Lec12: Designing Heap Allocators

Taesoo Kim

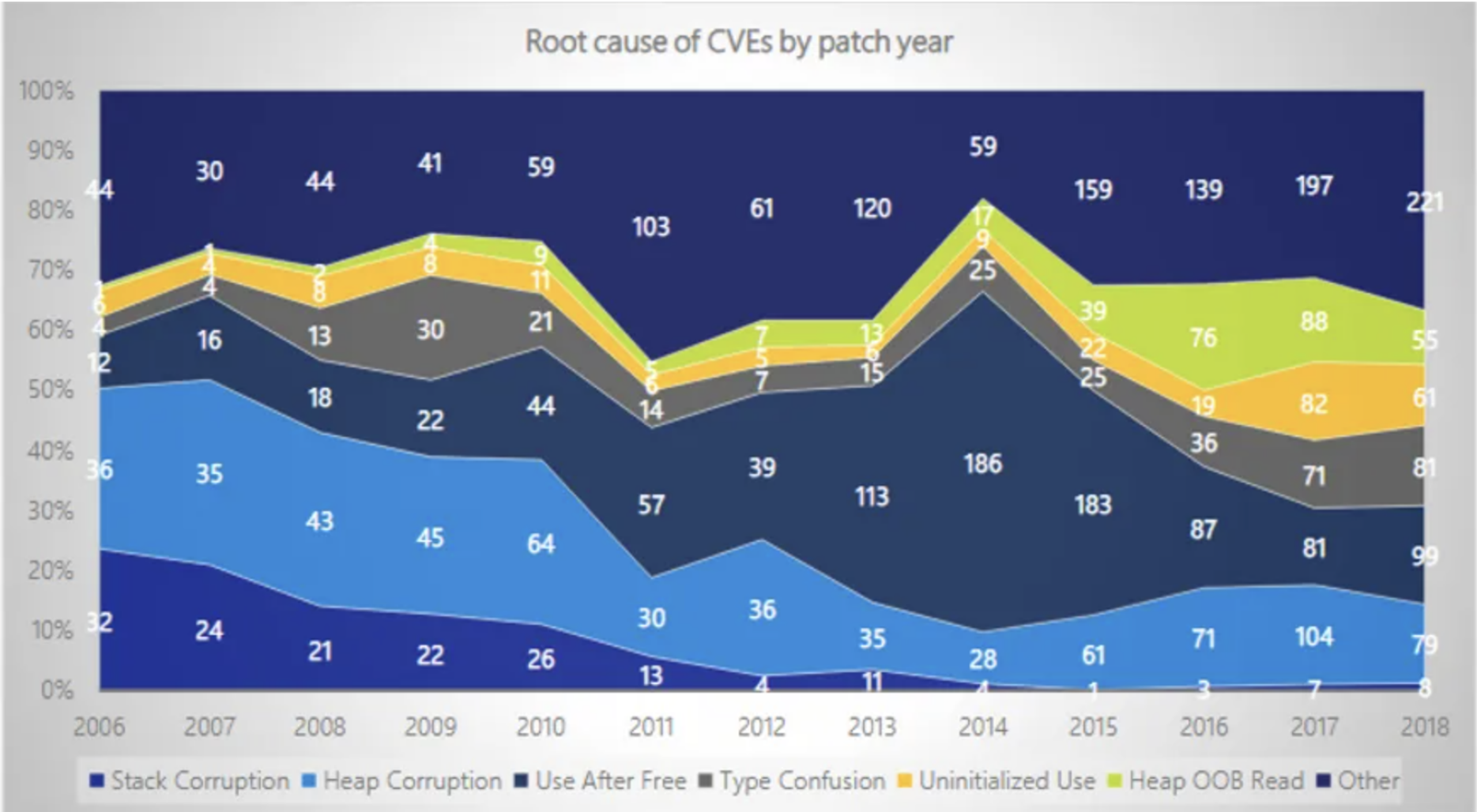
Administrivia

- In-class CTF on Dev 1 (24 hours)!
- Submit your team's challenge by Nov 27
- But submit it early for our feedback!
- Or you can brainstorm your challenge during the office hours!
- NSA Codebreaker Challenge → Due: Dec 08

Summary of Lab08

- **Integer overflows**: 2048, intq,
- **Race conditions**: race, tictou (web)
- **Sandbox bypassing**: seccomp
- **Miscellaneous**: django (pickle), simple-aeg (aeg), concat (api), type (c++), srop

Trends of Vulnerability Classes



Classifying Heap Vulnerabilities

- Common: buffer overflow/underflow, out-of-bound read
  - *Much prevalent* (i.e., quality, complexity)
  - *Much critical* (i.e., larger attack surface)
- Heap-specific issues:
  - **Use-after-free** (e.g., dangled pointers)
  - Incorrect uses (e.g., double frees)

Let’s Design Heap Allocators Together!

See, Lecture Note and Whiteboard

Lab09: Heap Exploitation

- Various malloc implementation (e.g., dlmalloc, ptmalloc)
- Use-after-free
- Double-free techniques

Today's Tutorial

- In-class tutorial:
  - Your first heap exploitation
  - Exploring heap memory structure

```
$ ssh lab09@54.88.195.85
Password: <password>

$ cd tut09-heap
```

References

- CVE-2014-0160
- CVE-2018-11360
- CVE-2018-17182
- Vudo - An object superstitiously believed to embody magical powers