CrowdStrike-gate

**Hoe de halve wereld opeens platlag,
hoe dit voorkomen had kunnen worden
en hoe de toekomst eruitziet**

Scopisto

Voorwoord_

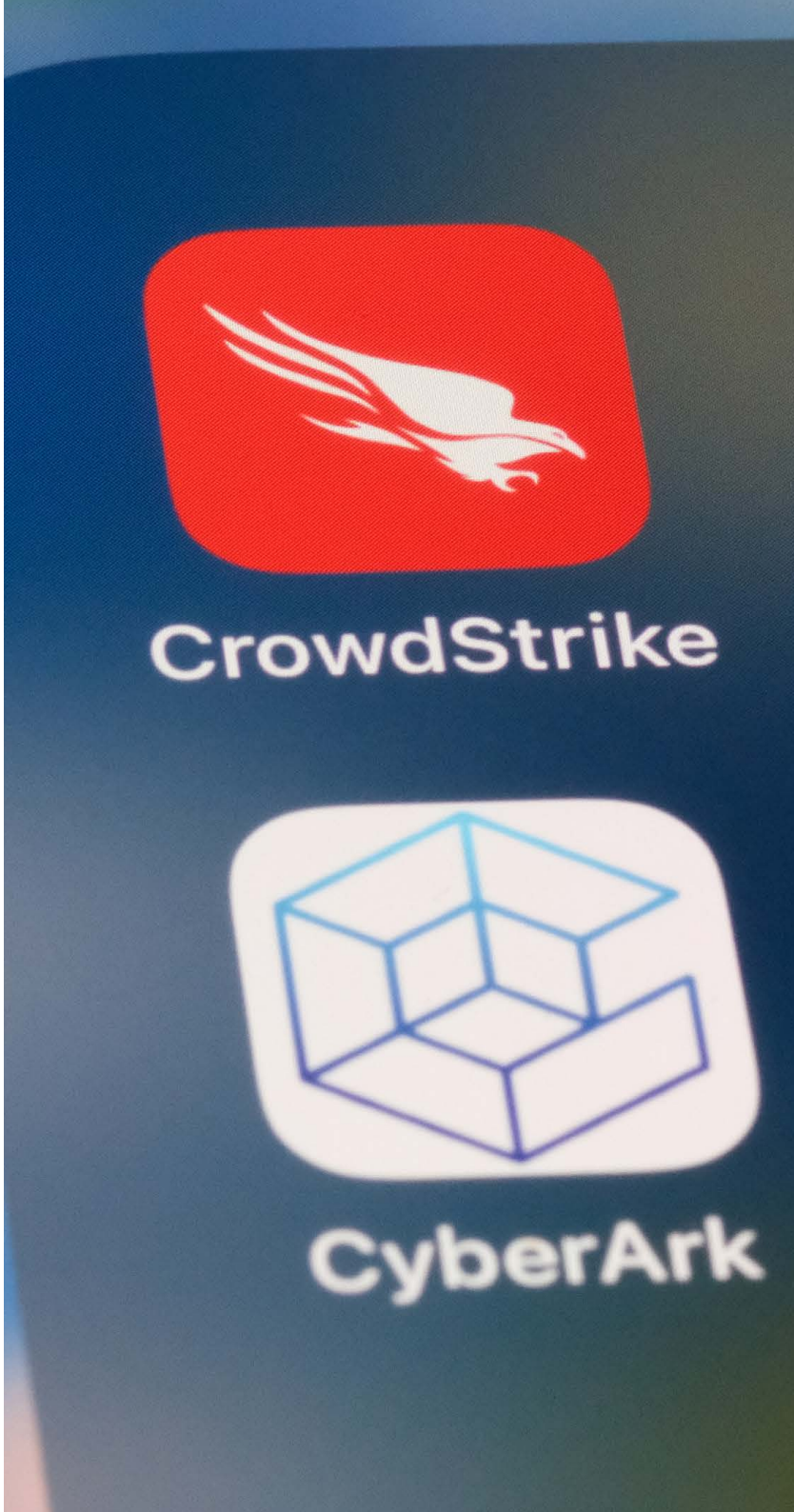
Op 19 juli zorgde een update van de Falcon-applicatie van leverancier CrowdStrike voor een wereldwijde storing van in totaal 8,5 miljoen Windows-pc's. Hierdoor werd kritieke infrastructuur lange tijd onbruikbaar. Vluchten werden geannuleerd, operaties in ziekenhuizen konden niet doorgaan, banken werden geraakt.

Wat is er nou precies gebeurd? En kunnen dit soort problemen in de toekomst voorkomen worden? In dit e-book leg ik dit in Jip & Janneke-taal uit. Tenminste, dat probeer ik. :)

Michiel Oliemans
Scopisto

Disclaimer: mijn analyse is gebaseerd op ruim 25 jaar ervaring als software-ontwikkelaar en de best practices die komen kijken bij de ontwikkeling en het beheer van dit soort platformen. Ik heb echter nooit een antivirusplatform ontwikkeld en ben ook niet bekend met de code van het Falcon-platform.

Inhoud_



Cheat sheet_

Falcon is beveiligingssoftware bedoeld om te voorkomen dat virussen en hackers toegang tot je computer kunnen krijgen. CrowdStrike heeft in een blog uiteengezet wat de precieze oorzaak van de crash is geweest. Om dit goed uit te kunnen leggen, is het handig om te weten hoe de architectuur van dit soort applicaties in elkaar steekt.

Architectuur
Voor dit artikel zijn de belangrijkste onderdelen van het platform:

1. De **Falcon-sensor** is een applicatie die op de lokale machine (de Windows-pc) draait;

2. Deze applicatie is verbonden met **een webapplicatie** die draait in de cloud;

3. **Channel files:** kleine bestandjes die meerdere keren per dag vanaf de webapplicatie naar de Falcon-applicatie worden verzonden.

Achtergrond_

Kwetsbaarheden en patches

Er worden continu nieuwe kwetsbaarheden gevonden in applicaties en systemen. Dit zijn de achterdeurtjes waardoor bijvoorbeeld hackers naar binnen kunnen glippen, met alle gevolgen van dien. Gelukkig worden er ook continu oplossingen voor deze kwetsbaarheden ontwikkeld, zogeheten ‘patches’. Zie het als de sleutels om zo’n achterdeurtje weer stevig op slot te draaien.

Om je computer zo veilig mogelijk te houden, is het belangrijk dat zo’n patch zo snel mogelijk bij de kwetsbaarheid terechtkomt. Op die manier minimaliseer je namelijk de tijd dat een hacker er misbruik van kan maken en daarmee dus de kans dat jij wordt geraakt. Beveiligingssoftware stuurt deze patches naar jouw computer in de vorm van losse bestandjes.

In de betreffende situatie

In het geval van Falcon worden die bestandjes ‘Channel Files’ genoemd. De patches in deze vorm noemen ze Rapid Response Content.

De Falcon-applicatie leest die externe bestandjes in en gebruikt ze dus om je computer veilig te houden en de achterdeurtjes weer op slot te draaien. Om dit te kunnen doen heeft de applicatie wel veel rechten nodig op je computer.

Daarom draait deze applicatie niet zoals bijvoorbeeld Word, maar wordt het uitgevoerd in een diepe laag van Windows. Het nadeel daarvan is dat als er een ernstige fout optreedt in die applicatie, de hele computer onbruikbaar kan worden. En dit is precies wat we zagen op 19 juli met het “blue screen of death”.

Oorzaak

De crash van 19 juli werd veroorzaakt door zo'n Channel File. CrowdStrike zegt hierover:

> *"The configuration update triggered a logic error that resulted in an operating system crash. "*

Dat betekent dat zodra de Falcon-applicatie deze corrupte Channel File probeerde te gebruiken er een fout optrad (= bug), die zo ernstig was dat de hele applicatie crashte. En omdat Falcon in die diepe laag van Windows draait, crashte ook de hele computer.

Net als het beveiligen van computersystemen is het stabiel maken en houden van applicaties een kwestie van het implementeren van meerdere controlelagen. NU.nl kopte dat "de wereldwijde computerstoring werd veroorzaakt door een foutje in testsoftware". Dat zou echter maar één van de benodigde controlelaagjes moeten zijn, waarna het probleem afgevangen had moeten worden in een van de andere controlelagen.

Oplossing CrowdStrike_

Op 20 juli gaf CrowdStrike aan dat ze het probleem hadden opgelost:

"CrowdStrike has corrected the logic error by updating the content in Channel File 291."

Dit zou ik zelf niet een oplossing noemen, maar een workaround. Het is immers de Falcon-sensor die de fout in de logica bevat die getriggerd kan worden door een corrupte Channel File.

Als er in de toekomst nogmaals een corrupte patch wordt verzonden kan dezelfde fout en daarmee dezelfde crash weer plaatsvinden.

Op 24 juli volgt een uitgebreidere analyse en beschrijft CrowdStrike meerdere wijzigingen in het proces die het gaat doorvoeren om dit in de toekomst te voorkomen:

1. Rapid Response Content testen verbeteren

Beschreven oplossing door CrowdStrike

Improve Rapid Response Content testing by using testing types such as:

- Local developer testing
- Content update and rollback testing
- Stress testing, fuzzing and fault injection
- Stability testing
- Content interface testing

Mijn review_

Lokale testen door ontwikkelaar
Dit klinkt nogal vrijblijvend. Elke ontwikkelaar doet standaard wel iets aan lokale tests om wijzigingen te verifiëren. Het zou veel beter zijn als ze hun development pipeline aanpassen zodat elke code-wijzigingen geanalyseerd wordt door een statische code-analysetool zoals SonarQube.

Deze tool kan veel problemen vinden in de code-wijzigingen, zoals out-of-bounds memory reads - wat deze crash veroorzaakte. Daarnaast kan die tool het ontwikkelteam dwingen om ervoor te zorgen dat de nieuwe code op z'n minst aan bepaalde basiseisen rondom testing moet voldoen, oftewel zogeheten minimale test coverage moet hebben. Dat betekent dat alle nieuwe code automatische testen moeten bevatten waarmee de correctheid van die code gevalideerd wordt.

Bestaan die testen niet of zijn er potentiële bugs of andere issues in de code gevonden? Dan is het simpelweg niet mogelijk dat die code doorgaat naar de volgende laag. Op deze manier borg je kwaliteit in het proces in plaats van dat je zegt “de ontwikkelaar test het lokaal wel”.

Overigens lees ik niks over het uitvoeren van code reviews door andere ontwikkelaars, maar dát zullen ze wel doen, toch...?

Overige typen testen
CrowdStrike noemt nog de volgende testtypen die ze gaan toevoegen:

- Content update and rollback testing
- Stress testing, fuzzing and fault injection
- Stability testing
- Content interface testing

Dit valt eigenlijk allemaal onder tests die onder de verantwoordelijkheid van een QA-team worden uitgevoerd. Het blijft een beetje in het midden of ze het al deden, maar in ieder geval blijkt dat de testcases die nu gehanteerd werden onvoldoende coverage hadden.

Ook blijft onduidelijk of het hier gaat om handmatige of geautomatiseerde testen. Die laatste hebben altijd de voorkeur. Handmatig testen is niet alleen een stuk tijdrovender, maar het is ook foutgevoeliger omdat het mensenwerk blijft. Als het geautomatiseerd is kan het als volgende stap in de pipeline geplaatst worden als volgende laag in het proces om te voorkomen dat fouten naar productie worden gedeployed.

2. Content Validator voor Rapid Response Content verbeteren

Beschreven oplossing door CrowdStrike

Add additional validation checks to the Content Validator for Rapid Response Content. A new check is in process to guard against this type of problematic content from being deployed in the future.

Mijn review_

De Rapid Response Content updates worden niet in code geschreven, maar worden gegenereerd in het Falcon-platform in de cloud. Die applicatie bevat een module waarin nieuwe Rapid Response Content kan worden aangemaakt en geconfigureerd. Daarna gaat die door een validator ter controle voordat hij naar de sensoren wordt gestuurd. Deze validator ging deze keer de mist in en wist het probleem niet te herkennen in de update die bij de Falcon-sensor voor problemen zorgde.

CrowdStrike is bezig om deze validator te verbeteren zodat hij deze nu bekende fout voortaan wel kan herkennen en blokkeren. Maar is dat wel voldoende? Want op deze manier kun je alleen valideren wat je van tevoren hebt bedacht of al eens eerder hebt meegemaakt. Kortom: we weten niet wat we niet weten en wat dus eventueel in de toekomst weer voor andere problemen kan zorgen.

Klopt het hele idee van deze validator dan wel? Wat nou als je de validator vervangt of uitbreidt met een handeling die de Falcon-sensor ook zal uitvoeren. Oftewel: stuur ter validatie die update ook naar een speciale installatie van een Falcon-sensor waarvan je dan terug kunt krijgen of die update succesvol is ingeladen of niet. Op die manier ga je van een inschatting of de update goed is naar een daadwerkelijke validatie.

3. Foutafhandeling in Falcon-sensor verbeteren

Beschreven oplossing door CrowdStrike

Enhance existing error handling in the Content Interpreter.

Mijn review_

Een applicatie die externe bestanden inleest moet er rekening mee houden dat die bestanden corrupt zijn of door een kwaadwillende (= een hacker) aangepast zijn om bijvoorbeeld bepaalde checks uit te schakelen. De Falcon-sensor moet daarom elke fout die op kan treden bij het inlezen van die bestanden netjes kunnen afhandelen zonder dat de hele applicatie crasht.

CrowdStrike geeft aan dat zij in dit scenario op dit moment wel bepaalde fouten afhandelen, maar niet deze onverwachte fout. Zonder de code te hebben gezien, klinkt dat alsof ze alleen rekening hielden met fouten die ze zelf van tevoren hadden bedacht, maar niet een zogenaamde “catch all” hadden toegevoegd om echt onverwachte systeemfouten op te vangen.

Een applicatie die door een ernstige fout zorgt voor een ‘blue screen of death’ had dat wel moeten doen. CrowdStrike gaat nu deze foutafhandeling alsnog toevoegen. Dit is de bug in het platform die de blauwe schermen heeft veroorzaakt.

4. Canary Deployment en Monitoring

Beschreven oplossing door CrowdStrike

- Implement a staggered deployment strategy for Rapid Response Content in which updates are gradually deployed to larger portions of the sensor base, starting with a canary deployment.
- Improve monitoring for both sensor and system performance, collecting feedback during Rapid Response Content deployment to guide a phased rollout.

Mijn review_

Het ging wereldwijd mis op allerlei kritieke omgevingen. Het is heel bijzonder dat ze überhaupt nog geen strategie hadden om dit soort updates gefaseerd uit te kunnen rollen en ze het blijkbaar ook niet goed kunnen monitoren. Op het moment dat je op zoveel verschillende systemen draait en zoveel verschillende klanten hebt, dan had dit allang ingericht moeten zijn. Je verwacht het misschien ook gewoon van dit soort grote bedrijven, maar dit bewijst maar dat dat niks zegt over de daadwerkelijk geleverde kwaliteit en controle.

5. Controle updates door klanten

Beschreven oplossing door CrowdStrike

Provide customers with greater control over the delivery of Rapid Response Content updates by allowing granular selection of when and where these updates are deployed.

Mijn review_

Blijkbaar hebben klanten op dit moment geen controle over hoe een update wordt doorgevoerd in hun eigen netwerk. Schiphol bijvoorbeeld kan nu dus niet zelf testen uitvoeren voordat ze updates doorlaten naar al hun computers. Dat is opvallend en waarschijnlijk een ontwikkeling die door de opkomst van SaaS-applicaties is ontstaan.

Toen ik jaren geleden een applicatie voor een grote Nederlandse bank had ontwikkeld, konden daar echt niet zomaar updates van doorgestuurd worden. Deze werden streng gecontroleerd en getest door hun eigen IT-afdeling voordat deze verder door het netwerk werden gestuurd. Maar dit was nog vóór het cloud-tijdperk.

Tegenwoordig draait alles in de cloud en zijn blijkbaar dit soort controles versoepeld. Nu blijkt maar weer dat je als klant toch wel je eigen verantwoordelijkheid moet blijven nemen. Je ziet de consequenties nu ze dat hier massaal hebben nagelaten. Blijkbaar bood dit systeem daar geen mogelijkheden voor, maar dan dwing je dat als klant toch af of je kiest een andere leverancier?

Nu gaat CrowdStrike hier mogelijkheden voor implementeren, wat heel goed is. Nu maar hopen dat de klanten zelf ook zo slim zijn om dit ook echt te gaan gebruiken.

6. Release notes van content updates

Beschreven oplossing door CrowdStrike

Provide content update details via release notes, which customers can subscribe to.

Mijn review_

Leuk, maar ik denk niet dat iemand deze ooit gaat lezen.

Vraagtekens_

Waarom?

CrowdStrike lijkt met de aangekondigde wijzigingen in het proces nu de juiste richting op te gaan. Het zijn hele logische maatregelen waarvan je je eigenlijk af mag vragen waarom deze niet überhaupt allang werden toegepast? Waarom moet het eerst zo gigantisch misgaan voordat zo'n groot bedrijf zelf tot de conclusie komt om betere processen in te richten?

Voldoende maatregelen om bestanden te valideren

CrowdStrike geeft aan dat die Channel Files een eigen (geheim) bestandsformaat gebruiken. Gezien de problemen en de ontbrekende controles die tot deze ellende hebben kunnen leiden, kan ik niet helpen dan me afvragen of ze wel voldoende maatregelen hebben genomen om die externe bestanden te downloaden en te valideren voordat ze überhaupt proberen ze in te lezen?

Communicatie tussen Falcon-sensor en cloud-applicatie

Hoe veilig is de communicatie tussen de Falcon-sensor en de cloud-applicatie waar hij zijn updates van krijgt? Hoe weet de cloud-applicatie dat het een geldige Falcon-sensor is die verbinding met hem maakt? Is de data die de twee applicaties met elkaar uitwisselen versleuteld zodat anderen die niet zomaar kunnen afluisteren of een man-in-the-middle attack kunnen uitvoeren?

Hoe verifieert de Falcon-sensor of de gedownloade Channel File wel echt geldig is? Voert hij bijvoorbeeld een fingerprintcheck uit door te controleren of de digitale vingerafdruk van het gedownloade bestand hetzelfde is als verwacht? Het bestand heeft dan misschien wel een eigen bestandsformaat, maar is dat bestand zelf dan nog eens versleuteld zodat ook daar weer een extra beveiligingslaag omheen zit?

Conclusie_

De reputatie van CrowdStrike heeft in één keer een flinke deuk opgelopen. Ze geven nu een aantal goede stappen aan, maar eerlijk is eerlijk: dit had echt allang zo moeten werken. Het is een bedrijf met 8.000 medewerkers, maar deze processen waren gewoon slecht of niet georganiseerd. Hoeveel van die mensen zijn er eigenlijk verantwoordelijk voor de ontwikkeling en beheer van het product (dus de ontwikkelaars, QA, DevOps, Product Owners, etc) en hoeveel in andere functies, zoals bv account managers, marketing, sales?

En ook nu in de aangegeven oplossingen krijg ik niet het vertrouwen dat het vanaf nu wel goed zal gaan. Het leest voor mij als te vrijblijvend. Er worden geen systemen gebruikt om ook echt updates te blokkeren die waarschijnlijk voor problemen gaan zorgen.

Als klant ben je misschien geneigd te denken dat omdat een bedrijf (heel) groot is, veel grote klanten heeft en misschien zelfs beursgenoteerd is, dat die daarom heel goed weet hoe je software moet ontwikkelen en stabiel kunt houden. Deze wereldwijde storing en de opsomming van verbeterpunten die CrowdStrike zelf heeft gecommuniceerd tonen aan dat dat helemaal niet het geval is.

Belangrijkste take-aways_

1. Positief hoe open en (technisch) gedetailleerd CrowdStrike hierover communiceert

2. Automatische testen en gefaseerde uitrol werd wel al toegepast op de sensor updates zelf, maar niet op de Rapid Response Content

3. Vertrouwen in de volledigheid en correctheid van de Rapid Response Content Validator. Klopt deze oplossing überhaupt wel? Zou als onderdeel van validatie ook niet een echte sensor gebruikt moeten worden?

4. Alle bestaande of nieuwe testen van de Rapid Response Content lijken menselijk/handmatig te zijn. Dat opent de deur voor aannames, zoals ook in dit geval was gedaan, door niet nogmaals te (stress)testen omdat het vorige keer ook goed ging. Beter om het juiste proces te definiëren en dat in een systeem met tools af te dwingen, zodat je gedwongen wordt om de juiste stappen te doorlopen.

5. Sensor lokaal onvoldoende beschermd tegen onverwachte fouten, terwijl deze

    - tot een volledige systeemcrash van Windows kan leiden;
    - externe (en daarmee potentieel onveilige) bestanden inleest.

6. Ik mis een statische code analyse tool zoals SonarQube om dit soort problemen al tijdens ontwikkeling te vinden (is mijns inziens een van de tools van punt 3).

7. De grootte van een bedrijf zegt niets over de kwaliteit van de softwareontwikkeling en de stabiliteit en onderhoud.

8. Klanten namen zelf geen eigen verantwoordelijkheid door een gefaseerde uitrol af te dwingen en zelf te testen.

9. Te weinig detail om het zeker te kunnen stellen, maar zoals het nu beschreven is lijkt het erop dat de Rapid Response Content bestanden geen extra controles of encryptie bevatten om te verifiëren dat ze veilig zijn. Het klinkt alsof dit simpelweg geserialiseerde data-objecten zijn, die door de sensor gedeserialiseerd worden om vervolgens te gebruiken.

Over de auteur_

Michiel Oliemans heeft meer dan 25 jaar ervaring in software development. In 2013 richtte hij Scopisto op, dat inmiddels bestaat uit een team van bijna 25 dedicated developers, product owners en specialisten in QA, DevOps en UX in Nederland en Noord-Macedonië.

Over Scopisto

Al ruim 10 jaar leveren we kwalitatieve én extreem veilige software. Dit doen we met een toekomstbestendige software-architectuur en clean code. Zo ontwikkelen we software die makkelijk te wijzigen en te updaten is, wel zo duurzaam. Natuurlijk zorgen we er ook voor dat het beheer en de beveiliging staan als een huis. Afgeraffelde, mix-and-match software? Not on our watch!

Bronnen_

- https://www.crowdstrike.com/blog/falcon-update-for-windows-hosts-technical-details/
- https://www.crowdstrike.com/falcon-content-update-remediation-and-guidance-hub/
- https://www.nu.nl/tech/6321865/wereldwijde-computerstoring-werd-veroorzaakt-door-foutje-in-testsoftware.html
- https://www.zdnet.com/article/what-caused-the-great-crowdstrike-windows-meltdown-of-2024-history-has-the-answer/

Scopisto