中国科学院软件研究所学术年会’2018
暨计算机科学国家重点实验室开放周

学术论文

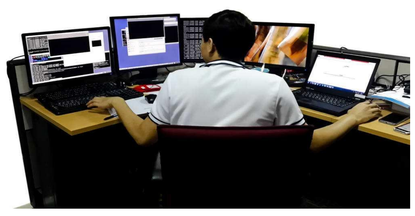
面向二进制堆溢出漏洞的可利用性自动化评估

Automatically Assessing Crashes from Heap Overflow

和亮, 蔡彦, 胡宏, 苏璞睿 (purui@iscas.ac.cn), 梁振凯, 杨轶,黄桦烽, 闫佳, 贾相堃, 冯登国

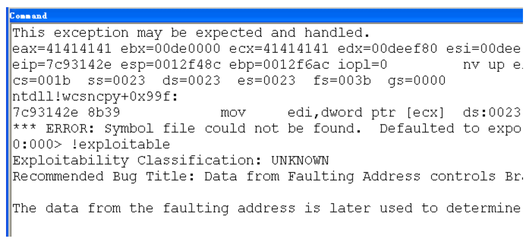
The 32nd IEEE/ACM Automated Software Engineering (ASE 2017)

现有方案



(1) 手工调试

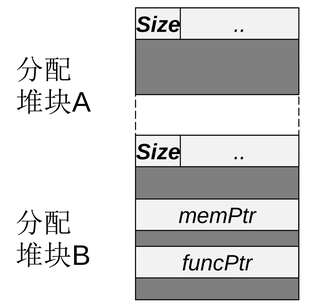
分析周期长，难以应对海量崩溃样本



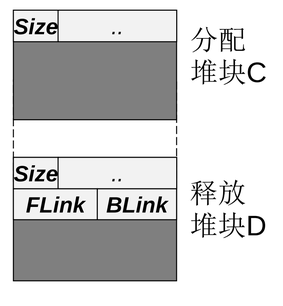
(2) 调试器插件(!exploitable)

结果简单，难以准确刻画漏洞利用难度

背景知识



利用漏洞从A中溢出破坏B中的函数指针或者数据指针实施特定攻击。



利用漏洞从C中溢出破坏D中的堆块管理数据实施特定攻击。

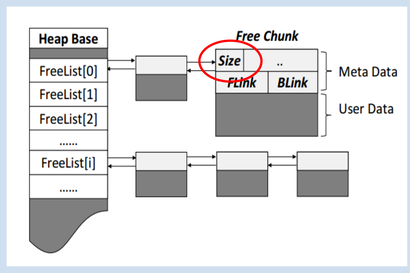
评估指标

攻击性指标：描述堆溢出漏洞的潜在危害

- 利用点(Exploit Point)：用来描述构建攻击代码的关键指令(序列）
- 溢出字节(Overflow Bytes)：越多的溢出字节代表着越大的潜在破坏力
- 污点字节(Taint Bytes)：指明漏洞发生时由用户可控的内存数据区域
- 污点关系(Overflow Relation)：攻击载荷是否由用户控制的数据填充

利用点（Exploit Point）	利用攻击（Exploit Attack）
call **taint**	控制流劫持类攻击，执行远程代码
call [**taint**]	控制流劫持类攻击，执行远程代码
mov [**taint**], **taint**	任意内存访问攻击，窃取用户隐私
mov [**taint**], [**taint**]	任意内存访问攻击，窃取用户隐私

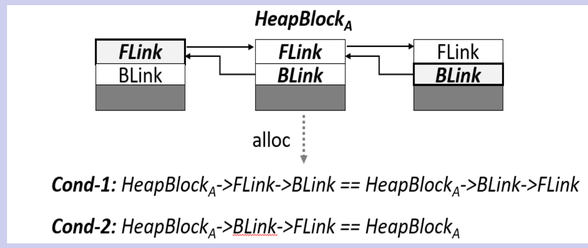
直接利用点
taint 指用户数据



间接利用点
通过溢出修改*Size*实现二次溢出

可实施性指标：描述构建攻击代码的困难程度

- 指针解引用次数(Pointer Dereference Count)：被破坏指针的解引用次数
  - 索引指针(Index Pointer)：数组下标和地址偏移，可轻易伪造
  - 基地址指针(Base Pointer)：内存基地址指针，受随机化影响，难以伪造
  - 多级指针(Multilevel Pointer)：常用于多级链表，更难伪造
  - 受保护指针(Protected Pointer)：系统或者程序重点保护，几乎无法伪造
- 取值约束(Value Constraint)：对于外部输入数据的合规性检验

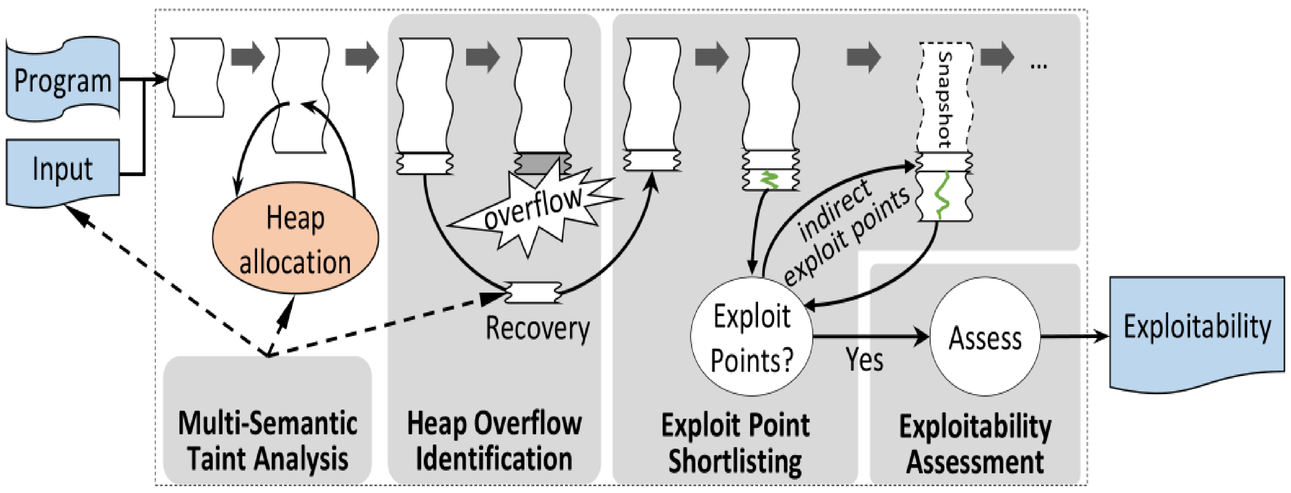


堆块中受保护指针的防护策略

```
1. for i=0 to i=N
2.  if (buf[i]>‘\0x3f’)
3.       break;
4.  else
5.       copy_heap_memory;
6. ……
```

常见取值约束

系统设计



HCSifter (Heap Crash Sifter) 系统框架

特色功能

1. 多语义污点传播
   - 1.1. 通过堆指针的污点跟踪寻找溢出
   - 1.2. 通过溢出内存的污点跟踪寻找利用点
   - 1.3. 通过输入数据的污点跟踪填充攻击代码
2. 动态内存实时恢复
   - 2.1. 通过1.1完成堆溢出的实时预测
   - 2.2. 在溢出发生前对将要溢出的内存数据进行保存
   - 2.3. 在溢出发生后立即对溢出的内存地址进行恢复

实验结果

表1. 实验结果概览

Programs	Heap Overflow Information			Basic Metrics			Exploitability Assessment	
	Instruction	Function	OB	TB	TR	VC	!exploitable	HCSIFTER
1ClickUnzip	mov [edx],al	lstrcpyA	2433	6573	(1-1)	B!0x0	✓	[IP]+
Acousitca Converter	mov [eax+edx], cl	lstrcpyA	1092	10613	(1-1)	B!0x0	UNKNOWN	[BP+PP]
CoreFTP Client	rep movsd	recv	8910	17653	(1-1)	B!0xA	UNKNOWN	[IP]
FoxitReader	mov [edx],al	lstrcpyA	4241	443	(1-1)*	B!0x0	✓	✓
HTTPDX	rep movsd	memcpy	1049	1059	(1-1)	B!0xA	UNKNOWN	[BP+PP]
Python	rep movsd	memcpy	64745	723419	(1-1)*	B!0x0	✓	✓
Vallen Zipper	mov [edx],al	lstrcpyA	4021	8443	(1-1)	B!0x0	UNKNOWN	✓
WMPlayer	rep movsd	-	2524	2547	(1-1)	-	✓	✓
ZipItFast	-	ReadFile	4021	20899	(1-1)	-	✓	✓

表2. 基于间接利用点的多轮迭代

Program	1st Round						2nd Round					
	Exp		D-Score		I-Score		Exp		D-Score		I-Score	
	DExp	IExp	Offset	PDC	Offset	PDC	DExp	IExp	Offset	PDC	Offset	PDC
1ClickUnzip	24	23	0x19c2	(1, 17, 4, 0)	0x19ba	(1, 0, 0, 0)	14	12	0x19be	(1, 0, 0, 0)	0x19b2	(1, 0, 0, 0)
Acousitca Converter	14	4	0x8	(0, 1, 0, 1)	0x0	(0, 0, 0, 0)	0	0	-	-	-	-
CoreFTP Client	1	13	0x1dd8	(2, 1, 1, 1)	0x1dd0	(2, 0, 0, 0)	2	0	0x1ddc	(2, 0, 0, 0)	-	-
FoxitReader	1	1	0x11c	(0, 1, 0, 0)	0x114	(0, 0, 0, 0)	1	0	0x120	(0, 0, 0, 0)	-	-
HTTPDX	6	1	0x408	(0,134, 0, 1)	0x400	(0, 0, 0, 0)	0	0	-	-	-	-
Python	157	1	0x282d	(0, 0, 0, 0)	0x2821	(0, 0, 0, 0)	0	0	-	-	-	-
Vallen Zipper	2	4	0x8	(0, 0, 0, 0)	0x50	(0, 0, 0, 0)	0	0	-	-	-	-
WMPlayer	950	0	0xe0	(0, 0, 0, 0)	-	-	-	-	-	-	-	-
ZipItFast	4215	0	0x100	(0, 0, 0, 0)	-	-	-	-	-	-	-	-

■实验评估结果：

- **基准数据集：**从exploit-db上搜寻到的9个Windows下可重现的（含PoC）堆溢出样本
- **评估结果：** 判定5个可直接利用，2个[IP]级漏洞，2个[BP+PP]级漏洞
- **!exploitable：**5个可直接利用，4个未知(UNKOWN)