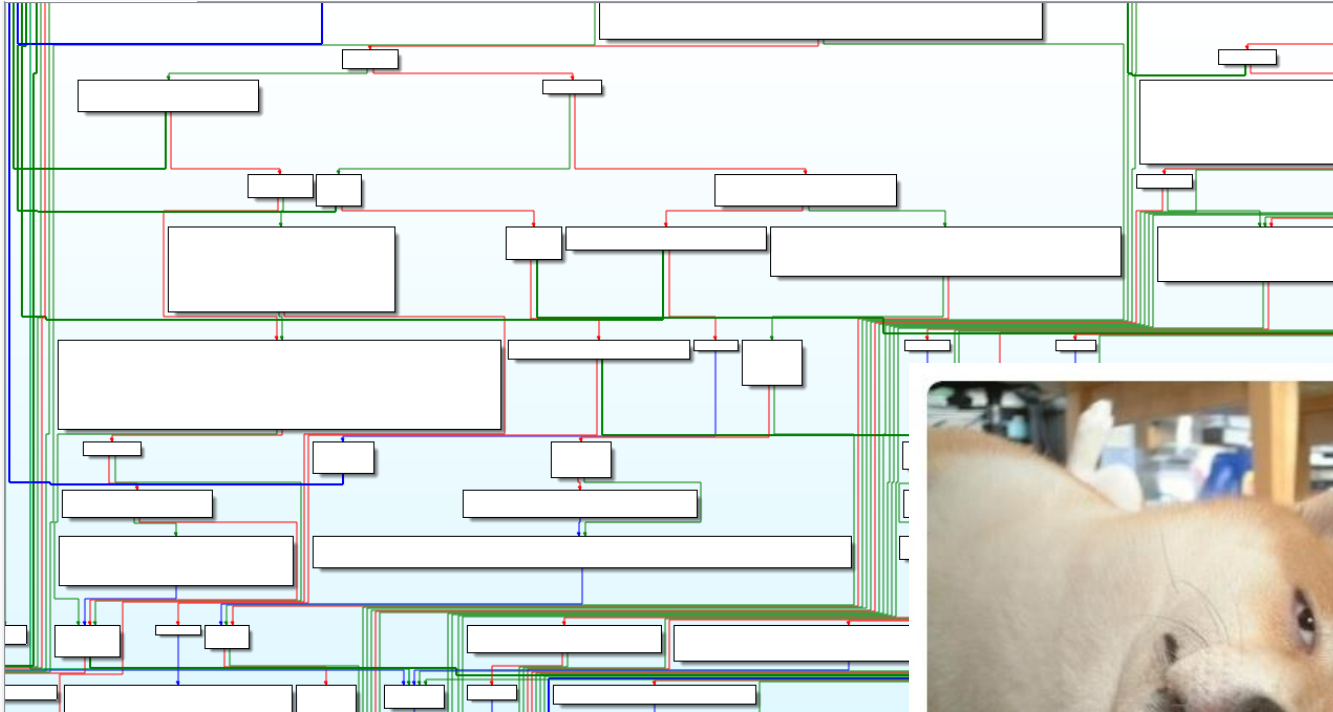
Immunity  
Debugger



IDA



# Analysis



# !exploitable

- Fuzzing으로 남은 생각보다 많은 Crash(bug).
  - 분류가 필요하다!
- Windbg 확장 프로그램
- Crash 형태를 보고 4가지의 상태로 분류한다.
  - Exploitable
  - Probably-Exploitable
  - Probably-not-Exploitable
  - Unknown

```
Exploitability Classification: PROBABLY_EXPLOITABLE
```

```
Recommended Bug Title: Probably Exploitable - User Mode Write AV near NULL starting at 7701418000000000
```

```
User mode write access violations that are near NULL are probably exploitable.
```



# Exploit

- Bypass Memory Protections
  - ASLR
  - DEP
  - SAFE SEH
  - Sandbox
- Unicode Payload
- ETC...

# Exploit

- Mona.py
  - Corelan Team이 제작한 Exploit Dev script
  - Immunity debugger, windbg 플러그인
  - Exploit 과정에 필요한 정보 수집 가능
    - 모듈에 설정된 메모리 보호기법
    - SEH Overwrite 에 사용될 수 있는 가젯
    - ROP 체인, ROP 가젯, Stack Pivot 가젯 등
  - <http://redmine.corelan.be/projects/mona>
  - <https://www.corelan.be/index.php/2011/07/14/mona-py-the-manual/>

# Exploit

- Metasploit
  - Exploit Framework, Pentesting Tool
  - Exploit, Shellcode, Shellcode Encoder 등 유용한 도구가 많음
  - <http://blog.sweetchip.kr/281> (shellcode 만들기)
- Exploit-DB
  - 프로그램의 Exploit code를 볼 수 있는 사이트
  - 1주일에 3번 이상 새로운 Exploit 등록
  - <http://www.exploit-db.com/>



# And then?

- Advanced Exploit
  - Post-Exploitation
    - Weaponizing
    - Stack Frame 복구
    - Register 복구
    - ETC
- Responsible disclosure
- Bug Bounty



# Bug Bounty

- 버그 바운티 제도 (Bug Bounty)

	accounts.google.com	Other highly sensitive services [1]	Normal Google applications	Non-integrated acquisitions and other lower priority sites [2]
Remote code execution	\$20,000	\$20,000	\$20,000	\$5,000
SQL injection or equivalent	\$10,000	\$10,000	\$10,000	\$5,000
Significant authentication bypass or information leak	\$10,000	\$5,000	\$1,337	\$500
Typical XSS	\$3,133.7	\$1,337	\$500	\$100
XSRF, XSSi, and other common web flaws	\$500 - \$3,133.7 (depending on impact)	\$500 - \$1,337 (depending on impact)	\$500	\$100

- 해커가 발견한 취약점의 가치를 매기고 구입하거나 포상 하는 제도
- 소프트웨어의 보안 강화를 위해 시행 중인 기업도 있음.

# Bug Bounty

- Bug Bounty
- 현재 국외에서 버그 바운티 제도가 활발히 진행 중.







# Bug Bounty

- Bug Bounty
- 국내에도 버그 바운티 제도가 존재
  - KISA 취약점 포상제
- 기업 차원의 버그 바운티 제도는 국외에 비해 미약.

**KISA** 인터넷침해대응센터

# Bug Bounty

- HP's Zeroday Initiative
- 주로 Major급 프로그램과 Scada의 취약점을 접수 받음.
- 취약점 구매 방식.
- Tipping point 제품에 적용 후 벤더에 취약점 정보 전달

<b>ZDI-14-261</b>	CVE: CVE-2014-1765	Published: 2014-07-23
Microsoft Internet Explorer CAttrValue Use-After-Free Remote Code Execution Vulnerability		
<b>ZDI-14-260</b>	CVE: CVE-2014-1799	Published: 2014-07-23
Microsoft Internet Explorer CMarkupPointer Use-After-Free Remote Code Execution Vulnerability		
<b>ZDI-14-259</b>	CVE: CVE-2014-2764	
Microsoft Internet Explorer CTreeNode Double Free Remote Code Execution Vuln		



# Bug Bounty

- KISA 취약점 포상제
- 국내 프로그램의 보안 취약점을 상시로 접수
- 3, 6, 9, 12월 매 분기마다 취약점을 점검 하고 포상

알마인드 임의코드실행 취약점 보안 업데이트 권고

애플 모바일 운영체제(iOS) 보안 업데이트 권고

Cisco Wireless LAN Controller 다중 취약점 보안 업데이트 권고

알씨 임의코드실행 취약점 보안 업데이트 권고

다음 팟플레이어 임의코드 실행 취약점 보안 업데이트 권고

gnutls 라이브러리 취약점 보안 업데이트 권고

## 기타 문의사항

- 한국인터넷진흥원 인터넷침해대응센터: 국번없이 118
- 본 취약점은 KrCERT 홈페이지를 통해                     님께서 제공해주셨습니다.

**KISA**  
한국인터넷진흥원

# Bug Bounty

- Pwn2Own
- 매년 3월에 Cansecwest 컨퍼런스에서 열리는 ‘어플리케이션’ 해킹 대회
- 대회에서 지정된 프로그램의 취약점을 공격하면 성공

The **2014** targets are:

Browsers:

- Google Chrome on Windows 8.1 x64: **\$100,000**
- Microsoft Internet Explorer 11 on Windows 8.1 x64: **\$100,000**
- Mozilla Firefox on Windows 8.1 x64: **\$50,000**
- Apple Safari on OS X Mavericks: **\$65,000**





# Tips For Beginner

- Fuzzing For Fun, Fun is First.
- Make Your **OWN** Fuzzer
- Code Coverage
  - 코드가 실행된 정도 (wikipedia)



# Tips For Beginner

- Target
  - Easy : 이전에 취약점이 공개된 프로그램
  - Normal : 워드 프로세서, 미디어 플레이어 (국내)
  - Hard : 워드 프로세서 (국외)
- Windows XP
  - XP -> 7 -> 8 식으로 Exploit 개발
  - Memory Protection중 ASLR이 없음
  - Vista 이후 운영체제에 비해 Exploit이 비교적 쉬움



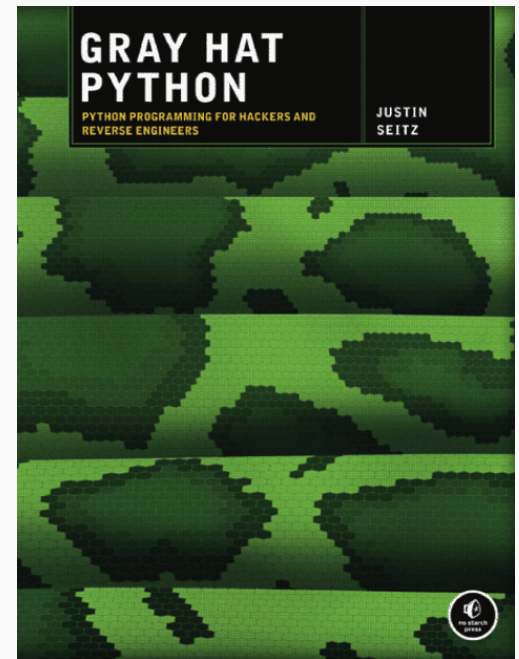
# Tips For Beginner

- Exploit writing tutorial
  - Corelan's tutorial
    - Corelan Team에서 제작한 Exploit writing tutorial
    - 국내 정보보호교육센터의 번역본
  - 구글링
  - ETC
    - <http://blog.sweetchip.kr/290> (Media Player Exploit – tutorial)



# Tips For Beginner

- 추천할 만한 책
  - Grayhat Python (원서)
  - 파이썬 해킹 프로그래밍 (번역)
  - File Fuzzer 등 다양한 예제





# Q & A

Any Questions?  
Please feel free to ask me



# Reference

- Beist - Everyone has his or her own fuzzer (Codeengn)
- Passket - 급전이 필요합니다. (Codeengn)
- Pwn3r - exploit case by case (Codegate)

THANK YOU

