

# Mestrado em Engenharia Informática (MEI)

# Mestrado Integrado em Engenharia Informática

## (MiEI)

Perfil de Especialização **CSI** : Criptografia e Segurança da Informação

Engenharia de Segurança



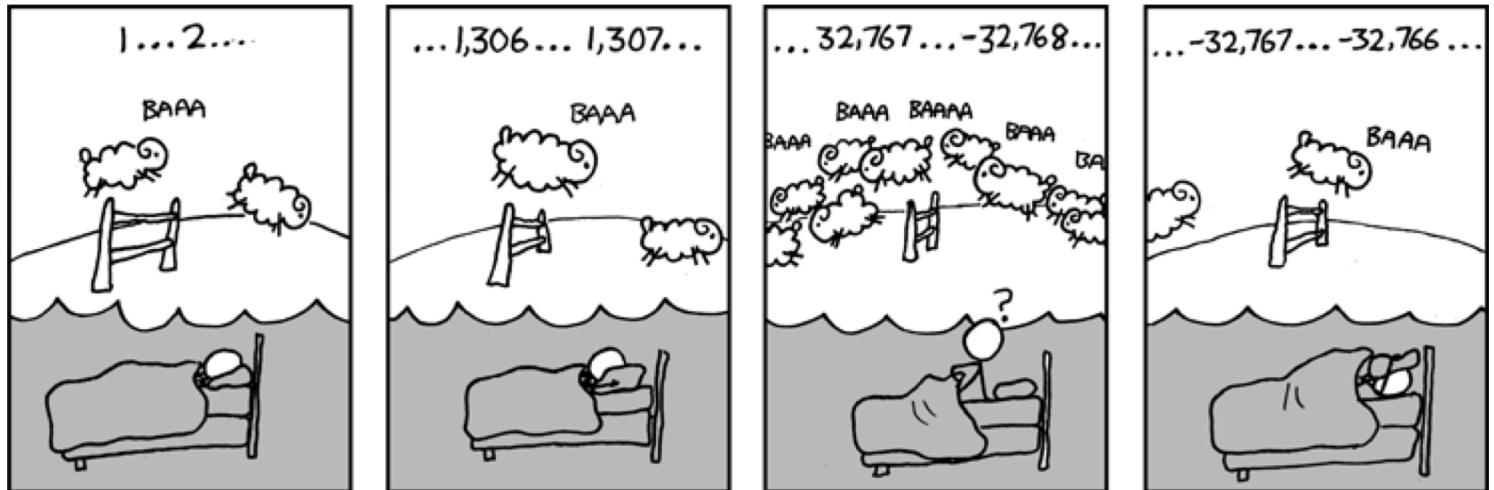
# Tópicos de Segurança de Software

- Vulnerabilidade de Inteiros



# Integer error

- Valores muito grandes ou muito pequenos de inteiros podem cair fora do intervalo do tipo de dados, levando a um comportamento indefinido que pode reduzir a robustez do código, assim como dar origem a vulnerabilidades de segurança.
- Por exemplo, um int de 32-bit pode conter valores de  $-2^{31}$  até  $2^{31}-1$ .
- Um erro de Inteiros pode levar a comportamento inesperado ou, pode ser explorado para causar o *crash* de um programa, corromper dados, levar a comportamento incorreto ou permitir a execução de software malicioso.



# Integer error

CVE ID	Vulnerability type	Publish Date	CVSS Score	Description
CVE-2019-9210	Integer Overflow or Wraparound	2019-02-27	7.8	In AdvanceCOMP 2.1, png_compress in pngex.cc in advpng has an integer overflow upon encountering an invalid PNG size, which results in an attempted memcpy to write into a buffer that is too small. (There is also a heap-based buffer over-read.)
CVE-2019-3857	Integer Overflow or Wraparound	2019-03-25	8.8	An integer overflow flaw which could lead to an out of bounds write was discovered in libssh2 before 1.8.1 in the way SSH_MSG_CHANNEL_REQUEST packets with an exit signal are parsed. A remote attacker who compromises a SSH server may be able to execute code on the client system when a user connects to the server.
CVE-2018-6543	Integer Overflow or Wraparound	2018-02-02	7.8	In GNU Binutils 2.30, there's an integer overflow in the function load_specific_debug_section() in objdump.c, which results in `malloc()` with 0 size. A crafted ELF file allows remote attackers to cause a denial of service (application crash) or possibly have unspecified other impact.
CVE-2016-3074	Denial Of Service Execute Code Overflow	2016-04-26	7.5	Integer signedness error in GD Graphics Library 2.1.1 (aka libgd or libgd2) allows remote attackers to cause a denial of service (crash) or potentially execute arbitrary code via crafted compressed gd2 data, which triggers a heap-based buffer overflow.
CVE-2018-6543	DoS Overflow	2018-02-02	6.8	In GNU Binutils 2.30, there's an integer overflow in the function load_specific_debug_section() in objdump.c, which results in `malloc()` with 0 size. A crafted ELF file allows remote attackers to cause a denial of service (application crash) or possibly have unspecified other impact.
CVE-2018-7471	Overflow	2018-02-25	7.5	KingView 7.5SP1 has an integer overflow during stgopenstorage API read operations.



# *Integer error*

- A maior parte dos sistemas Unix guarda a data/hora numa variável do tipo int 32-bit – a data/hora é guardada como o número de segundos desde as 00h00 de 1/Jan/1970. A 19/Jan/2038, ocorre o overflow dessa variável, passando a data/hora a ser negativa. Mais informação em [http://en.wikipedia.org/wiki/Year\\_2038\\_problem](http://en.wikipedia.org/wiki/Year_2038_problem)



# *Integer error*

- No Facebook existe um grupo que afirma o seguinte:



- Porque é que afirmam isso?
- Quais seriam os potenciais problemas se isso ocorresse?

# *Integer error*

- YouTube não aguentou o Gangnam Style

## **YouTube não aguentou o *Gangnam Style***

PÚBLICO 03/12/2014 - 16:38

É o próprio site a confirmar que teve problemas. "*Gangnam Style* foi visto tantas vezes que tivemos que fazer um *upgrade*."

É o próprio site a confirmar que teve problemas. "*Gangnam Style* foi visto tantas vezes que tivemos que fazer um *upgrade*", escreve o YouTube na sua página no Google+.

Se passarmos o cursor sobre o contador que se pode ver na página do vídeo este não pára de rodar e isso porque o YouTube nunca pensou que um tal número pudesse vir a ser atingido. "Nunca pensámos que um vídeo pudesse ser visto em números mais do que um número inteiro de 32-bit (=2.147.483.647 visualizações), mas isso foi antes de termos conhecido Psy", admite o YouTube.

# Integer error

- No dia 25 Dezembro 2004, a companhia aérea *Comair airlines* foi forçada a manter no solo 1.100 voos após o software de agendamento das tripulações colapsar. O software utilizava um inteiro de 16-bit (máximo 32.768) para numerar as alterações de tripulação durante um mês, tendo esse número sido excedido nesse mês devido a mau tempo que levou a inúmeras alterações de tripulação.

FEATURE

## Comair's Christmas Disaster: Bound To Fail

The 2004 crash of a critical legacy system at Comair is a classic risk management mistake that cost the airline \$20 million and badly damaged its reputation.



By Stephanie Overby

CIO | MAY 1, 2005 8:00 AM PT



ars TECHNICA

BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULT

UNCATEGORIZED —

## Comair/Delta airline debacle caused by the overflow of 16-bit pointer

One of the most nightmarish Christmas travel foul-ups in recent memory was ...

CLINT ECKER - 12/30/2004, 7:24 PM



# Integer error – problema de truncamento

- A 4 de Junho de 1996, o foguetão não tripulado Ariane 5 explodiu 40 segundos depois do lançamento. O foguetão fazia a sua primeira viagem após uma década de desenvolvimento, com custos na ordem dos \$7 biliões, estando o foguetão destruído e a sua carga avaliada em \$500 milhões.
- A causa da explosão foi um erro de software no sistema de referência de inércia. Mais especificamente, um número *float* de 64 bits relacionado com a velocidade horizontal do foguetão, foi convertido num inteiro de 16 bits (*signed int*). O número a converter era maior do que 32.767 (maior inteiro que pode ser guardado num *signed int*), pelo que a conversão falhou.



# *Integer error*

- Risco
  - Declarar uma variável com determinado tipo aloca um espaço fixo de memória. A maior parte das linguagens permite declarar diversos tipos de inteiros (short, int, long, etc.). Por exemplo, um int de 32 bits pode guardar valores entre  $-2^{31}$  (-2 147 483 648) e  $2^{31}-1$  (2 147 483 647).
  - Muitas vezes, o tamanho dos tipos de dados são dependentes da máquina e do compilador ...
- Codificação “responsável”
  - Conhecer os limites: como o tamanho do tipo de dados é dependente de máquina e compilador é uma boa ideia familiarizar-se com os limites na máquina onde o programa vai executar;
  - Tipo de dados: escolha o tipo de inteiro mais adequado para os valores que vai conter, na linguagem de programação que está a utilizar;
  - Validar o input: (mais detalhado na próxima secção);
  - Validar possíveis overflows/underflows antes de operações sobre inteiros; 
  - Configurar parâmetros do compilador: opções que verificam potenciais erros;
  - Utilizar bibliotecas específicas: por exemplo, a classe SafeInt no C++.