Softwarebugs……ZX ronde 22 mei 2016

In een van eerder verhaaltjes heb ik wel eens wat verteld over de software van computersystemen. Met name over de ontwikkeling van besturingssoftware voor PLC`s door middel van flowcharts als bijvoorbeeld nassi shneiderman diagrammen. Maar ook over het aandeel van software in besturingsystemen.

In het verhaaltje over elektriciteit systemen in vliegtuigen ging het ook over redundant uitgevoerde besturing systemen die elkaar controleren.

In dit verhaaltje aandacht voor de betrouwbaarheid van onze computer met name de betrouwbaarheid van de software!!!.

Wie heeft thuis geen problemen gehad met zijn desktop computer of labtop.

Soorten Software op onze PC

Software of programmatuur zijn andere woorden voor computerprogramma's. Software is de tegenhanger van Hardware en daardoor de niet-tastbare variant.

Het BIOS van een computer (Basic Input-Output System) is ingebouwde (embedded) software op een moederbord. Het is de eerste software die je PC laadt, zodat het randapparaten kan gebruiken zoals CD-stations, de muis en het toetsenbord, zo ongeveer meteen vanaf het moment dat je de computer aanzet.

Door middel van de BIOS kan het systeem een zelftest (POST, Power-On Self Test) doen. Hierbij checkt het systeem de processor, het geheugen, de videokaart en de opslagschijven. Ook bepaalde randapparatuur wordt gecontroleerd op juist functioneren. Ook kunnen er in de BIOS veel functies gewijzigd en/of uitgeschakeld worden.

Het BIOS zelf is meestal een EEPROM (Electronically Erasable Programmable Read-Only Memory), geprogrammeerd met "firmware" en heeft de mogelijkheid om kleine hoeveelheden informatie op te slaan, specifiek voor gebruikersconfiguraties. BIOS componenten zijn vaak op het moederbord gesoldeerd en daardoor niet te verwijderen door de gebruiker. Anderen kunnen worden toegevoegd middels een socket, waardoor die gemakkelijk te vervangen zijn.

Het besturingssysteem van een computer, ook wel systeemsoftware genoemd, is nodig voor het functioneren van het systeem. De hierbij bekende voorbeelden zijn Windows, Linux en Mac OS. Ik neem in dit artikel even

Windows 7 als voorbeeld besturingssysteem.
Het besturingssysteem stuurt, door middel van stuurprogramma's (Drivers), de gehele computer aan. Drivers zijn kleine softwarematige pakketjes die ervoor zorgen dat er een verbinding komt tussen de hardware en het besturingssysteem.
De kern van het besturingssysteem, ook wel de kernel genoemd, implementeert alle diensten die voor het hele systeem beschikbaar zijn, zoals multitasking en geheugenbeheer.
Het besturingssysteem maakt het werken op een computer een stuk makkelijker.

Denk hierbij aan het kopiëren van bestanden, aanmaken van mappen en het installeren en gebruiken van applicatie softwarepakketten, zoals Word, Excel, PowerPoint en foto- en videobewerking en spellen enz.

Firmware:
Firmware is software die in hardware geprogrammeerd is. Zoals de BIOS van het moederbord. Maar ook bijvoorbeeld routers en zelfs televisies. Firmware wordt ook vaak gebruikt als besturingssoftware voor elektronica waar een processor aanwezig is.

Fouten in onze PC

Zodra het mis gaat met opstarten of met het werken met een applicatie programma hebben we de neiging te wijzen naar de instellingen of beperkingen van het Windows besturingsprogramma of naar een virus, malware of wat dan ook. Maar niet direct naar een software fout in een van de programma`s.

Gelukkig kunnen we tegenwoordig bij Windows 7 en 10 een betrouwbaarheid test uitvoeren

Betrouwbaarheidtest.
Windows 7 en 10 .Open het menu Start en typ *Betrouwbaarheid* in het zoekvak.

Hier is de betrouwbaarheid – en probleemgeschiedenis per dag of per week te controleren. Zelf heb ik dat een tijdje bijgehouden en het valt me op hoeveel Windows fouten, toepassingsfouten,overige fouten en waarschuwingen er worden weergegeven.

Bij veel van deze fouten merken we daar niet zo veel van omdat het Windows besturingsprogramma onder water veel van deze fouten herstelt. Alleen de fatale fouten merken we doordat het systeem vastloopt of we een melding krijgen. (*Fouten – Bugs*)

Als we deze fouten in onze huis computer tegen komen zou het dan ook zo zijn dat er in professionele ICT software systemen ook software fouten zitten. Mijn ervaring inmiddels met deze computernetwerken is dat ook zo is. Er wordt door ICT bedrijven heel veel storing verholpen en onderhoud gepleegd aan deze netwerkwerken om ze in bedrijf te houden.

De noodzaak om fouten op te sporen en ze te verhelpen is urgent omdat kwaadwillende op zoek zijn naar fouten in de software. En over de gevolgen hiervan zijn er te veel voorbeelden.

Het is dus noodzaak om software te gebruiken met een hele lage foutkans.

Niet zo zeer voor onze huiscomputers maar zeker voor professionele systemen die door banken , overheid , vliegtuigen, auto`s enz. gebruikt worden.

De vraag die bij me opkomt is, in hoe verre is het mogelijk om foutloze software maken. En hoe is deze software te testen en te borgen.

De vraag is vrij simpel maar het antwoordt wat lastiger.

Softwarebedrijven zoals bijvoorbeeld microsoft geven heel veel geld uit voor het controleren en testen van de nieuw op de markt uit te brengen software maar ook voor de update van bestaande software pakketten.

Verder kunnen we debugger programma`s gebruiken om fouten op te sporen en te herstellen.

Maar soms gaat het om bugs die met de huidige middelen erg moeilijk te vinden zijn. Een belangrijk type softwarefouten zijn de zogenaamde Heisenbugs , genoemd naar het onzekerheidsprincipe van de Duitse natuurkundige Heisenberg, omdat deze soms wel, soms niet optreden.

Bijvoorbeeld een bug die met een timingprobleem te maken heeft. Wanneer de programmeur deze bug probeert op te sporen in een debugger verandert de timing en lijkt de bug te verdwijnen. Door het ongrijpbare karakter kan het erg veel tijd kosten voor zo'n bug gereproduceerd kan worden. Vervolgens kan het nog veel langer duren om de uiteindelijke oorzaak te vinden.

Mede hierdoor lukt het niet een software pakket aan te bieden wat 100% foutloos is. De vraag is of dit überhaupt te realiseren is.

Foutloze software bestaat kortom niet en zeker niet in grotere en complexe omgevingen. Het aantal fouten dat in grote systemen kan sluipen, lijkt wel eindeloos. Achteraf testen kan veel problemen oplossen, maar leidt slechts tot foutarme software.

In een artikel hierover in het blad Computable schreef Professor Ed Brinksma van de Universiteit Twente dat hij deze problemen wijt aan het gebrek aan professionaliteit in de IT-branche.

Nodig zijn een wetenschappelijke theorie, mathematische modellen, bewezen ontwerpoplossingen en rigoureuze kwaliteitscontrole." De software-branche kenmerkt zich door een gebrek aan vrijwel al deze factoren.

"We beschikken niet over een standaard professionele opleiding, er bestaan geen standaard productieprocessen, hetgeen zich uit in slecht voorspelbare resultaten en kwaliteit.

Hij vindt dat de branche afhankelijk is van een beperkt aantal begaafde 'ambachtslieden. Hij pleit dan ook voor verdere professionalisering, temeer omdat investeringen op dit gebied behoorlijk lonend kunnen zijn.

Tja als je dit soort artikelen lees dan wordt het vertrouwen in foutloze software er niet groter op.

Maar we moeten het zeker voor thuis gebruik niet overdrijven,wie bijvoorbeeld met Windows 7 werkt en legale applicatie sofware gebruikt zal buiten de onderlinge software conflicten niet zoveel merken van software fouten. Updates zorgen voor aangepaste software waar weer de nodige fouten uit gehaald zijn.

We moeten ook niet vergeten dat we vaak zelf veroorzakers zijn van computer ongemak. Dit bijvoorbeeld door het downloaden van software van onbekende herkomst, of door onvoldoende bescherming tegen criminele software als virussen , malware enz.

Ook kan het aansluiten van hardware met eigen interne firmware kan tot nodige conflicten leiden doordat bijvoorbeeld de interne drivers de aangesloten firmware niet herkend.

De meeste computerstoringen worden veroorzaakt door de software en niet door de hardware. Over het algemeen maakt een computer geen fouten het is het menselijk falen die ten grondslag ligt aan de storing.

Het systeem is afhankelijk van goede inputinstructies wat moet leiden naar het gewenste resultaat. Plat gezegd shit leidt tot shit out.

Daar ben ik zelf ook achter gekomen met het maken van selectiviteit grafieken uit berekening in Excel. Ook hier geven fouten in de berekening niet het gewenste resultaat in de grafiek en daar kan de computer niets aan doen.

Oorzaken dat bugs in software blijven bestaan

1. **Menselijke natuur**

De meeste - maar niet alle - bugs ontstaan door menselijke fouten. Sommige kunnen we ze wijten aan onverwachte of rare gevolgen van een programmeertool of compiler, maar de meerderheid komt voort uit fouten van een ontwikkelaar. Het maakt niet uit hoe goed onze SDL-training is of welke beveiligingstools we hanteren, we blijven mensen en die maken fouten. Als je wilt weten waarom software fouten bevat: dat komt omdat mensen feilbaar zijn.

2. **Toegenomen complexiteit**

Software wordt logischerwijs steeds complexer en dat betekent meer regels code. Met programmeren is het zo dat ongeacht hoe goed je bent in je werk er een aantal bugs - niet altijd te benutten voor exploits - per x aantal regels code ontstaan. Mensen die dat soort foutjes tellen zeggen dat als je één bug per 50 regels hebt je het nog best goed gedaan hebt. De meeste programmeurs zitten dichter in de buurt van een fout per 15 regels. Bedenk dat programmatuur een heleboel regels heeft, bijvoorbeeld de Linux-kernel met 15 miljoen, en je kunt wel uitrekenen hoeveel bugs dat oplevert.

3. **Fuzzers weten niets meer dan wij**

Vandaag de dag zetten we fuzzers in om kwetsbaarheden bloot te leggen. Fuzzing-tools - programma's die naar codefouten en kwetsbaarheden zoeken - zijn geschreven door mensen. Fuzzers vonden die oorspronkelijke buffer overflow-fout niet omdat ze niet waren ontworpen om daar rekening mee te houden. Na het succes van deze aanval, werden de tools bijgewerkt, en nu zoeken ze op allerlei gebieden op dat soort buffer overflow-problemen. Kortom, ze ontdekken alleen maar de dingen waar we ze van op de hoogte stellen

4. **Leveranciers zijn niet verantwoordelijk te houden**

Veel beveiligingsexperts stellen dat we nooit beter beveiligd zullen worden zolang bedrijven niet aangeklaagd kunnen worden voor softwarefouten. Ik ben het ermee eens dat we bedrijven aan hun verantwoordelijkheid moeten kunnen houden, maar nog meer juridische aansprakelijkheid zou de vooruitgang alleen maar beperken. Je zou niet die coole smartphone, superlichte mp3-speler of films op internet kunnen streamen als softwaremakers nóg vaker in de rechtszaal stonden.

5. **Hackers zijn niet verantwoordelijk te houden**

De waarheid is helaas dat we bovenstaande dingen niet eenvoudig kunnen oplossen. Maar eigenlijk zijn de softwarefouten het issue niet. Het echte probleem is de exploit die de kwetsbaarheid uitbuit voor sinistere doeleinden. Zolang criminele hackers ermee wegkomen dat ze in het wilde weg gaten slaan en malware pushen, blijven we deze problemen houden.