

[PREVIOUS](#)[NEXT](#)

TECHNOLOGY

10 Seriously Epic Computer Software Bugs

ANDREW JONES DECEMBER 24, 2012



The majority of software bugs are small inconveniences that can be overcome or worked around by the user – but there are some notable cases where a simple mistake has affected millions, to one degree or another, and even caused injury and loss of life.

Software is written by humans – and every piece of software therefore has bugs, or “undocumented features” as a salesman might call them. That is, the software does something that it shouldn’t, or doesn’t do something that it should. These bugs can be due to bad design, misunderstanding of a problem, or just simple human error – just like a typo in a book. However, whereas a book is read by a human who can usually infer the meaning of a misspelled word, software is read by computers, which are comparatively stupid, and will do only what they’re told.

Here are ten cases where the consequences of these bugs were enormous, in some way or another:



The Therac-25 was a machine for administering radiation therapy, generally for treating cancer patients. It had two modes of operation. The first consisted of an electron beam targeted directly at the patient in small doses for a short amount of time. The second aimed the electron beam at high energy levels at a metal 'target' first, which would essentially convert the beam into X-rays that were then passed into the patient.

In previous models of the Therac machine, for this second mode of operation, there were physical fail-safes to ensure that this target was in place as, without it, very high energy beams could be mistakenly fired directly into the patient. In the new model, these physical fail-safes were replaced by software ones.

Unfortunately, there was a bug in the software: an 'arithmetic overflow' sometimes occurred during automatic safety checks. This basically means that the system was using a number inside its internal calculations that was too big for it to handle. If, at this precise moment, the operator

was configuring the machine, the safety checks would fail and the metal target would not be moved into place. The result was that beams 100 times higher than the intended dose would be fired into a patient, giving them radiation poisoning. This happened on 6 known occasions, causing the later death of 4 patients.

9 World of Warcraft "Corrupted-Blood" Glitch September 13, 2005



The hugely successful World of Warcraft (WoW), an online computer game created by Blizzard Entertainment, suffered an embarrassing glitch following an update to their game on September 13, 2005 – causing mass (fictional) death. Following an update to the game content, a new enemy character, Hakkar, was introduced who had the ability to inflict a disease, called Corrupted Blood, upon the playing characters that would drain their health over a period of time. This disease could be passed from player to player, just as in the real world, and had the potential to kill any character contracting it. This effect was meant to be strictly localised to the area of the game that Hakkar inhabited.

However, one thing was overlooked: players were able to teleport to other areas of the game while still infected and pass the disease onto others – which is exactly what happened. I can't find any figures on the body count, but entire cities within the game world were no-go areas, with

dead player's corpses littering the streets. Fortunately, player death is not permanent in WoW and the event was soon over when the administrators of the game reset the servers and applied further software updates. Particularly interesting is the way players reactions in the game could closely reflect their reactions to a similar real-life incident.

8

North American Blackout
August 14, 2003



Affecting around 55 million people, mainly in the North Eastern United States, but also Ontario Canada, this was one of the biggest power blackouts in history. It started when a power plant along the southern shore of Lake Erie, Ohio went offline due to high demand which put the rest of the power network under greater stress. When power lines are under heavier electrical load, they heat up, meaning the material making up the cable (usually aluminum and steel) expands. Several power lines hung lower as they expanded and caught trees, bringing them down and putting the system under yet more pressure. This led to a cascading effect that eventually reduced the power network to 20% of normal output.

While the causes of this blackout were nothing to do with a software bug, it could have been averted were it not for a software bug in the control centre alarm system. In what is called a 'race condition' scenario, two parts of the system were competing over the same resource and were unable to resolve the conflict, which caused the alarm system to freeze and stop processing alerts. Unfortunately, the alarm system failed 'silently', meaning it broke, but didn't notify anybody that it had broken. This meant no audio or visual alerts were provided to control room staff, who over relied on such things for situational awareness. The aftermath was well reported and left many areas without power for several days and affected industry, utilities, communication. It was also blamed as at least a contributing factor in several deaths.

7

USS Yorktown Incident September 21, 1997



In the world of software development, there are several commonly known bugs that programmers encounter and have to cater for. One such example is the 'divide by zero' bug, where a calculation is performed that divides any number by zero. Such a calculation isn't possible to resolve, at least not without using higher mathematics, and most software – for everything from super computers to pocket calculators – is written to take this scenario into account.

It was with some embarrassment, then, that the USS Yorktown suffered a complete failure of its propulsion system and was dead in the water for nearly 3 hours when a crew member typed a "0" into the on-board database management system which was then used in a division calculation. The software was installed as part of a wider operation to use computers to reduce the manpower needed to run some ships. Fortunately, the ship was engaged in maneuvers at the time of the incident, rather than deployed in a combat environment, which could have had more severe consequences.

6 Trans-Siberian Gas Pipeline Explosion 1982



This one is a bit of a stretch, and may never have in fact happened, but – if it is true – it is a prominent example of a deliberately introduced software bug causing a big incident.

During the Cold War, when relations between the US and Soviet Russia were a tad frosty, the Central Intelligence Agency are said to have deliberately placed bugs inside software being sold by a Canadian company -software that was used for controlling the trans-siberian gas pipeline. It was thought by the CIA that Russia was purchasing this system via a Canadian company as a means of covertly obtaining US technology, and that this would be an opportunity to feed them defective material.

Such practices were later referenced in the declassified "Farewell Dossier" where, amongst other things, it is alleged that faulty turbines were in fact used on a gas pipeline. It is claimed by former Secretary of the Air force, Thomas C. Reed, that a series of bugs were introduced so that the system would pass tests but break during actual use. Settings for pumps and valves were set to exceed the pressures that the pipeline could withstand, which led to an explosion said to be the largest non-nuclear explosion ever recorded.

These claims, however, have been contradicted by KGB veteran, Anatoly Medetsky, who claims that the explosion was caused by sub-par construction rather than deliberate sabotage. Whatever the cause, no known casualties were reported as the explosion occurred in a very remote area.

5 Cold War Missile Crisis September 26, 1983



Stanislav Petrov was the duty officer of a secret bunker near Moscow responsible for monitoring the Soviet early warning satellite system. Just after midnight, they received an alert that the US had launched five Minuteman intercontinental ballistic missiles. As part of the mutually assured destruction doctrine that came into prevalence during the Cold War, the response to an attack by one power would be a revenge attack by the other.

This meant that if the attack was genuine, they needed to respond quickly. However, it seemed strange that the US would attack with just a handful of warheads: although they would cause massive damage and loss of life, it wouldn't be even nearly sufficient to wipe out the Soviet opposition. Also, the radar stations on the ground weren't picking up any contacts, although these couldn't detect beyond the horizon because of the curvature of Earth, which could have explained the delay.

Another consideration was the early warning system itself, which was known to have flaws and had been rushed into service in the first place. Petrov weighed all these factors and decided to rule the alert as a false alarm. Although Petrov didn't have his finger on the nuke button as such, had he passed on a recommendation to his superiors that they take the attack as real, it could have led to all-out nuclear war. Whether based on experience, intuition, or just luck, Petrov's decision was the right one.

It was later determined that the early detection software had picked up the sun's reflection from the top of clouds and misinterpreted it as missile launches.

4 Sony CD Malicious Copy Protection



The seemingly never-ending war between media and pirates ebbs and flows every year. As soon as new ways of protecting and securely distributing media is found, new ways of circumventing and compromising these measures are uncovered.

Some would argue that Sony BMG went a step too far in 2005, when they introduced a new form of copy protection on some of their audio CDs. When played using a Windows computer, these CDs would install a piece of software called a 'rootkit'. A rootkit is a form of software that buries its way deep into a computer and alters certain fundamental processes. Though not always malicious in nature, a rootkit is often used to stealthily plant malicious and hard to detect (or remove) software, such as viruses, trojans etc. In the case of Sony BMG, the aim was to control the way a Windows computer used the Sony CDs to prevent copying them or converting them to MP3s, which would help them cut down on piracy of their media.

The rootkit achieved this – but by taking measures to hide itself from the user, it enabled viruses and other malicious software to hide along with it. The poorly thought-out implementation, and a growing perception that Sony BMG had no business sneakily manipulating users PCs, meant that the whole scheme backfired. It resulted in the rootkit being classified as malware by many computer security companies, as well as several law suits and a product recall of the offending CDs.

3 Patriot Missile Bug February 25th, 1991



During Operation Desert Shield, the US military deployed the Patriot Missile System as a defense against aircraft and missiles – in this case Iraqi Al Hussein (SCUD) missiles. The tracking software for the Patriot missile uses the velocity of its target and the current time to predict where the target will be from one instant to another. Since various targets may travel at speeds of up to MACH 5, these calculations need to be very accurate.

At the time, there was a bug in the targeting software – which meant that over time, the internal clock would ‘drift’ (much like any clock) further and further from accurate time the longer the system was left running. The bug was actually already known about and was simply fixed by regularly rebooting the system, and thereby resetting the system clock.

Unfortunately, those in charge didn’t clearly understand how ‘regularly’ they should reboot the system, and it was left running for 100 hours. When an Iraqi missile was launched, targeting a US

airfield in Dhahran, Saudi Arabia, it was detected by the Patriot missile system. However, by this point, the internal clock had drifted out by 0.34 of a second, so when it tried to calculate where the missile would be next, it was looking at an area of the sky over half a kilometer away from missiles true location. It promptly assumed there was no enemy missile after all and cancelled the interception. The missile carried on to its destination where it killed 28 soldiers and injured a further 98.

2 Millennium Bug



The Millennium Bug, or Y2K, is the best known bug on this list and the one that many of us remember hearing about at the time. Basically, this bug was the a result of the combined short-sightedness of computer professionals in the decades leading up to the year 2000. In many computer systems, two digits were used to show the date, e.g. 98 instead of 1998, a practice that seemed reasonable and which pre-dated computers by some time.

Many didn't anticipate, however, that there may be a problem when the date went past the year 2000. Using current systems, the year 2000 could only be represented as '00', which might confuse computers into thinking it meant the year 1900. Such a thing would break any calculations involving ranges of years that crossed the millennium. For example, it might show somebody born in 1920 and dying in 2001 as being minus 19 years old.

In response to the problem, software companies rapidly updated their products, which already controlled just about everything from banking and payrolls to hospital computers and train ticket systems. Also, in recognition of its worldwide nature, the International Y2K Cooperation Centre was created in February 1999 to help coordinate the work required to prepare for the new millennium between governments and organisations, where needed. In the end, the New Year passed without too much incident, besides the universal mother-of-all-hangovers.

It's hard to say how much of this success was a result of the work carried out to alleviate the problem, or whether the problem had been exaggerated in the media in the first place – probably a mix of both.

1 Year 2038



Although Y2K is passed, we're not out of the woods just yet. Not all computers handle dates in the same way, and many computers based on the UNIX operating system handle dates by counting how many seconds a date is since 01/01/1970. For example, the date 01/01/1980 is 315,532,800 seconds after 01/01/1970. This number is stored on these computers as a "signed 32-bit integer", which has a size limit of 2147483647. That basically means it can only handle dates that are up to 2147483647 seconds after 01/01/1970 – which only takes us up to the 19th of January 2038, after which, we may have problems again.

This is especially true when we consider that UNIX-based software is more commonly used in "embedded systems" rather than a home PC – that is, systems that have a very specific purpose closely related to their hardware, such as software for robotic assembly lines, digital clocks, network routers, security systems and so on.

Also, somebody is going to have to consider what we're going to do on the 1st of January 10000. Not me though.



Follow us on [Facebook](#) or subscribe to our [daily or weekly newsletter](#) so you don't miss out on our latest lists.

Sponsored Web Content



Netflix Announces Binge-Worthy December Lineup

[Refinery29](#)



Wife Dies Hours After Giving Birth. Then Husband's Gut Tells Him To Log Into Her...

[Scribol](#)



[Gallery] Boy Missing For 4 Years Found Hidden Inside Wall, Thanks To Clues He Left For Police

[Scribol](#)



[Gallery]See How The "Perfect" Female Body Has Changed Over 100 Years

[Healthy Living](#)



If You Have Any Of These 20 Toys Around, You Just Became Rich

[Definition](#)



Treatment Options Increasing for COPD

[Healthgrades](#)

More From Listverse



10 Plans Christian Radicals Have For America



10 People Who Found Big Money And Returned It



Top 10 Hidden Images Found In Cartoons



Top 10 Religions You Never Knew Existed



10 New Facts To Make You Seem Smarter



Top 10 Worst Engineering Disasters

MORE GREAT LISTS



TECHNOLOGY

10 Easy Arithmetic Tricks

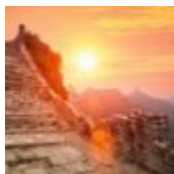
SEPTEMBER 17, 2007



TECHNOLOGY

10 Recent Scientific Advances That Signal The Future

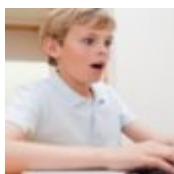
JULY 12, 2013



TECHNOLOGY

10 Silly Myths About Famous Architectural Wonders

FEBRUARY 12, 2015



TECHNOLOGY

10 Overwrought Media Scare Campaigns About The Internet

SEPTEMBER 12, 2015

30 Comments Listverse

 Login ▾ Recommend 1  Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

 | Log InOR SIGN UP WITH DISQUS 

Name

Email

Password

By signing up, you agree to the Disqus [Basic Rules](#), [Terms of Service](#), and [Privacy Policy](#).

☐ I'd rather post as a guest**formerly timmar** • 5 years ago

When midnite of the year 2000 came, the tv we were watching coincidentally had a blip at that exact moment. We all yelled, "Y2K is real!" then the tv went back on. What are the odds? lol

23 ^ | ▾ • Reply • Share >

oouchan • 5 years ago

I found this to be quite good. When too much control is given to the robotic mind, stuff like this happens....and I suspect we will see much more. "Ghosts in the machine." Since the movie iRobot, this phrase stuck with me. Working in computers as I do, I see things like this. Random bits of code that just perform weirdly once thrown into the mix.

Interesting list.

8 ^ | ▾ • Reply • Share >

Ni99a ➔ oouchan • 5 years ago

I HTML, therefore I code.
lol.

The kitchen is that way.

19 ^ | ▾ • Reply • Share >

TeoneSofle ➔ oouchan • 4 years ago

Yes, but who creates these "robots" or software. In the end, it is still human error.

^ | ▾ • Reply • Share >

Mom424 Mod • 5 years ago

Holy crap on #10. Like going for radiation treatment isn't bad enough already; now you've got to worry about the software killing you too.

Great list.

7 ^ | v • Reply • Share >

burroughs_tony • 5 years ago

Great list, really intetesting stuff that I never knew about or even thought about. I appreciate all of the detailed information you laid out too. Sometimes it really annoys me when lists are too long and way over-explained but here it totally works. I would much rather have all of the information and facts supporting the list than simply saying something and expecting me to believe and accept it. Anyway great job, it's nice to see the effort you put into it... unlike that horrible thing on "facts" about color. This was a very nice change.

3 ^ | v • Reply • Share >



jimbob • 5 years ago

the millenium bug was not a bug. in the 1970s and 1980 when disk space was minimal and prohibitively expensive, programmers saved space and money by not including the century as part of the date. Companies postponed adding the century to dates until they were forced to by the approaching new cenury.

2 ^ | v • Reply • Share >

Poisson Volant ➔ jimbob • 5 years ago

Absolutely! The only shortsightedness was on the part of managers who kept dismissing the problem as bogus, and some media trolls who claimed it was a plot by those long-ago developers to make sure they'd have jobs when they were all 75 and 80 years old.

My first task as a high school "programming aide" in the late 1960s was to scour code for unused/unnecessary fields because each wasted byte cost at least 1 penny a year to store. Multiply that by hundreds of thousands of databases with millions of records and it added up to some very serious bucks.

3 ^ | v • Reply • Share >



RS • 5 years ago

The Cold War one was actually caused by a gamma-ray burst from a nearby star (nearby meaning several light years away) picked up by radar, not by the sun. I also heard about this crazy story from "How the Universe Works" on Netflix :P!

2 ^ | v • Reply • Share >

EarthLove • 5 years ago

The picture on #5 is the USS Hopper launching an SM-2 surface to air missile, not a Minuteman ICBM. I enjoyed the list though.

2 ^ | v • Reply • Share >

TeblorTribe • 5 years ago

I've heard about the 2038 problem. I assume that the potential problem will be fixed though long before the date some time in the next 25 years.

Good list.

2 ^ | v • Reply • Share >

Poisson Volant ➔ TeblorTribe • 5 years ago

Hmmm ... I remember hearing something like that around 1985 when people were first becoming aware of Y2K - "won't all that old software be replaced by the turn of the century?" Businesses weren't interested in spending money on what was seen as a non-problem because "the code works OK now, doesn't it?" And besides, none of the then-current decision makers would be around in the far-off year of 1999 :)

2 ^ | v • Reply • Share >

Mike Horton • 7 months ago

ANDREW JONES! You made a Naval faux faux. ..the story is about USS YORKTOWN Tico class cruiser CG-48... Your photo is of the NOW MUSEUM CV-10 Essex class air craft carrier. . USS YORKTOWN decommed in 1975... TURN TO and fix it... now or the bosun and chief of the boat are going the send you to captains mast!

1 ^ | v • Reply • Share >



deadspark • 5 years ago

Ariane 5, anyone?

1 ^ | v • Reply • Share >



SPQR • 5 years ago

The USS Yorktown was struck from the Navy in 1973 so I'm sure it wasn't a serious national security issue when something went wrong in the late 90s. It's a museum ship in Charleston, SC?

1 ^ | v • Reply • Share >



Andrew Jones ➔ SPQR • 5 years ago

The list refers to the USS Yorktown (CG-48) that was in service between 1984 to 2004, it's the USS Yorktown (CV-10) that is a museum ship.

^ | v • Reply • Share >



SPQR ➔ Andrew Jones • 5 years ago

Sorry I was thrown off by the Aircraft Carrier picture.Cheers!

^ | v • Reply • Share >

segue • 5 years ago

Good list. I did not know about most of them, but the North American blackout one was well documented in the news, so I knew that one. Therac-25 was another I had heard of and, of course, Y2K. Most of the rest are new to me, so I learned something today. Re the Y2K bug: I didn't really buy into the whole end of the world as we know it scenario, but thought it prudent

didn't really buy into the whole end-of-the-world-as-we-know-it scenario, but thought it prudent to take some small precaution, just in case. So, 18 months before it was supposedly going to hit, I began saving all \$1 and \$5 bills I received in change when shopping. I kept it in a big bowl in a kitchen cabinet. When the date came and went without incident I counted my money. I had painlessly saved over \$2500! That was nice.

1 ^ | v • Reply • Share >

bigski → segue • 5 years ago

hahaha...

^ | v • Reply • Share >



Gemfyre • 5 years ago

That WoW bug is pretty fascinating from an epidemiological perspective. I wonder if anyone documented it properly.

1 ^ | v • Reply • Share >

Frodo28 → Gemfyre • 5 years ago

Mostly just idiots running around infecting people on purpose, I doubt it would happen like that in real life.

3 ^ | v • Reply • Share >



the mick → Frodo28 • 5 years ago

I agree, it was an accidental 'Zombie-mode' that turned the game into a huge 'tag' fest...

2 ^ | v • Reply • Share >



Jimmy Saville • 3 months ago

The Millennium bug didn't exist like Jesus and the Dinosaurs so I don't like this cuz it be fake if you report me Ill be deported back to Romania

^ | v • Reply • Share >

Hyungnam Gu • 3 months ago

No.6

A report in the Moscow Times quoted KGB veteran Vasily Pchelintsev as saying that there was a natural gas pipeline explosion in 1982, but it was near Tobolsk on a pipeline connecting the Urengoy gas field to the city of Chelyabinsk, and it was caused by poor construction rather than sabotage; according to Pchelintsev's account, no one was killed in the explosion and the damage was repaired within one day. Reed's account has also not been corroborated by intelligence agencies in the United States.

Another point of criticism of the sabotage allegations is that, according to Prof. V.D. Zakhmatov, an explosion safety expert who has overseen the safety measures on many of the Soviet oil and gas pipelines built in the Eighties, at the described timeframe Soviet Union simply didn't practice digital control of its pipeline system. Most of the control was manual, and whatever automation was used utilized the analog control systems, most of which worked through pneumatics.

^ | v • Reply • Share >

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**AceSoHigh** • 5 years ago

i cant believe people take stupid shit like that with extreme seriousness. i thought like when the computers reset it would launch nukes or something serious, not something as retarded as that.

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Hillyard** • 5 years ago

Interesting list.

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Joe** • 5 years ago

"somebody is going to have to consider what we're going to do on the 1st of January 10000"

LISTVERSE

Listverse is a Trademark of Listverse Ltd.

Copyright (c) 2007–2017 Listverse Ltd

All Rights Reserved.

[Privacy](#)