Fosdem, Beer, Curry and Pwnage
and why Metasploit is written in Ruby... </troll>

Simon Lewis and Rob Shields

FOSDEM, 2007
First Things First

- Beer is a **MAJOR** feature of FOSDEM
- *The conference is held at ULB Campus every year*
- [http://fosdem.org](http://fosdem.org) (slides, videos and photos)
The beer event

Figure: David and Rich at the beer event
The beer event #2

Figure: Simon tries to pwn an American girl in a pub
M$ - We are in your customer base spoofing your runtime

- Sun - Annouced they are Patrons of the FSF, Open Sourcing of Java
- Novell - Talked about Mono and tools for porting Windows applications
Other Interesting talks

- Andrew Morton - *The state of the Linux Kernel, user types*
- *Debian Package Management* - How to be more clever
- X.org - *Why Beryl/Compiz is very hard*
One Laptop Per Child

- Jim Gettys gets mobbed at the end of his talk
Andrew Morton - The state of the Linux Kernel

- Server - infiniband, network protocols, SATA, SCSI, virtualisation, containerisation
- Desktop - hot pluggability (devices, CPU, nodes, memory), power man., DRI drivers, input, sound, 1394
- Embedded - DVB/VFL, dynamic ticks (needed by OLPC), footprint reduction, NoMMU device support
- General - instrumentation (per process I/O and memory stats), kevent, utrace (rewrite of ptrace), async syscalls, fault injection framework, kdbg possible
Terminology

- **Exploit**: A weakness in a program that allows an attacker to gain control of the process (EIP)
- **Payload**: This is what we want to be executed
We like to pwn it pwn it... we like to... pwn it

- A framework to separate exploits from payloads
- Benefits from the egos of FLOSS Hackers
- After a certain amount of time exploits are released.
- Point, Click and pwn
Workflow

Input Data -> Program Crash -> EIP

Debug and develop shell code

Working Exploit
Getting dirty with memory

<table>
<thead>
<tr>
<th>env variables</th>
<th>0xffffffff - High Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmd line args</td>
<td></td>
</tr>
<tr>
<td><strong>envp</strong></td>
<td></td>
</tr>
<tr>
<td><strong>argv</strong></td>
<td></td>
</tr>
<tr>
<td>argc</td>
<td></td>
</tr>
<tr>
<td>main vars</td>
<td></td>
</tr>
<tr>
<td>stack</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Strings are written to the address space this way</td>
</tr>
<tr>
<td></td>
<td><strong>Heap</strong></td>
</tr>
<tr>
<td></td>
<td>.bss</td>
</tr>
<tr>
<td></td>
<td>.data</td>
</tr>
<tr>
<td></td>
<td>.text</td>
</tr>
<tr>
<td></td>
<td>0x00000000 - Lower Addresses</td>
</tr>
</tbody>
</table>

A simple buffer overflow will allow an attacker to write to a higher memory address.

A function call works by pushing return address onto the stack. By overwriting the return address of a function you can gain control of EIP (the instruction pointer).
Demonstration
Summary

• Going to conferences is fun and exciting # beer++
• Metasploit is an exploitation framework now written in Ruby # OO++
• Clarity over speed where appropriate
• Patch, patch, PATCH!
Resources List

Diagrams were shamelessly borrowed and adapted from the following resources:

- [http://www.metasploit.com](http://www.metasploit.com) The main website for Metasploit