Introduction

Vasileios (Vasilis) Kemerlis

January 30, 2017

Department of Computer Science
Brown University
Course Overview (1/2)

► What is this course about?

State-of-the-art in software exploitation and defense

CSCI 1951H+

Memory unsafe code (written in C/C++, asm, ...)

Software Security
1. Prevalent software defects
   • Stack/Heap smashing
   • Format string bugs
   • Pointer errors
   • ...
2. Modern defenses
   • W^X, ASLR
   • Stack/Heap canaries
   • RELRO, BIND_NOW
   • BPF_SEC
   • Fortify_SRC
   • CFI, CPI, ...

Software Exploitation
1. Code injection
2. Code reuse
   • Ret2libc
   • Return-oriented prog. (ROP)
   • Just-In-Time ROP (JIT-ROP)
   • Blind ROP (BROP)
   • Signal-oriented prog. (SROP)
   • ...
3. Data-only attacks
What is this course about?

- State-of-the-art in software exploitation and defense → CSCI 1951H++
Course Overview (1/2)

What is this course about?

- ✓ State-of-the-art in software exploitation and defense ➔ CSCI 1951H++
- ✗ Memory unsafe code (written in C/C++, asm, ...)

Software Security

1. Prevalent software defects
   - Stack/Heap smashing
   - Format string bugs
   - Pointer errors
   - ...

2. Modern defenses
   - W^X, ASLR
   - Stack/Heap canaries
   - RELRO, BIND_NOW
   - BPF_SECCOMP, FORTIFY_SRC
   - CFI, CPI, ...

Software Exploitation

1. Code injection
2. Code reuse
   - Return-to-libc (ret2libc)
   - Return-oriented prog. (ROP)
   - Just-In-Time ROP (JIT-ROP)
   - Blind ROP (BROP)
   - Signal-oriented prog. (SROP)
   - ...

3. Data-only attacks

vpk@cs.brown.edu (Brown University)
Why take this course?

- Understand the boundaries of protection mechanisms and argue about their effectiveness
- Familiarize with experimental mitigation techniques
- Learn how and why (certain) defenses can be bypassed

Exploit "weaponization"

Why are these useful?

- To design effective (and efficient) software protection mechanisms you need to:
  (a) understand what sorts of attacks are possible
  (b) how exactly these attacks work
  (c) why previous attempts failed
Why take this course?

- **Offense**
  - Learn *how* and *why* (certain) defenses can be **bypassed**
    - Exploit “weaponization”
### Why take this course?

<table>
<thead>
<tr>
<th>☑️ Defense</th>
<th>☑️ Offense</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Understand the <strong>boundaries</strong> of protection mechanisms and argue about their <strong>effectiveness</strong></td>
<td>✓ Learn how and why (certain) defenses can be <strong>bypassed</strong></td>
</tr>
<tr>
<td>✓ Familiarize with experimental mitigation techniques</td>
<td>• Exploit “weaponization”</td>
</tr>
</tbody>
</table>
Course Overview (2/2)

▶ Why take this course?

😊 **Defense**

- Understand the **boundaries** of protection mechanisms and argue about their **effectiveness**
- **Familiarize** with experimental mitigation techniques

😊 **Offense**

- Learn **how** and **why** (certain) defenses can be **bypassed**
  - Exploit “weaponization”

▶ Why are these useful?

- To design effective (and efficient) software protection mechanisms you need to:
  - (a) understand **what** sorts of attacks are possible
  - (b) **how** exactly these attacks work
  - (c) **why** previous attempts failed
Prerequisites

- **CSCI 1951H (Software Security and Exploitation)**
  - Control-flow Hijacking
  - Code Injection (Shellcode dev.)
  - Code Reuse (ROP)

- **CSCI 1670 (Operating Systems)**
  - C/C++, x86 *asm*
  - Linking and Loading
  - Virtual Memory

Having taken the following courses is a plus, but not required:

- **CSCI 1660 (Computer Systems Security)**
- **CSCI 2951E (Topics in Computer System Security)**
Prerequisites

- **CSCI 1951H** (Software Security and Exploitation)
  - Control-flow Hijacking
  - Code Injection (Shellcode dev.)
  - Code Reuse (ROP)

- **CSCI 1670** (Operating Systems)
  - C/C++, x86 \texttt{asm}
  - Linking and Loading
  - Virtual Memory

- Having taken the following courses is a plus, but not required:
  - **CSCI 1660** (Computer Systems Security)
  - **CSCI 2951E** (Topics in Computer System Security)
Prerequisites

- **CSCI 1951H (Software Security and Exploitation)**
  - Control-flow Hijacking
  - Code Injection (Shellcode dev.)
  - Code Reuse (ROP)
- **CSCI 1670 (Operating Systems)**
  - C/C++, x86 **asm**
  - Linking and Loading
  - Virtual Memory

✔ Having taken the following courses is a plus, but not required:
  - **CSCI 1660** (Computer Systems Security)
  - **CSCI 2951E** (Topics in Computer System Security)

🌟 We will review (most of) the important concepts
Logistics

Meetings
• Mondays, 3PM – 5:20PM (M hour)
  ▶ CIT 506

Grading
• Paper reviews: 10%
• Paper presentations: 20%
• Discussion part: 20%
• Project report: 40%
• Project presentation: 10%

Communication
• http://cs.brown.edu/courses/csci2951-u/
• course.csci.2951u.2017-spring.s01@lists.brown.edu
• Check the website!
• Announcements
  • Lecture slides
  • Readings

Study material
• No required textbook
  • Assigned readings

vpk@cs.brown.edu (Brown University)
Logistics

 Meetings

• Mondays, 3PM – 5:20PM (*M hour*)
• CIT 506
## Logistics

<table>
<thead>
<tr>
<th>Meetings</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Mondays, 3PM – 5:20PM (M hour)</td>
<td>• <a href="http://cs.brown.edu/courses/csci2951-u/">http://cs.brown.edu/courses/csci2951-u/</a></td>
</tr>
<tr>
<td>• CIT 506</td>
<td>• <a href="mailto:course.csci.2951u.2017-spring.s01@lists.brown.edu">course.csci.2951u.2017-spring.s01@lists.brown.edu</a></td>
</tr>
</tbody>
</table>

@ Check the website!

• Announcements
• Lecture slides
• Readings

■ No required textbook
Logistics

Meetings

- Mondays, 3PM – 5:20PM (**M hour**)
- CIT 506

Communication

- [http://cs.brown.edu/courses/csci2951-u/](http://cs.brown.edu/courses/csci2951-u/)
- [course.csci.2951u.2017-spring.s01@lists.brown.edu](mailto:course.csci.2951u.2017-spring.s01@lists.brown.edu)

**Check the website!**

- Announcements
- Lecture slides
- Readings
Logistics

**Meetings**
- Mondays, 3PM – 5:20PM (*M hour*)
- CIT 506

**Grading**
- ✔ Paper reviews → 10%
- ✔ Paper presentations → 20%
- ✔ Discussion part. → 20%
- ✔ Project report → 40%
- ✔ Project presentation → 10%

**Communication**
- [http://cs.brown.edu/courses/csci2951-u/](http://cs.brown.edu/courses/csci2951-u/)
- [course.csci.2951u.2017-spring.s01@lists.brown.edu](mailto:course.csci.2951u.2017-spring.s01@lists.brown.edu)

**Check the website!**
- Announcements
- Lecture slides
- Readings
Logistics

 Meetings

- Mondays, 3PM – 5:20PM (M hour)
- CIT 506

 Grading

- Paper reviews → 10%
- Paper presentations → 20%
- Discussion part. → 20%
- Project report → 40%
- Project presentation → 10%

 Communication

- http://cs.brown.edu/courses/csci2951-u/
- course.csci.2951u.2017-spring.s01@lists.brown.edu

 Check the website!

- Announcements
- Lecture slides
- Readings

 Study material

- No required textbook → Assigned readings

vpk@cs.brown.edu (Brown University)

CSCI 2951U
Instructor

Vasileios (Vasilis) Kemerlis
  • vpk@cs.brown.edu
  • https://www.cs.brown.edu/~vpk
Office hours: Mon. 6PM – 8PM (CIT 505)
Virtual function of the old object is called and the virtual function pointer is looked up, the contents of the new object will be interpreted as the vtable pointer of the old object. This allows the corruption of the vtable pointer, comparable to exploiting a spatial write error, but in this case the dangling pointer is only dereferenced for a read. An additional aspect of this attack is that the new object may contain sensitive information that can be leaked when read through the dangling pointer of the old object’s type.