

To provide an appropriate level of protection in the transmission of your digital data.

Target audience

Everyone performing any function related to the development of information system and network monitoring (research and development manager, project leader, developer...)

Goal

The purpose of this training is to improve the cybersecurity (and the quality as a side effect) of your C codes. After the presentation of the most common errors in the field, this training will give you the good practices and the tools to minimize the vulnerabilities in software development.

Location

Within your company.

Duration

It is a 2-day training.



```

    .find( ".item.active" );this.$items=this.$active.parent(
    this.getActiveIndex(),r=this;if(t>this.$items.length-1
    r.to(t)):n==t?this.pause().cycle():this.slide(t>n?"ne
    this.$element.find(".next, .prev").length&&e.support.tr
    cle(!0)),clearInterval(this.interval),this.interval=nu
    t,prev:function(){if(this.sliding)return;return this.s
    ve"),i=n||r[t](),s=this.interval,o=t=="next"?left:"r
    t=i.length?i:this.$element.find(".item")[u](),f=e.Event
    this.$indicators.length&&(this.$indicators.find(".activ
    indicators.children()[a.getActiveIndex()]);t&&t.addClass
    e.$element.trigger(f);if(f.isDefaultPrevented())return
    e.support.transition.end,function(){i.removeClass([t,
    "]),a.sliding=!1,setTimeout(function(){a.$element.tri
    turn;r.removeClass("active"),i.addClass("active"),this
    ar n=e.fn.carousel;e.fn.carousel=function(n){return th
    carousel.defaults,typeof n=="object"&&n,o=typeof n=="
    to(n):o?i[o]():s.interval&&i.pause().cycle()}}},e.fn.c
    structor=t,e.fn.carousel.noConflict=function(){return
    , [data-slide-to]",function(t){var n=e(this),r,i=e(n.
    
```

Training programme

Day 1 :

- A bug example that becomes a vulnerability : the buffer overflow, battery operation
- The list and bug practices (format string, integer overflow...)
- The risks of dynamically allocated memory
- The dangers of multithreading
- Differences between kernel programming and userland
- Minimization of privileges (chroot, strlimit...)
- Cybersecurity principles : minimization of attack surface, in-depth defense

Day 2 :

- Workflow and versioning code (git, ticketing)
- Code documentation, doxygen example, performances
- Tests : unitary, non-regression, integration, performances
- Continuous integration (buildbot)
- Dynamic analysis, valgrind, debugging (gdb, dynamRIO)
- Static analysis, code coverage, fuzzing
- Project management and cybersecured development
- Conclusion : top 10 secure development practices

In addition to this training, we recommend to you to carry out an audit to optimize your protections.

Contact us for more information



Rempart-International

26, rue de Louvigny-L 1946 Luxembourg

T : +352 288 557 - contact@rempart-international.com

RCS Luxembourg B196245- Autorisation ME N°10058858/0 - 10058858/1 - 10058858/2 - Autorisation MJ N° 22-2-800-458