Malicious File for Exploiting Forensic Software

Takahiro Haruyama / Hiroshi Suzuki
Internet Initiative Japan Inc.

Who am I?

- Forensic Investigator & Malware Analyst @ Internet Initiative Japan
- Presentations
  - SANS DFIR Summit, Blackhat EU, RSA Conference Japan, etc..
- Hands-ons
  - CEIC, FIRST TC Kyoto/Lisbon, etc..
- Tools
  - EnCase EnScript, IDAPython script, etc..
  - http://cci.cocolog-nifty.com/blog/
- EnCase Certified Examiner since 2009
- twitter: @cci_forensics

Overview

- Background
- Fuzzing Oracle Outside In
- Anti-forensics by exploiting bugs
- Countermeasures
- Wrap-up

Background

File Viewer in Forensic Software

- Forensic software needs a function viewing file content
  - Most commercial tools adopt the same library
    - e.g., EnCase, FTK, X-Ways, etc..
- Oracle Outside In Technology [1]
  - analyze/extract/convert over 500 different file types data
  - also used by enterprise software
    - e.g., Microsoft Exchange, Cisco Security Agent, IBM OmniFind Enterprise Edition, McAfee GroupShield, Symantec Enterprise Vault

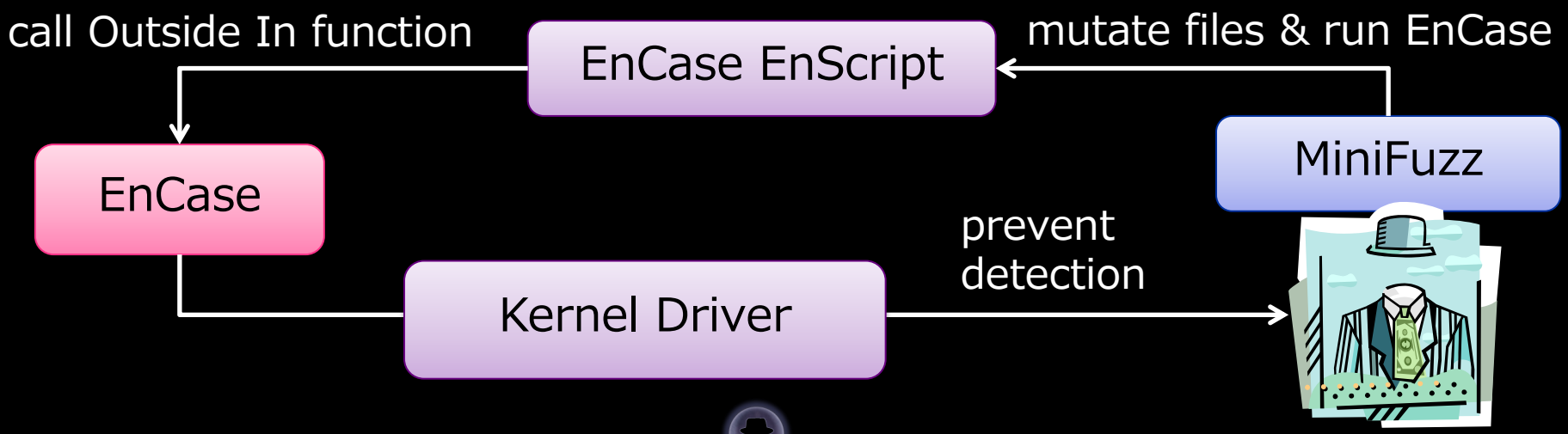
Motivation

- Several bugs of Oracle Outside In were reported last year [2]
- If still exploitable, lots of forensic investigators are exposed to risks when viewing/processing crafted malicious files
  - process hang-up
  - malware infection with privilege
  - other anti-forensic techniques
    - e.g., data alternation or hiding, evidence deletion
- Researched about the exploitability

Fuzzing Oracle Outside In

Fuzzer Implementation

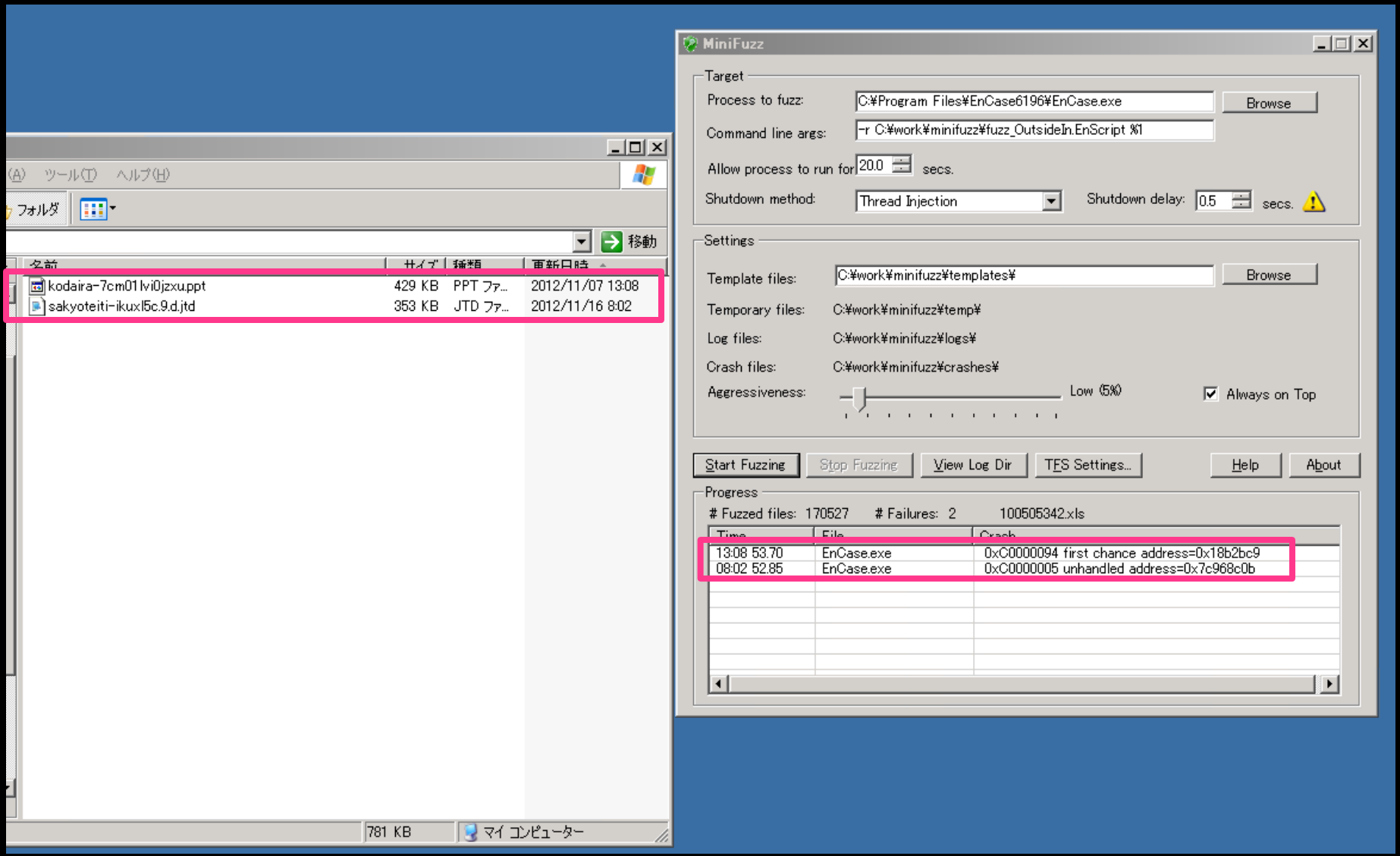
- The fuzzer using EnCase
  - MiniFuzz [3]
  - EnCase EnScript
  - kernel driver



EnScript for Fuzzer

- EnScript methods calling Outside In
  - DocumentClass
    - GetDocView
      - Returns a list containing one image for each page in the document
    - WriteTranscript
      - Returns a text extracted from the document
- One pitfall
  - EnScript can’t receive command-line arguments
    - Use 3rd party tool [4]

Check the Result



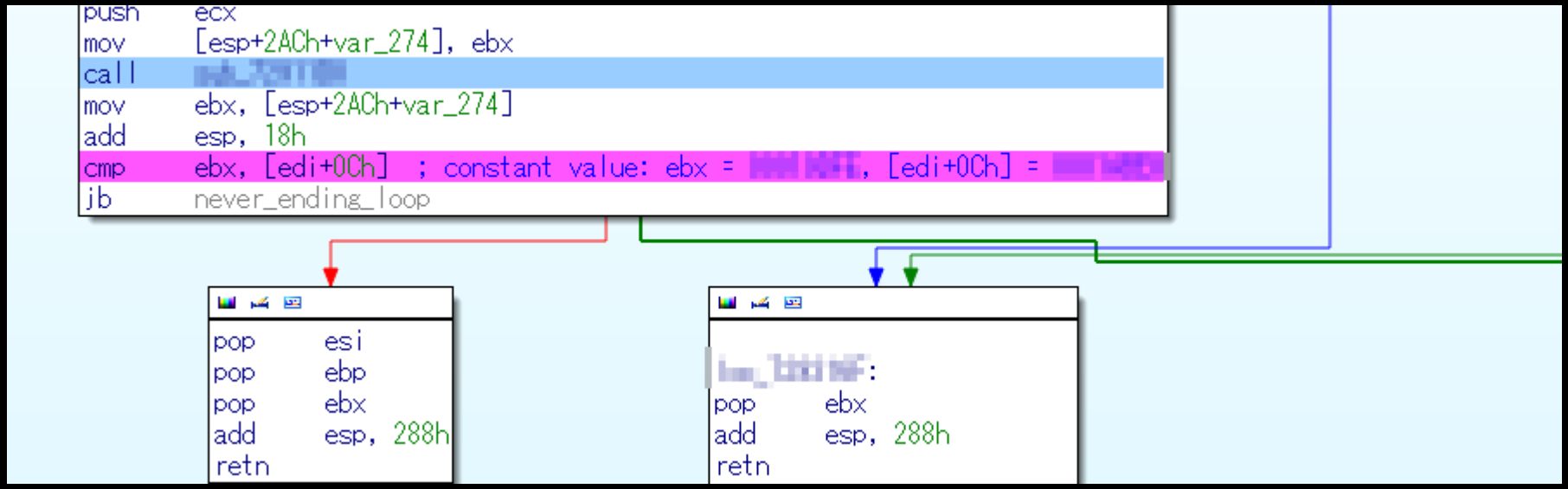
Anti-forensics by exploiting bugs

Anti-forensics by exploiting bugs

- Two examples
  - Process hang-up using infinite loop DoS vulnerability
  - Arbitrary code execution using heap overflow vulnerability
- Tested versions
  - 8.3.5 (too old!) on EnCase 6
    - To tell the truth, most EnCase users prefer 6 to 7 ☺
  - 8.3.7 on X-Ways Forensics
    - X-Ways recommends 8.3.7 instead of 8.4.1 for web history examination
  - 8.4.0 on EnCase 7

Process Hang-up using Infinite Loop DoS Vulnerability

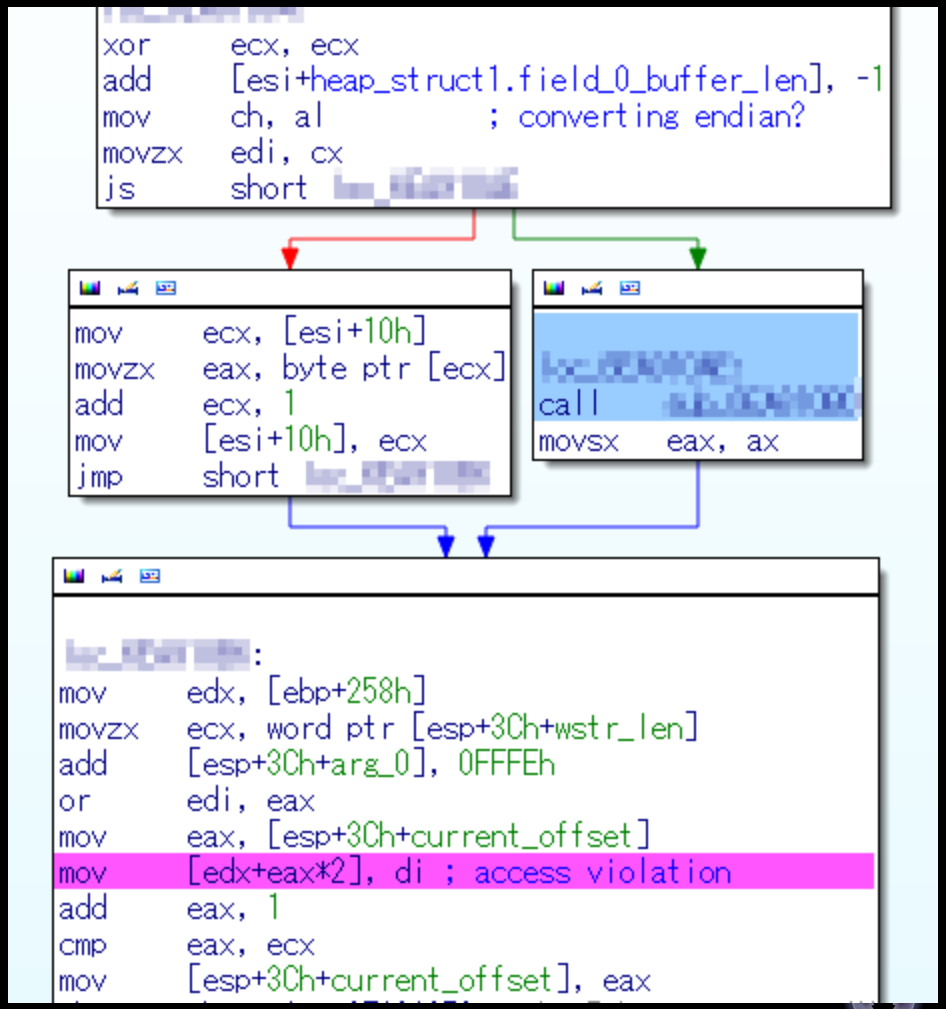
- Included in the function parsing Hangul Word Processor format document
  - A malicious file can cause infinite loop in vshwp2.dll
- JVN#68663052 [5] (CVE-2013-3776 [6])
  - Affected version: 8.4.1 and earlier
  - Resolved on July 16th, 2013 [7]



Demo

- 8.4.0 on EnCase 7

Arbitrary Code Execution using Heap Overflow Vulnerability



- Included in the function parsing Ichitaro format document
  - A malicious file can overwrite heap chunks
- JVN#07497769 [8] (CVE-2013-3781 [9])
  - Affected version: 8.3.7 and earlier
  - Resolved on July 16th, 2013 [7]

Overwriting Function Pointers

- Vista or later Windows OS adopt various mitigation techniques to prevent an exploitation of heap overflow [10]
  - Look-aside lists have been replaced by the Low Fragmentation Heap (LFH)
  - heap entry metadata randomization
  - randomized heap base address
  - etc...
- One promising method is overwriting function pointers in heap chunks
  - But the offset values to them are not constant

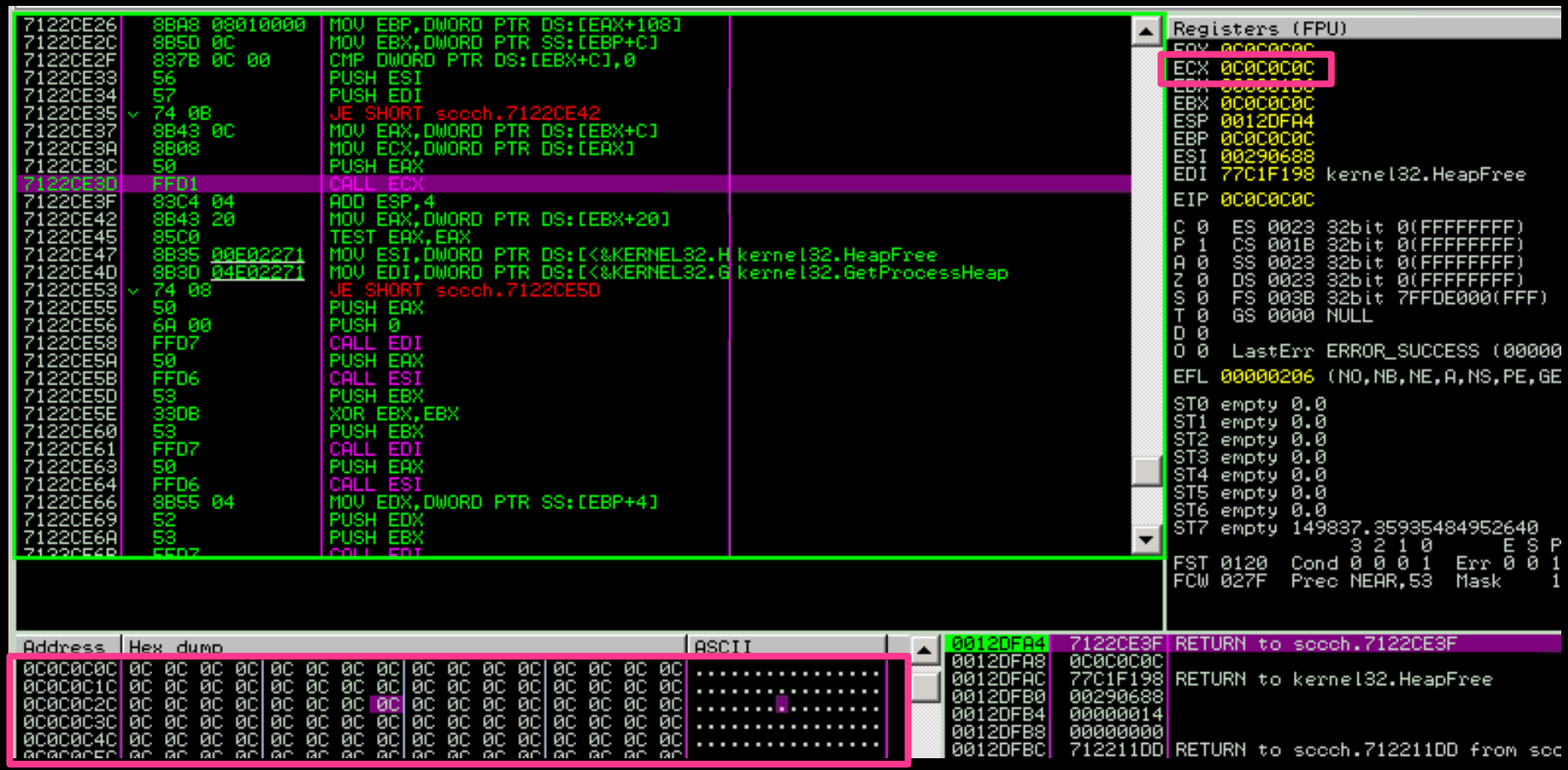
Heap Spraying

- Heap spraying is a payload delivery technique
  - It allows us to put our shellcodes at a predictable address (e.g., 0x0c0c0c0c)
- We need to fill chunks of memory in the heap before gaining control over EIP
- How?
  - Javascript or vbscript in web browsers
  - Javascript or ActionScript in Adobe Reader
  - ? in forensic software

Heap Spraying with Bitmap Images

- Bitmap heap spraying [11] is an effective technique because forensic investigators often examine image files
- Most forensic software supports displaying several images at the same time
  - EnCase
    - "Bookmark Page as Image" in Doc view
    - Make the bookmarked images "set-included"
  - X-Ways Forensics
    - Double-click image files

Heap Spraying with Bitmap Images: Example on EnCase 6



Demo

- 8.3.5 on EnCase 6
- 8.3.7 on X-Ways Forensics

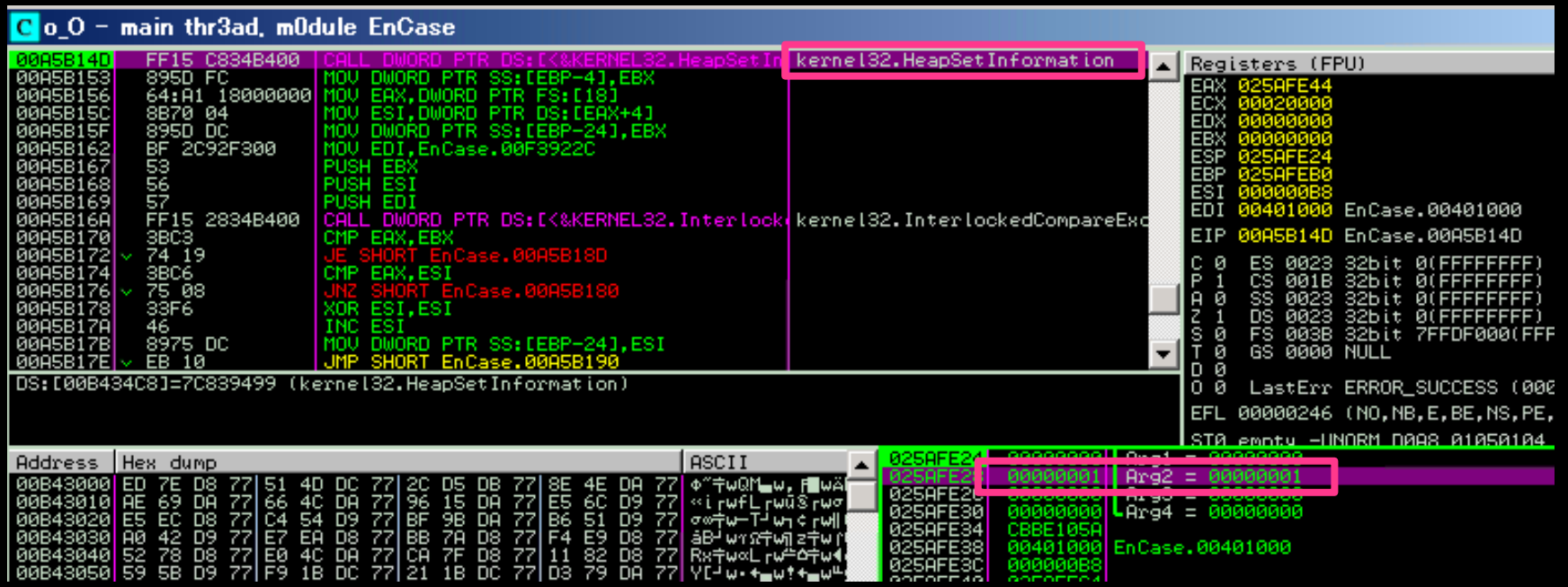
Success Probability of the Heap Overflow Exploitation

- Not necessarily succeed
  - Function pointers called in a short time should be included in overwritten area
    - e.g., sccch/sccut/sccvw, ole32, etc..
  - current success probability
    - EnCase 6
      - 40-50%
    - X-Ways
      - 10-20%
  - To improve the probability, we need to manipulate heap chunk layout before causing overflow ☹

Countermeasures

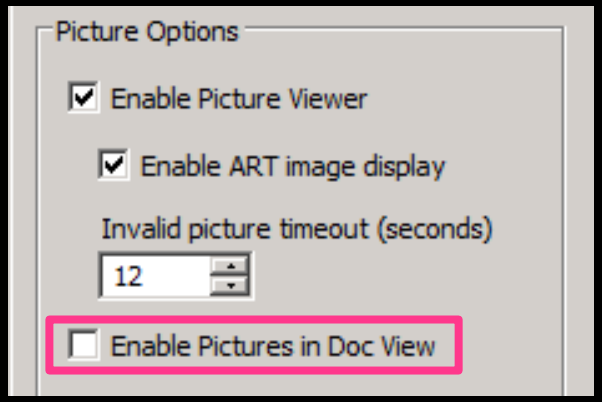
Prevention of Arbitrary Code Execution using Heap Overflow

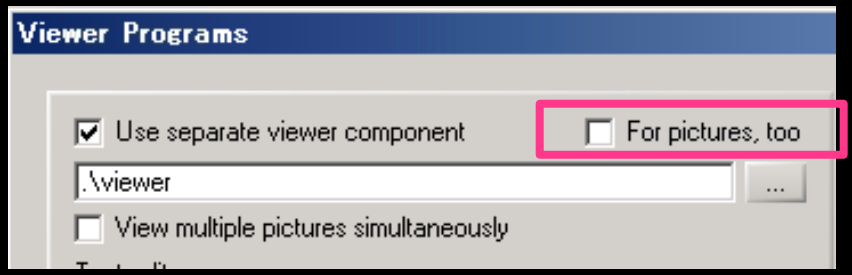
- Developers should
  - check buffer boundary to prevent overflow
  - enable “HeapEnableTerminateOnCorruption“
    - Use HeapSetInformation [12] API
    - If a heap corruption detected, the process terminates immediately
      - 0xc0000374 (STATUS_HEAP_CORRUPTION)



Heap spray Prevention

- Disable the operation for bitmap heap spray
  - EnCase 6
    - [Tools] -> [Options] -> [Global]
    - uncheck “Enable Pictures in Doc View”
  - X-Ways Forensics
    - [Options] -> [Viewer Programs]
    - uncheck “For pictures, too”





Common Countermeasures

- Use the latest version
  - 8.4.0 or later seems to fix most bugs discovered by my fuzzer
    - Do not use EnCase 6 / X-Ways with 8.3.7
- Do not install file viewer
  - We can install EnCase/X-Ways without Outside In
- Configure for exploit mitigation [13]
  - e.g., EMET, AppLocker, etc...

Wrap-up

Wrap-up

- Know the risk when examining unknown files acquired in forensic investigation
  - The file viewer component is fragile
    - The two bugs may be just a little bit of the problem
  - Investigators should pay attention to the security settings of their workstations
- Forensic software vendors tend to use an old version of the component
  - They should update their products as soon as the latest version is released

Questions?
(twitter: @cci_forensics)

Please complete the Speaker Feedback Surveys

References

[1] Oracle Outside In Technology <http://www.oracle.com/us/technologies/embedded/025613.htm>

[2] Oracle Outside In contains multiple exploitable vulnerabilities <https://www.kb.cert.org/vuls/id/118913>

[3] SDL MiniFuzz File Fuzzer <http://www.microsoft.com/en-us/download/details.aspx?id=21769>

[4] Command Line DLL.zip <http://www.swiftforensics.com/p/downloads.html>

[5] JVN#68663052 Oracle Outside In vulnerable to denial-of-service (DoS) <http://jvn.jp/en/jp/JVN68663052/index.html>

[6] CVE-2013-3776 <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-3776>

[7] Oracle Critical Patch Update Advisory - July 2013 <http://www.oracle.com/technetwork/topics/security/cpujuly2013-1899826.html>

[8] JVN#07497769 Oracle Outside In vulnerable to buffer overflow <http://jvn.jp/en/jp/JVN07497769/index.html>

[9] CVE-2013-3781 <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-3781>

[10] Preventing the exploitation of user mode heap corruption vulnerabilities <http://blogs.technet.com/b/srd/archive/2009/08/04/preventing-the-exploitation-of-user-mode-heap-corruption-vulnerabilities.aspx>

[11] Exploit writing tutorial part 11 : Heap spraying Demystified <https://www.corelan.be/index.php/2011/12/31/exploit-writing-tutorial-part-11-heap-spraying-demystified/>

[12] HeapSetInformation function <http://msdn.microsoft.com/en-us/library/windows/desktop/aa366705(v=vs.85).aspx>

[13] Forensics Software and Oracle Outside In <https://www.cert.org/blogs/certcc/2013/07/forensics_software_and_oracle.html>