Firefox Exploitation

Patroklos Argyroudis <argp@census-labs.com>
Chariton Karamitas <huku@census-labs.com>

AthCon 2013

Who we are

- Patroklos Argyroudis, **argp**
  - Researcher at Census, Inc.
  - Kernel/heap exploitation, source/binary auditing
- Chariton Karamitas, **huku**
  - Researcher at Census, Inc.
  - Compilers, reversing, exploitation, formal methods

Outline

- High level overview of the **jemalloc** allocator
- Attack techniques against **jemalloc** and Firefox
- UAF a.k.a. use after **free()**
- **XMLSerializer()** UAF vulnerability (CVE-2013-0753)
- Demonstration of **unmask_jemalloc**

Yummy jemalloc

- Several flavors (Mozilla Firefox, FreeBSD, standalone, Linux port and probably more?)
- Used by:
  - NetBSD & FreeBSD C library
  - Mozilla Firefox (pretty much everywhere)
  - Facebook
  - DefCon CTF

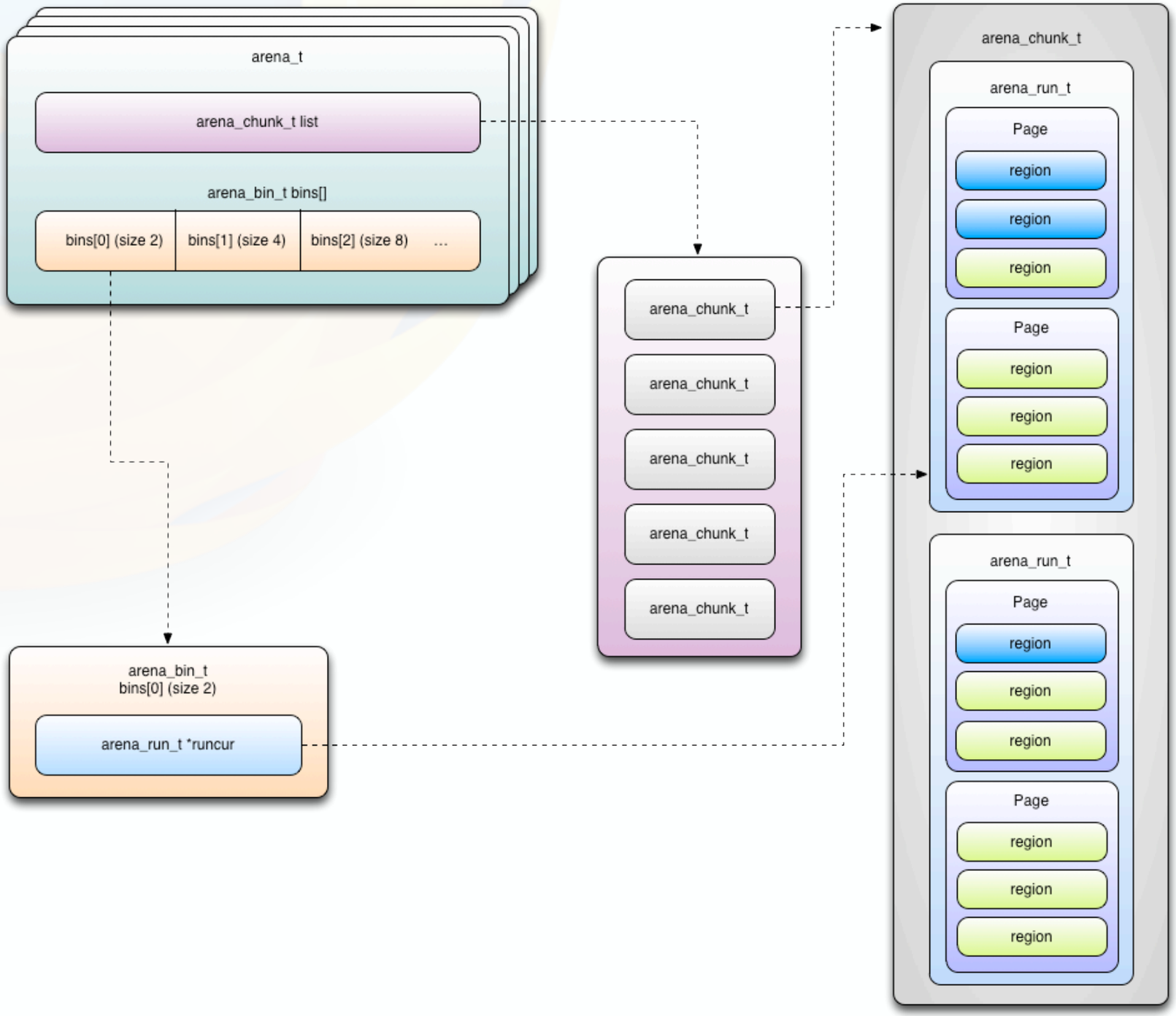
Allocator principles

- Minimal page utilization is not important anymore
  - You can buy few gigabytes of RAM with a few drachm… euros!
- Major design goal: Enhanced performance in retrieving data from RAM
- Principle of locality
  - Allocated together, used together (temporal and spatial locality)
  - **Effort to situate allocations contiguously in memory**

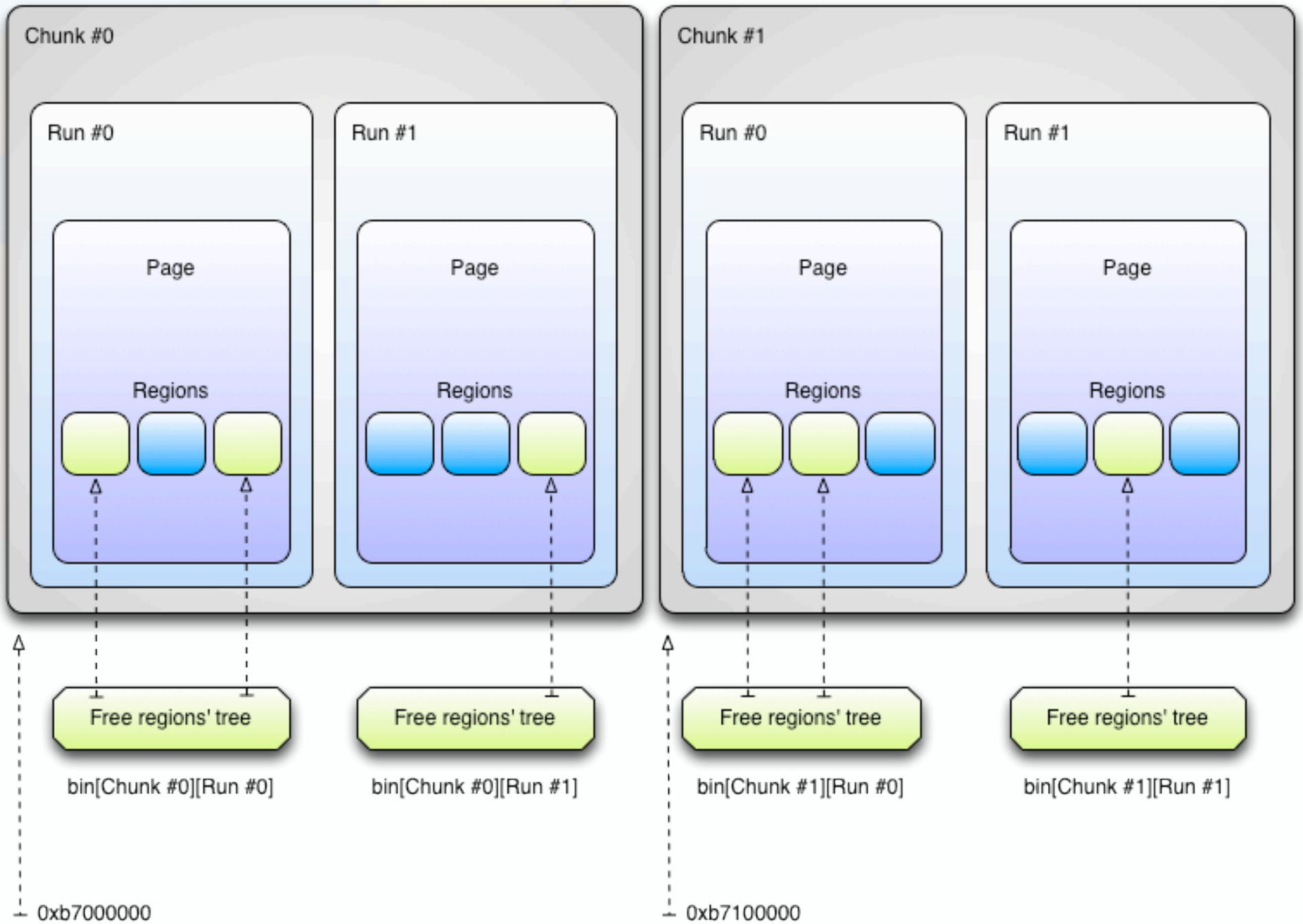
Firefox

- Firefox switched to jemalloc for dynamic memory management
- On all its supported platforms: Windows, Linux, OS X, Android
- Used for: DOM objects, JS objects (strings, array elements, function arguments, other JS API objects, etc)

jemalloc architecture



jemalloc architecture



unmask_jemalloc

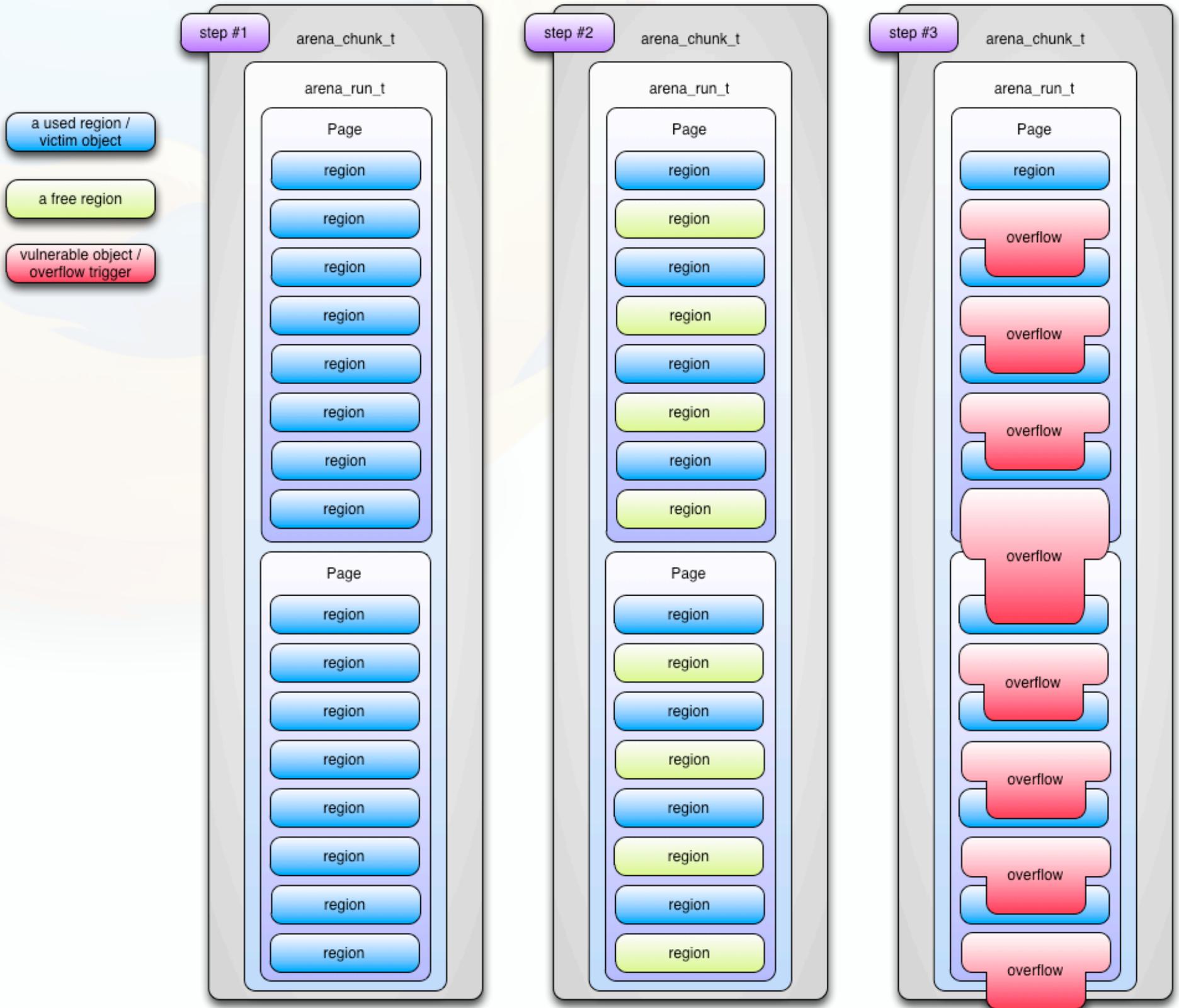
demo

https://github.com/argp/unmask_jemalloc

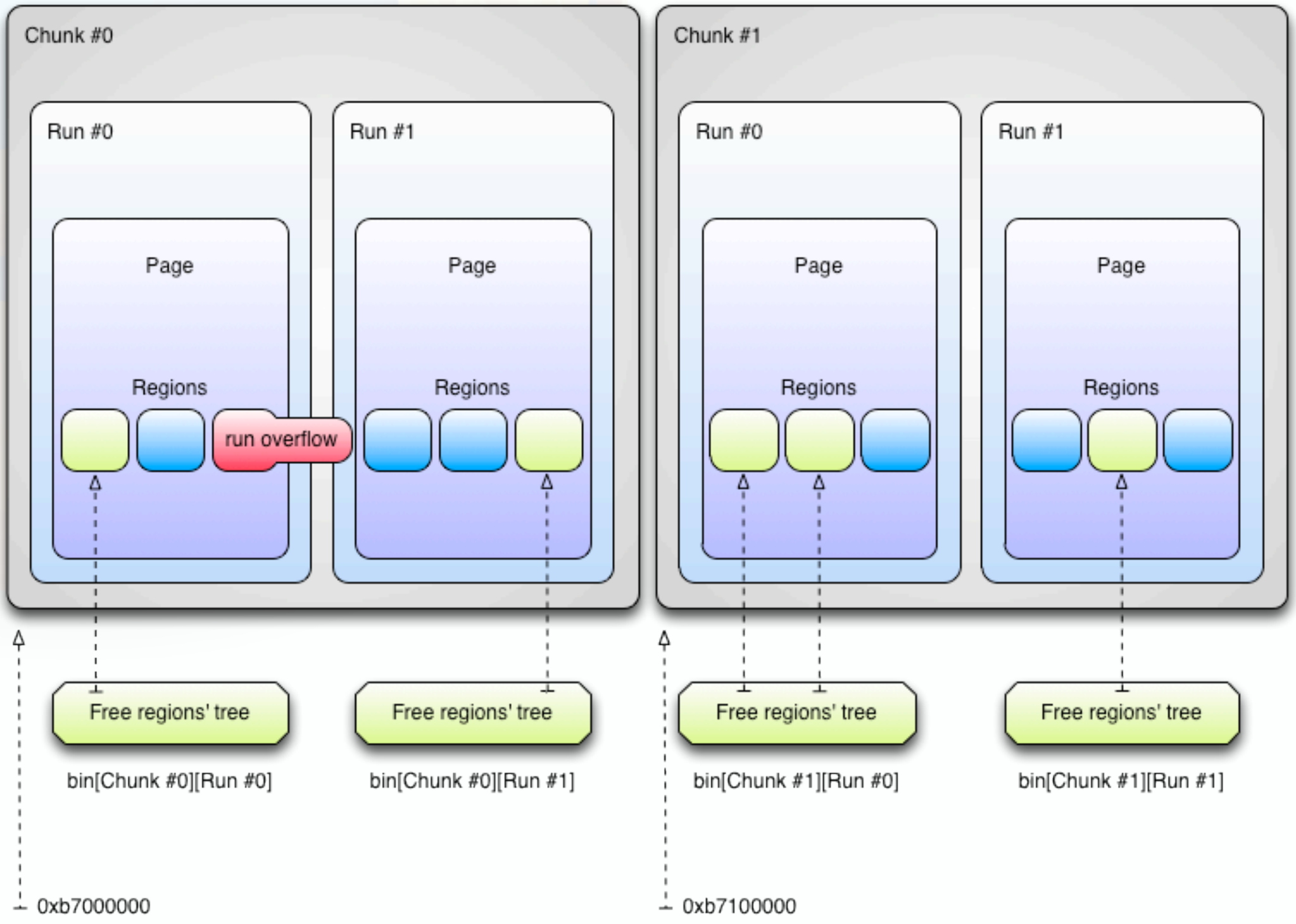
Exploitation techniques

- Adjacent region corruption [BlackHat]
- Run header corruption [BlackHat]
- Chunk header corruption [Phrack]
- Magazine (thread cache) corruption [Phrack]
- Double free - future work (maybe ;)
- Use after free - our focus for this presentation

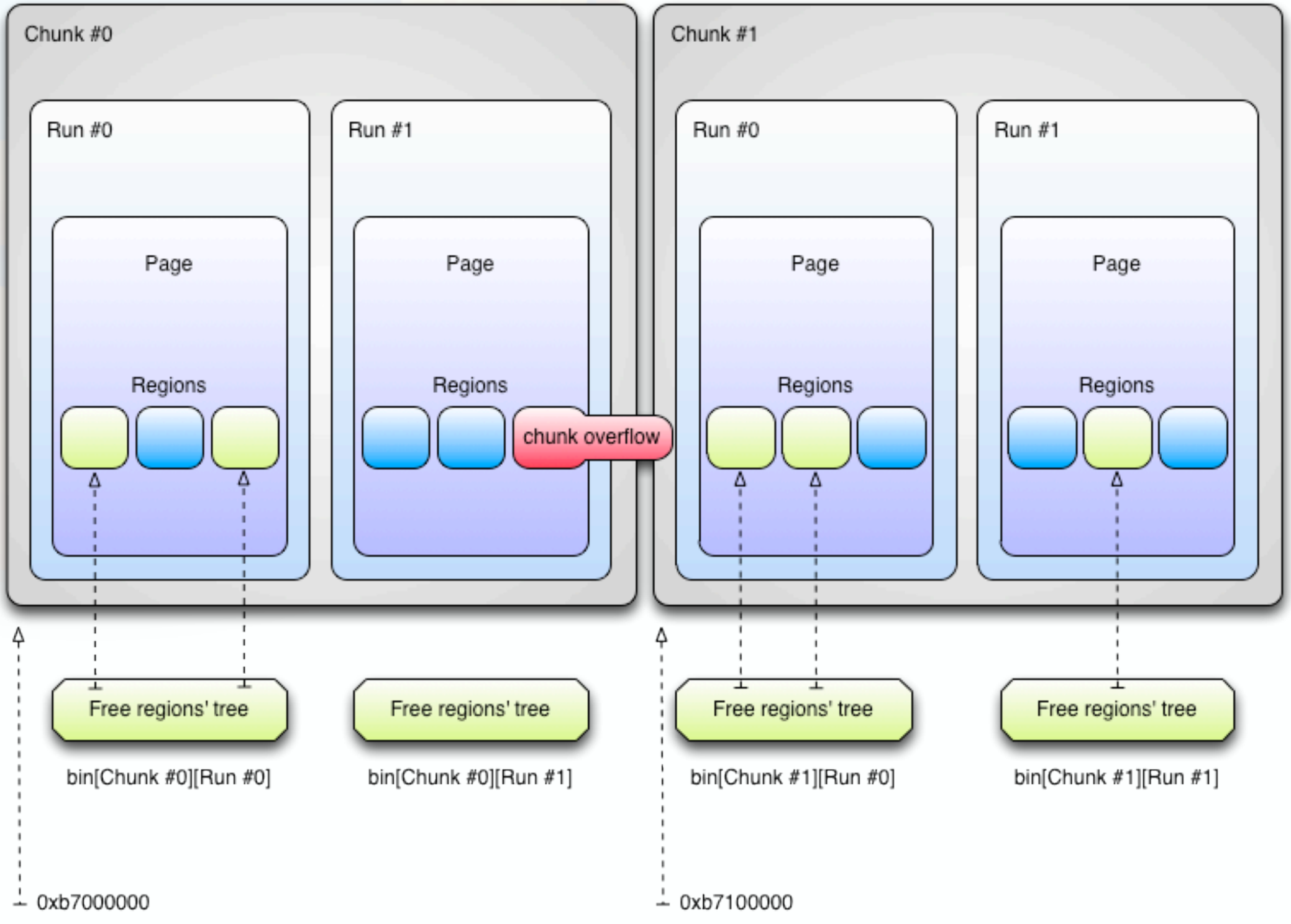
Adjacent region corruption



Run header corruption



Chunk header corruption



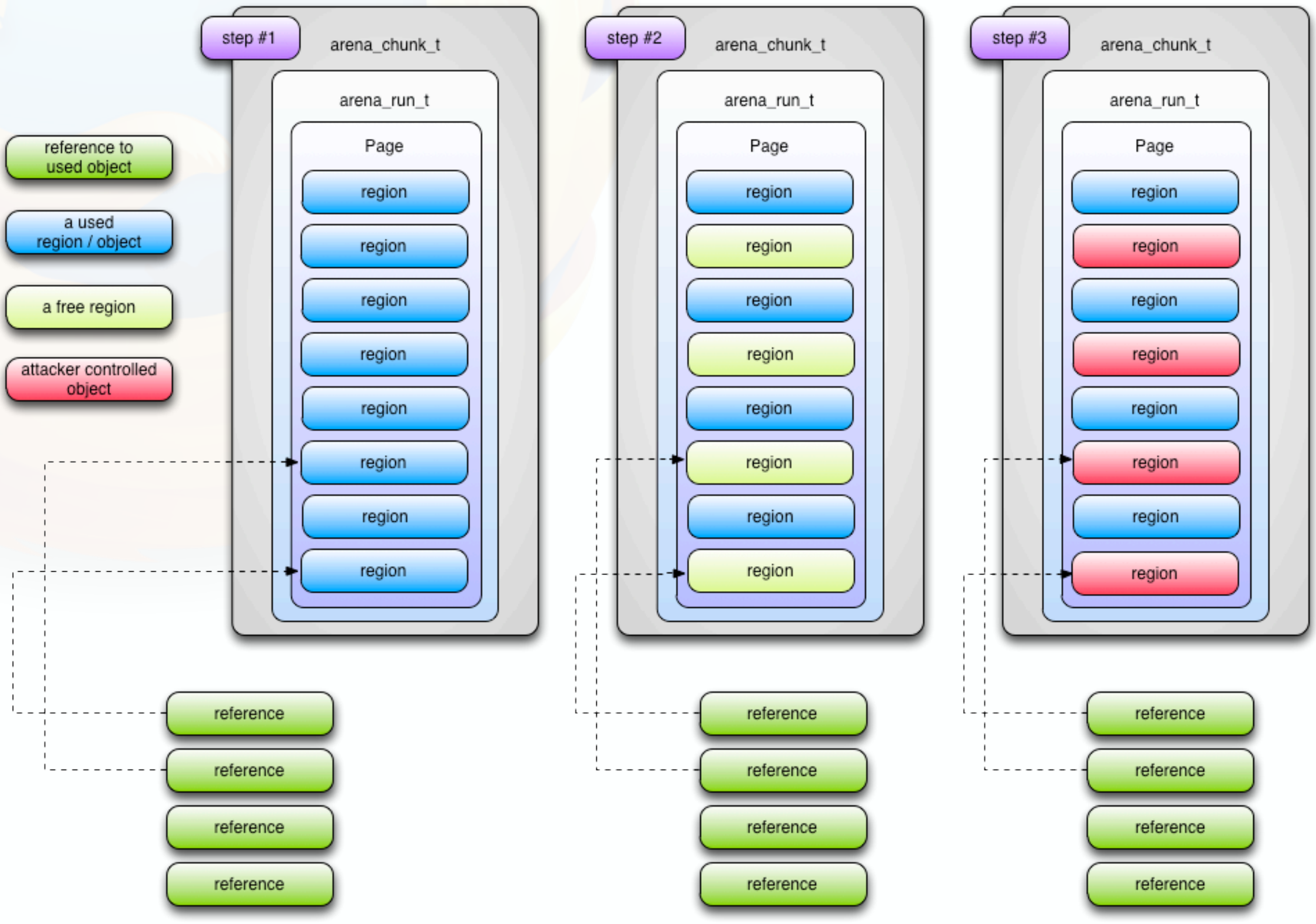
Use after free()

- One of the most prevalent vulnerability classes in the past few years
- Quoting http://cwe.mitre.org/data/definitions/416.html:
  - “The use of previously-freed memory can have any number of adverse consequences, ranging from the corruption of valid data to the execution of arbitrary code, **depending on the instantiation and timing of the flaw**.”

Use after free()

- Heap region gets allocated (usually holds C++ class instance), call it **A**
- **A** is referenced in object **B** (probably a C++ container object)
- **A** gets **free()**'ed but dangling references (**B**) remain
- Attacker gains control of **free()**'ed region contents by manipulating the heap (e.g. using heap spraying)
- **A** is dereferenced via the reference we hold (**B**), e.g. calling its functions, accessing its attributes
- Code execution :)

Use after free()



CVE-2013-0753

- Quoting https://developer.mozilla.org/en-US/docs/XMLSerializer:
  - "**XMLSerializer** can be used to convert DOM subtree or DOM document into text. **XMLSerializer is available to unprivileged scripts**."
- **XMLSerializer** traverses the DOM tree to figure out which node to serialize and how to serialize it
  - Shit happens when the DOM tree is modified while Firefox serializes it

CVE-2013-0753

Original

```
mStream = aStream;
...
rv = EncodeToString(buf);
...
FlushText(buf, true);
```

Patched

```
rv = EncodeToString(buf);
...
mStream = aStream;
...
FlushText(buf, true);
```

XMLSerializer exploit demo

References

- [BlackHat] https://www.blackhat.com/html/bh-us-12/bh-us-12-archives.html#Argyroudis
- [Phrack] http://phrack.org/issues.html?issue=68&id=10#article
- [unmask_jemalloc] https://github.com/argp/unmask_jemalloc

Questions?


IN CASE OF EMERGENCY
EMERGENCY EXIT
SORTIE D'URGENCE
BECOME THOR