4 ANALYSE SYSTEEM

Hoofdstuk 3 analyseerde het voorval als gevolg van een kwetsbaarheid in Citrix-software. Dit voorval stond niet op zichzelf. Het hoofdstuk analyseerde tevens vergelijkbare voorvallen waarbij kwetsbaarheden in software leidden tot beveiligingslekken bij organisaties. In sommige gevallen had dit directe gevolgen voor de veiligheid van mensen. Dit illustreert dat kwetsbaarheden in software niet op zichzelf staan. Het zijn symptomen van een groter probleem. Deze voorvallen laten namelijk een rode draad zien: organisaties en de mensen die daarvan afhankelijk zijn worden blootgesteld aan digitale onveiligheid. Zij gebruiken namelijk onbewust kwetsbare software. Daarbij bereiken waarschuwingen hen in veel gevallen niet en hebben organisaties niet altijd of hebben organisaties niet de middelen om de kwetsbaarheid te kunnen verhelpen.

Dit hoofdstuk analyseert het vraagstuk op systeemniveau. Daarbij wordt onderscheid gemaakt tussen het proces waarin software tot stand komt; het proces waarin organisaties bepaalde software selecteren om aan te schaffen en in gebruik te nemen; en de processen die plaatsvinden zodra er een kwetsbaarheid in de software is aangetroffen (incidentbestrijding). In aanvulling daarop wordt ingegaan op hoe betrokken partijen, zoals fabrikanten, organisaties die software gebruiken en de overheid als beleidsmaker van digitale voorvallen leren. Ook gaan we in op de rol die de internationale context speelt in de beheersing van de onveiligheid als gevolg van kwetsbaarheden in software.

4.1 Software produceren en op de markt brengen

Software vervult een cruciale rol in het functioneren van digitale systemen van organisaties. Software wordt bijvoorbeeld gebruikt voor toegang tot het bedrijfsnetwerk vanuit huis en vormt daarmee ook de koppeling tussen het interne en externe netwerk (internet). Dergelijke producten spelen daardoor ook een essentiële rol bij het waarborgen van de digitale veiligheid.

Softwareproducten dragen altijd inherente kwetsbaarheden met zich mee, waarvan sommige leiden tot grote veiligheidsrisico's. Deze risico's zijn reëel en hebben zich al diverse keren voorgedaan, met ontwrichtende gevolgen voor de publieke dienstverlening. Zo heeft een kwetsbaarheid in een softwareproduct (indirect) geleid tot ernstige verstoringen in de dienstverlening van bijvoorbeeld Nederlandse gemeenten (toeslagen konden niet meer worden uitbetaald aan inwoners) en een ziekenhuis (patiënten konden niet meer bij hun dossiers en er kon geen informatie worden uitgewisseld met andere ziekenhuizen). Hoofdstuk 3 beschreef voorbeelden van kwetsbaarheden in zulke producten en de gevolgen die deze hadden. In deze paragraaf gaan we in op hoe het kan dat software kwetsbaarheden bevat en hoe fabrikanten het risico op deze kwetsbaarheden en de gevolgen ervan inschatten en maatregelen nemen om deze te voorkomen of beperken.

Paragraaf 4.1.1 beschrijft de factoren die verklaren dat kwetsbaarheden in software kunnen ontstaan en de prikkels die daarop van invloed zijn. Vervolgens wordt in 4.1.2 geschetst welke maatregelen fabrikanten treffen om kwetsbaarheden te ontdekken, zowel vóór als nadat software op de markt komt, welke moeilijkheden dat met zich meebrengt en welke dilemma's daarbij ontstaan voor de fabrikant. Tot slot staan we in 4.1.3 stil bij het repareren van kwetsbaarheden en de rol van de fabrikant bij incidentbestrijding.

4.1.1 Voorkomen dat kwetsbaarheden ontstaan in de levenscyclus van software

Kwetsbaarheden kunnen ontstaan gedurende de hele levenscyclus van een softwareproduct. Zo kan een kwetsbaarheid ontstaan tijdens de initiële ontwikkeling van een nieuw product, maar ook bij het vernieuwen of verbeteren van bestaande software, door bijvoorbeeld een *upgrade*, of zelfs als gevolg van het repareren van een andere kwetsbaarheid. Uit interviews met fabrikanten en literatuurstudie blijkt dat een aantal factoren bijdragen aan het ontstaan van kwetsbaarheden tijdens de levenscyclus van een product. We gaan hieronder op een aantal factoren in.

Softwareproducten hebben een geschiedenis

De eerste factor betreft de ontstaansgeschiedenis, die soms lang en ingewikkeld is bij softwareproducten. In het verleden hebben fabrikanten meermaals functionaliteit aan een bestaande softwarepakket toegevoegd en zo doorgebouwd op een bestaand product. In sommige gevallen is de oorspronkelijke code van het softwarepakket (het fundament) zelfs meer dan twintig jaar oud. Door veranderende behoeften en een samenleving die steeds verder en sneller digitaliseert gaat software een andere rol vervullen. Daardoor is een softwareproduct nooit af. Fabrikanten spelen hier steeds weer op in door bestaande platforms te gebruiken en extra functionaliteit toe te voegen of bestaande componenten te hergebruiken.

Doordat fabrikanten herhaaldelijk extra functionaliteit toevoegen groeit het aantal regels code en wordt software complexer.[109] Niet zelden bestaat een softwareproduct bijvoorbeeld uit meer dan één miljoen regels code.[110] Uit interviews en literatuur blijkt dat, zelfs met een uitgebreid raamwerk voor productontwikkeling, het veilig onderhouden van dergelijke hoeveelheden code een significante taak is. Fabrikanten beperken zich daarom meestal tot het verhelpen van de specifieke kwetsbaarheid zoals gepubliceerd in de CVE[111]. Het verhelpen van de achterliggende oorzaak in het fundament van het product (programmeertaal, componenten, architectuur) kan de volledige heropbouw van het product vergen. Fabrikanten vinden dit te kostbaar. Grote softwarebedrijven zijn vaak beursgenoteerde bedrijven en financiële afwegingen spelen een rol. Maar door de ontstaansgeschiedenis van software is een product soms zo gegroeid dat het repareren van een kwetsbaarheid slechts symptoombestrijding is. Dit terwijl een herziening van de basis van het product nodig kan zijn om het echte (veiligheids)probleem op te lossen.

109 https://www.extremetech.com/computing/259977-software-increasingly-complex-thats-dangerous.
110 https://www.informationisbeautiful.net/visualizations/million-lines-of-code/.
111 Common Vulnerabilities and Exposures. Een openbare lijst van bekende en zwakke plekken in software. De lijst staat op https://cve.mitre.org. (Bron: Cybersecurity Alliantie, *Cybersecurity Woordenboek* (2019), https://www.cybersecurityalliantie.nl/binaries/cybersecurityalliantie/documenten/publicaties/2019/09/30/cybersecurity-woordenboek/VCNL-Woordenboek-2eDruk-webversie-Final-2.pdf).

Programmeertaal
De gebruikte programmeertaal kan, als tweede verklarende factor, ook van invloed zijn op het ontstaan van kwetsbaarheden in software. De op dit moment meest gebruikte programmeertaal (C/C++) staat bekend als 'onveilig', omdat het programmeurs veel ruimte laat om fouten te maken.[112]

Er zijn enkele algemene hulpmiddelen die fabrikanten kunnen gebruiken om hele klassen van kwetsbaarheden te elimineren of de werking daarvan te mitigeren. Ongeveer de helft van de beveiligingslekken van de afgelopen jaren betrof bijvoorbeeld kwetsbaarheden in de geheugenveiligheid, die kunnen worden verholpen door code te schrijven in veiligere talen zoals Rust, of door bestaande C/C++ code te onderwerpen aan verificatietools.[113]

Volgens onderzoek is het voor fabrikanten niet aantrekkelijk om de softwareontwikkeling te beveiligen tegen kwetsbaarheden: de software zelf wordt er langzaam van en programmeurs krijgen tijdens het programmeren zo veel foutmeldingen (ook terechte) dat ze de beveiliging uitzetten.[114]

Het is ook niet mogelijk om bij alle programmeertalen tools te gebruiken om kwetsbaarheden op te sporen tijdens het ontwikkelproces.[115] In de Citrix casus speelde bijvoorbeeld een rol dat programmeertaal Perl niet of nauwelijks werd ondersteund door deze *scanning tools*. Zie verder 4.1.2 over wat fabrikanten doen om kwetsbaarheden te ontdekken en welke belemmeringen daarbij spelen.

Gebruik van standaardcomponenten
De derde factor is het gebruik van standaardcomponenten. Fabrikanten maken bij het ontwikkelen van software veelvuldig gebruik van bestaande (open source) software-componenten. Voorbeelden hiervan zijn de Apache en NGINX HTTP server, die vaak dienen als de basis van software met webfunctionaliteit. Ook kan een fabrikant componenten uit reeds bestaande software van henzelf of van overgenomen fabrikanten hergebruiken.

Met het overnemen van andere componenten en de code die daartoe behoort, neemt een fabrikant ook alle (onontdekte) kwetsbaarheden over die de code bevat.[116] Als de

112 De basis voor de SSL VPN (een virtual private network die gebruik maakt van het SSL of TLS protocol) en veel andere software is C/C++. Programmeertalen zoals C stellen programmeurs in staat om code te schrijven op een hoger abstractieniveau. Dit geeft aan hoe dicht een programmeertaal bij de hardware staat. Met een hoger abstractieniveau wordt software ontwikkelen eenvoudiger en begrijpelijker dan op een lager niveau, waarbij specifiekere machine-instructies nodig zijn. Maar dat kan ook leiden tot fouten. C is een programmeertaal die bekend staat als 'onveilig', omdat het werkgeheugenbeheer in deze taal handmatig plaatsvindt (Kroes, T. How to Keep Your Memory Safe and Your Software Fast, 2020; AG Connect, Einde van de oneindige reeks softwarefouten in zicht, 2021). Dit is foutgevoelig en bovendien gebruiken de meeste SSL VPN's eigen toevoegingen aan bestaande programmeertalen. Daardoor kunnen eenvoudig geheugenfouten ontstaan; de meest voorkomende bron van softwarebugs en een belangrijk aanvalsoppervlak voor hackers (zie https://www.zdnet.com/article/microsoft-70-percent-of-all-security-bugs-are-memory-safety-issues/). Desondanks is C nog steeds één van de meest gebruikte programmeertalen.
113 Anderson, R., *Security Engineering*, 2020.
114 Kroes, T., *How to Keep Your Memory Safe and Your Software Fast*, 2020 https://research.vu.nl/en/publications/how-to-keep-your-memory-safe-and-your-software-fast
115 Tjong Tjin Tai, E. en Knoops, B., Zorgplichten tegen cybercrime (*Nederlands Juristenblad* 24-04-2015, afl. 16), 2015.
116 AG Connect, *Veel kritieke lekken door open source in standard apps*, 2021. https://www.agconnect.nl/artikel/veel-kritieke-lekken-door-open-source-standaard-apps

code eenmaal geïntegreerd is in het eigen pakket, kan het veel inspanning kosten om de onderliggende component in het geval van een kwetsbaarheid te updaten. Het softwarepakket is dan immers afhankelijk van een bepaalde versie van het component. Fabrikanten hebben ook niet altijd de juiste kennis in huis om componenten van anderen te kunnen updaten.[117]

Architectuur
De vierde factor die bijdraagt aan de aanwezigheid van kwetsbaarheden betreft het gegeven wanneer verschillende lagen in de architectuur van het product onderling niet consistent zijn. Het is essentieel voor de werking van software dat de verschillende onderdelen waaruit het bestaat op elkaar aansluiten. De aansluiting van verschillende componenten moet op een gecontroleerde manier tot stand zijn gekomen, onder het toeziend oog van een persoon met veel ervaring, voldoende kennis en die een groot belang heeft bij de beveiliging van een product.[118]

Configuratie
Een laatste factor, die niet zozeer bijdraagt aan het ontstaan van kwetsbaarheden, maar wel de gevolgen kan beperken, is de wijze waarop de software wordt geconfigureerd door de fabrikant (de standaard-instellingen). Daarbij gaat het onder meer om welke rechten er aan de verschillende soorten gebruikers kunnen worden toegekend, hoe deze rechten standaard staan ingesteld en of het mogelijk is als afnemer om deze rechten te beperken.

Verschillende factoren dragen bij aan het ontstaan van kwetsbaarheden tijdens de levenscyclus van een product. Vaak wordt doorgebouwd aan een bestaand product, daardoor wordt software steeds complexer. Ook kan de gebruikte programmeertaal bijdragen aan het ontstaan van fouten en het gebruik van bestaande componenten en (inconsistente) lagen in de architectuur kwetsbaarheden introduceren.

Als (veiligheids)problemen gekoppeld zijn aan fundamentele keuzes in het product kan dat een belemmering vormen voor de fabrikant om het probleem bij de wortel aan te pakken. Hiervoor is namelijk een investering nodig in de vorm van geld en/of capaciteit voor het oplossen van het probleem. De keuze van de fabrikant om in plaats daarvan alleen de kwetsbaarheid te repareren is verklaarbaar, maar soms is een herziening van de basis nodig om het echte (veiligheids)probleem op te lossen.

4.1.2 Kwetsbaarheden opsporen tijdens de levenscyclus
Fabrikanten hebben processen ingericht om kwetsbaarheden op te sporen tijdens de ontwikkeling van een product en om kwetsbaarheden op te sporen als het product al in gebruik is. In deze paragraaf gaan wij nader in op welke maatregelen fabrikanten kunnen treffen om kwetsbaarheden op te sporen en welke dilemma's daarbij voor hen ontstaan.

117 Tsai, O., *Infiltrating Corporate Intranet Like NSA*, 2020. https://i.blackhat.com/USA-19/Wednesday/us-19-Tsai-Infiltrating-Corporate-Intranet-Like-NSA.pdf
118 Anderson, R., *Security Engineering*, 2020.

Handelingsperspectief van de fabrikant

Fabrikanten sporen kwetsbaarheden op door middel van het doen van verschillende testen, zowel voor, tijdens als na het afronden van het ontwikkelproces. Voor open source software is de broncode voor iedereen inzichtelijk, daarom kunnen fouten in de code worden opgespoord door derde partijen, ook zonder dat daar specifiek naar wordt gevraagd. Voor *closed source code* is dit niet mogelijk, en de fabrikant moet zelf het initiatief nemen tot het uitvoeren van een audit.

Van een fabrikant kan verwacht worden dat deze een constante veiligheidsanalyse maakt van de gehele architectuur van het product (zie referentiekader: de rol van fabrikant en afnemer in hoofdstuk 2). Fabrikanten gebruiken verschillende methodologieën voor het ontwikkelen van software, bijvoorbeeld de *Secure Development Lifecycle* (SDLC).[119] Onderdeel hiervan is dat fabrikanten op verschillende momenten tijdens het ontwikkelproces (tijdens de initiële ontwikkeling maar ook bij het uitbrengen van patches) testen op kwetsbaarheden. Dat testen wordt gedaan op individuele componenten (*unit testing*), samenhang tussen componenten (*integration testing*) en het gehele product (audit[120] of *security code review*).

Door het gebruik van geautomatiseerde tools zijn fabrikanten in staat om meer kwetsbaarheden uit software te halen. Op die manier hopen zij de levensduur van het software te kunnen verlengen. Maar de *security code reviews* van het gehele product herkennen echter niet altijd het type kwetsbaarheden dat we hier behandelen. Kwetsbaarheden zijn niet (altijd) het gevolg van fouten in de broncode, maar ook een gevolg van de samenhang binnen het product. De fabrikant heeft dan ook nog de mogelijkheid om het product uitvoerig te testen op de beoogde werking (*end-to-end testing*). Uit interviews met fabrikanten blijkt dat voor oudere producten *end-to-end testing* zeer tijdrovend kan zijn, omdat oudere producten vaak bestaan uit een grote hoeveelheid broncode.

Fabrikanten kunnen kwetsbaarheden ook opsporen zonder derden daarbij directe toegang tot de broncode te geven. Veel fabrikanten hebben *bug bounty* programma's waarbij ethische hackers in ruil voor een beloning op zoek gaan naar kwetsbaarheden in de software. Deze ethische hackers zoeken handmatig en gaan als eerste op zoek naar eenvoudig vindbare kwetsbaarheden die het gevolg zijn van fundamentele ontwerpkeuzes en problemen in de samenhang of configuratie.

Veel van deze *bug bounty* programma's zijn voor iedereen toegankelijk, maar sommige fabrikanten kiezen ook voor een gesloten variant of hebben geen *bug bounty* programma. Ook krijgen fabrikanten soms informatie over kwetsbaarheden doorgestuurd uit *bug bounty* programma's van andere partijen, zoals toeleveranciers of klanten. Citrix had bijvoorbeeld ten tijde van het voorval alleen een gesloten *bug bounty* programma, het bedrijf is recentelijk gestart met een open programma.

119 Zie bijvoorbeeld https://owasp.org/www-pdf-archive/Jim_Manico_(Hamburg)_-_Securiing_the_SDLC.pdf

120 Onderdeel van de audit die de fabrikant doet is bijvoorbeeld threat modeling (het identificeren van bedreigingen en mitigerende maatregelen) en pentesting (het testen op kwetsbaarheden en hiermee proberen in te breken op het systeem). Pentesten kan deels geautomatiseerd worden gedaan, maar ook deze software kan weer kwetsbaarheden bevatten (zie bijvoorbeeld https://arstechnica.com/gadgets/2021/08/critical-cobalt-strike-bug-leaves-botnet-servers-vulnerable-to-takedown/).

Fabrikanten moeten een overzicht hebben van de afnemers van een softwareproduct (zie referentiekader in hoofdstuk 2). In dat geval kan de fabrikant afnemers snel waarschuwen in het geval van een kwetsbaarheid. Niet alle fabrikanten hebben een *up-to-date* overzicht van wie de afnemers van een product zijn. Dit komt omdat producten niet altijd rechtstreeks worden verkocht aan een afnemer; regelmatig gaat dit via allerlei tussenpartijen. Uit interviews blijkt dat sommige fabrikanten dit hebben opgelost door contactgegevens van afnemers, ook als een product wordt verkocht via een tussenpartij, te koppelen aan hun eigen overzicht. Vanuit het veiligheidskundig perspectief lijkt het logisch dat fabrikanten een overzicht hebben van de afnemers van een product. Het kan echter een dilemma opleveren dat onder andere raakt aan de autonomie van de afnemer. Zo is het niet mogelijk hen te dwingen zich bij een fabrikant te registreren en inzage te geven in hoe het systeem is geïnstalleerd.

Een trend van de laatste jaren is dat fabrikanten producten naar de *cloud* verplaatsen (*Software as a Service*) om beter te kunnen testen en sneller patches toe te passen op systemen van klanten. Het patchen van een product wordt hiermee de verantwoordelijkheid van de fabrikant. Dit brengt echter voor afnemers ook nadelen met zich mee, zie paragraaf 4.2.

Asymmetrie: fabrikant moet alles vinden, hackers hebben maar één lek nodig
De fabrikant moet veel tijd en moeite steken in het opsporen van kwetsbaarheden, voor en na het op de markt brengen van software. Met technieken zoals *end-to-end testing* kunnen veel kwetsbaarheden uit software gehaald worden. Maar het kost veel moeite om een enkele kwetsbaarheid op te sporen. Op het gebied van preventie hebben fabrikanten vaak al uitgebreide procedures opgenomen in het ontwikkelproces. Problemen in software die al langer bestaan (zie ontstaansgeschiedenis in 4.1.1) zijn onoverkomelijk vanwege de omvang en de preventieparadox. Het is niet mogelijk alle kwetsbaarheden op te sporen.

Aanvallers proberen met verschillende methodes, bijvoorbeeld door een *brute-force attack*, een kwetsbaarheid in een systeem te vinden. Soms doen ze dat aan de hand van bepaalde aanwijzingen (bijvoorbeeld met informatie uit een CVE), maar ook regelmatig komen ze per toeval een kwetsbaarheid tegen. Aanvallers hebben soms genoeg aan een enkel lek om volledige toegang tot een systeem te krijgen. Daarmee ontstaan een onbalans tussen aanvaller en verdediger (fabrikant).

Waar het vroeger nodig was zelf op het internet op zoek te gaan naar kwetsbare servers, een tijdrovend proces, is dat tegenwoordig makkelijker gemaakt door het gebruik van diensten die het internet scannen.[121] Aanvallers kunnen hiermee gemakkelijk een lijst van IP-adressen kopen die betrekking hebben op een (net gepubliceerde) CVE. Een aanvaller die een kwetsbaarheid heeft ontdekt heeft dus meteen toegang tot een lijst aan potentieel kwetsbare servers.

121 Bijvoorbeeld Shodan (https://www.shodan.io), een zoekmachine die het internet scant en benaderbare IP-adres en poortcombinaties indexeert. Als een server wordt geïndexeerd dan is de server via het internet te benaderen. Dat betekent niet automatisch dat een server ook kwetsbaar is. Dat is iets wat de hacker moet nagaan.

Relatie fabrikant en ethische hackers / redteams

Ethische hackers leveren een belangrijke bijdrage aan het opsporen van kwetsbaarheden. *Bug bounties* (een beloning ontvangen als men een kwetsbaarheid meldt) zijn daarbij een relevante prikkel. Zo hebben de meeste grote fabrikanten een *bug bounty* programma[122] waarmee ethische hackers geld kunnen verdienen voor het opsporen en rapporteren van kwetsbaarheden. Ook kan opsporen en publiceren over een specifieke kwetsbaarheid bijdragen aan de bekendheid van een hacker of hackersgroep. Dit mechanisme in combinatie met potentieel financieel gewin zorgt ervoor dat er veel wordt gespeurd naar kwetsbaarheden door derde partijen en leidt ertoe dat veel kwetsbaarheden worden gevonden.

Maar kwetsbaarheden vormen steeds vaker een potentiele aanvalsroute (zie 4.1.3) en het kost fabrikanten veel moeite om kwetsbaarheden te voorkomen en te verhelpen (zie 4.1.1). In dat licht zou het fabrikanten helpen en systemen beschermen om kwetsbaarheden geheim te houden. Het is mogelijk te publiceren over kwetsbaarheden zonder daarbij de details van een kwetsbaarheid prijs te geven. Maar sommige fabrikanten kiezen er bewust voor om niet alle (informatie over de aanwezigheid van) kwetsbaarheden openbaar te maken.[123] Deze gedachte staat echter haaks op het tijdig bekend maken van informatie over kwetsbaarheden voor het mitigeren en bestrijden van potentiele risico's. Er is een duidelijke impuls ter voorkoming dat informatie over kwetsbaarheden openbaar wordt. Bekendmaken stelt afnemers in staat de gevolgen te bestrijden maar leidt tegelijkertijd tot een nieuw veiligheidsprobleem: een dilemma.

Degene die de kwetsbaarheid vindt meldt dit niet altijd aan de fabrikant. Kwetsbaarheden in software zijn handelswaar, die niet alleen worden gemeld aan de fabrikant (al dan niet tegen een beloning), maar die ook worden aangeboden aan de hoogste bieder. Zo is het interessant voor statelijke actoren en criminelen om een lijst met onbekende kwetsbaarheden te bezitten zodat zij deze zelf kunnen gebruiken.[124] Ook worden commerciële spywareproducten verkocht, waarbij niet zichtbaar is of dit product gebaseerd is op onbekende kwetsbaarheden en ook niet aan welke partijen en voor welk doeleinde deze producten worden ingezet.[125]

122 Voor een overzicht van bug bountry programma's, zie bijv. https://www.bugcrowd.com/bug-bounty-list/.

123 Bijvoorbeeld Palo Alto, waar volgens de beveiligingsonderzoeker die de kwetsbaarheid vond geen CVE was gepubliceerd over een kwetsbaarheid (die ook al door de fabrikant gerepareerd was) in GlobalProtect. Bron: https://blog.orange.tw/2019/07/attacking-ssl-vpn-part-1-preauth-rce-on-palo-alto.html. Het is onduidelijk of Palo Alto via directe kanalen met hun klanten heeft gecommuniceerd over de kwetsbaarheid. De Onderzoeksraad heeft dit niet kunnen verifiëren omdat Palo Alto niet heeft gereageerd op onze verzoeken om mee te werken aan het onderzoek.

124 Perlroth, N., *This is how they tell me the world ends: the cyberweapons arms race*, 2021.

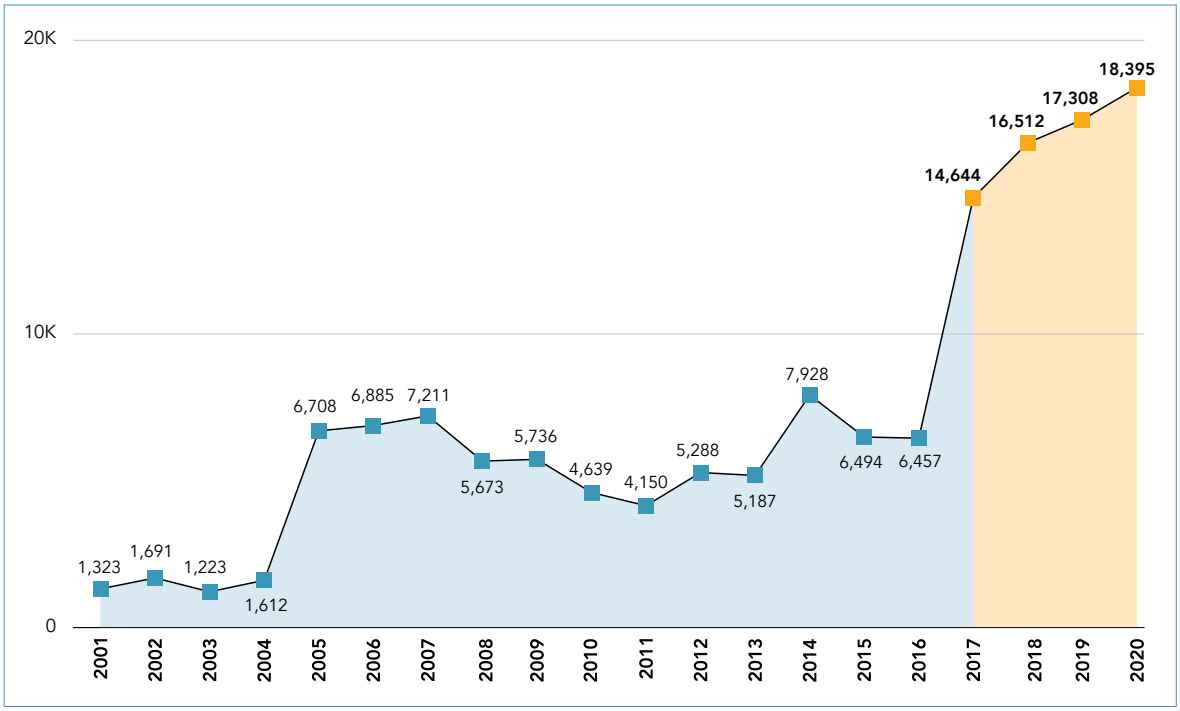
125 https://www.wired.com/story/nso-group-hacks-ios-android-observability/, https://www.nrc.nl/nieuws/2021/07/26/de-overheid-moet-stoppen-met-gebruik-van-zero-day-software-a4052412

Ethische hackers worden met beloningen aangespoord om te zoeken naar kwetsbaarheden in software. Daardoor worden veel kwetsbaarheden opgespoord. Fabrikanten sporen daarnaast kwetsbaarheden op door het doen van verschillende testen. Maar het is niet mogelijk *alle* kwetsbaarheden op te sporen. Kwetsbaarheden vormen steeds vaker een aanvalsroute. Een kwetsbaarheid bekend maken kan organisaties helpen zich beter te wapenen tegen mogelijk misbruik, het kan aanvallers echter ook in staat stellen de kwetsbaarheid te misbruiken. Temeer omdat hackers soms maar één lek nodig hebben om toegang te krijgen tot een systeem en het voor hen relatief eenvoudig is om kwetsbare servers op te sporen. Daarmee ontstaat een dilemma met onveiligheid tot gevolg.

4.1.3 De rol van kwetsbaarheden bij digitale onveiligheid

Kwetsbaarheden spelen een grotere rol

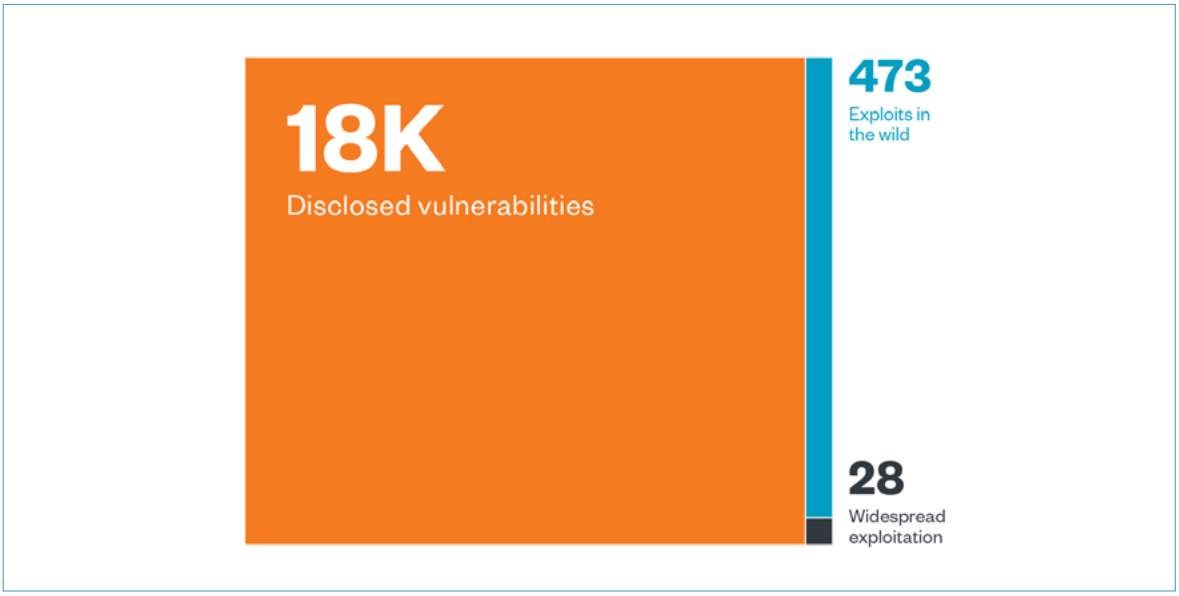
Organisaties worden jaarlijks blootgesteld aan een groot en alsmaar groeiend aantal kwetsbaarheden. In 2020 ging het om ruim 25 duizend kwetsbaarheden. Een deel van deze kwetsbaarheden, 18 duizend, is in 2020 gepubliceerd met een CVE nummer[126] (zie Figuur 16). Slechts een klein deel van het aantal gepubliceerde kwetsbaarheden (circa 3%) wordt gebruikt om organisaties en/of individuen aan te vallen. Een nog kleiner deel (0,5%) is in de praktijk succesvol gebruikt om een wijdverbreide aanval zoals bij de beveiligingslekken beschreven in hoofdstuk 3 tot stand te laten komen (zie Figuur 17). Toch groeit dit aantal en experts waarschuwen dat dit slechts het topje van de ijsberg is.[127]



Figuur 16: Het aantal CVE meldingen per jaar. (Bron: Trend Micro)

126 Er zijn ook veel kwetsbaarheden die door de fabrikant worden verholpen zonder dat deze openbaar worden gemaakt. https://vulndb.cyberriskanalytics.com/#statistics

127 AG Connect, *Einde van de oneindige reeks softwarefouten in zicht*, 2021.



Figuur 17: Het aantal exploits en wijdverbreide aanvallen in relatie tot het totaal aantal gerapporteerde kwetsbaarheden. (Bron: Trend Micro)

Ook de gevolgen van deze aanvallen worden steeds groter. Zo waarschuwde de NCTV in het Cybersecuritybeeld 2020 voor aanvallers die op zoek gaan naar zwakke schakels in de leveranciersketen als opstap naar interessante doelen en de grote gevolgen daarvan.[128] Waar een kwetsbaarheid in een softwarepakket in het verleden niet automatisch tot ernstige gevolgen leidde, kan dit tegenwoordige verregaande gevolgen hebben voor achterliggende, afhankelijke systemen, zoals geïllustreerd door de ketenaanvallen waarbij gebruik werd gemaakt van kwetsbaarheden in SolarWinds en Kaseya (zie paragraaf 3.3 voor een beknopte analyse).

Kwetsbaarheden zoals in de door de Onderzoeksraad onderzochte voorvallen spelen dus een steeds grotere rol bij cyberaanvallen en worden door aanvallers steeds vaker gebruikt als startpunt voor het opzetten van een aanval.[129] Vooral grote organisaties (zoals overheidsorganisaties en vitale bedrijven) lopen risico om aangevallen te worden met behulp van deze aanvalsvector.[130] Het misbruiken van kwetsbaarheden in software om *ransomware*-aanvallen uit te voeren is een economisch aantrekkelijke werkwijze voor *ransomware*-bendes, zo is sinds 2020 duidelijk geworden.

Het steeds vaker voorkomen van wijdverspreide aanvallen waarbij gebruik wordt gemaakt van een kwetsbaarheid toont het belang van tijdig patchen van software en/of mitigeren van een kwetsbaarheid aan. Het gebruik van software brengt risico's met zich mee. Zo is het voor afnemers niet altijd mogelijk om te voorspellen welke van de kwetsbaarheden uiteindelijk een gevaar zullen vormen voor hun organisatie. Dit is bijvoorbeeld afhankelijk van hoe eenvoudig het is om de kwetsbaarheid in de software actief te misbruiken, of er een mitigatie voorhanden is en hoe eenvoudig het is om deze toe te passen, en van de

128 NCTV, *Cybersecuritybeeld Nederland 2020*, 2020. https://www.ncsc.nl/documenten/publicaties/2020/juni/29/csbn-2020

129 Modderkolk H., 'Overheid doet te weinig tegen ransomeware', *De Volkskrant*, 4 augustus 2021; CISA, *Alert (AA21-209A) Top Routinely Exploited Vulnerabilities*, 2021.

130 Coveware, *Ransomware Attack Vectors Shift as New Software Vulnerability Exploits Abound,* 2021. https://www.coveware.com/blog/ransomware-attack-vectors-shift-as-new-software-vulnerability-exploits-abound

versie en configuratie van een product. Het verhelpen van kwetsbaarheden door een mitigerende maatregel of het toepassen van patches vereist van een investering van de organisatie. In de meeste gevallen krijgen ze daar niet direct meer veiligheid voor terug.

Voor fabrikanten en afnemers is het voorkomen, tijdig mitigeren of patchen van een kwetsbaarheid niet de enige verdedigingslinie. In paragraaf 4.2 wordt verder stilgestaan bij maatregelen die organisaties kunnen nemen om de veiligheidsrisico's van kwetsbaarheden in software te mitigeren. Voorbeelden hiervan zijn het gebruik van een firewall om toegang tot het netwerk in te dammen en het gebruik van redundante hard- en software zodat bij het bekend worden van een kwetsbaarheid snel omgeschakeld kan worden naar een ander product.

De keerzijde van patchen en mitigeren
Als de fabrikant software op de markt heeft gebracht waarvan later blijkt dat deze een kwetsbaarheid bevat, zal de fabrikant doorgaans een patch uitbrengen en afnemers adviseren om de software te patchen. In het geval er nog geen patch beschikbaar is kan een fabrikant ook een mitigatie publiceren om het acute gevaar weg te namen. Maar patchen en mitigeren zijn niet altijd makkelijk te nemen oplossingen.

Patches en mitigaties vormen in zekere zin ook een risico. Het is niet altijd te voorzien wat het effect van een patch of mitigatie zal zijn op reeds geconfigureerde en in gebruik zijnde software. Elke mitigatie en patch kan leiden tot (deels) onvoorziene gevolgen, bijvoorbeeld voor de compatibiliteit van belendende/verbonden systemen. In sommige gevallen kunnen patches zelfs zorgen voor verstoringen of het geheel uitvallen van systemen.[131] Ook kunnen *p*atches en mitigaties weer nieuwe fouten in de software of kwetsbaarheden introduceren, zoals bijvoorbeeld het geval bij de patch van Microsoft om de problemen met de *print-spooler* op te lossen die leidde tot problemen bij het printen.[132]

Kwetsbaarheden in software vormden een escalatiefactor. De voorvallen in dit onderzoek illustreren dat. Nadat de kwetsbaarheden bekend waren gemaakt (bijvoorbeeld door de publicatie van een CVE of een *security bulletin*), zochten aanvallers met geautomatiseerde hulpmiddelen naar alle servers die nog niet waren gepatcht om deze aan te vallen. Een mitigatiemaatregel kan bovendien informatie geven over hoe een kwetsbaarheid misbruikt kan worden. De onderzochte voorvallen laten zien dat dit in een tijdbestek van enkele dagen kan gebeuren (of dat de aanvallen al plaatsvonden, in het geval van een *zero day*). De publicatie van een kwetsbaarheid kan de opmaat vormen naar wijdverbreide aanvallen.

Ook aan de kant van afnemers kunnen problemen ontstaan rondom het patchen. Zo is het vanwege het grote aantal patches dat jaarlijks verschijnt niet altijd mogelijk om alles tijdig te installeren. Afnemers hebben ook niet altijd een up-to-date overzicht van welke software gepatcht moet worden, hebben vaak geen zicht op welke onderliggende (kwetsbare) componenten een softwarepakket bevat en zijn niet altijd overtuigd van de

131 https://www.trendmicro.com/vinfo/us/security/news/vulnerabilities-and-exploits/the-nightmares-of-patch-management-the-status-quo-and-beyond
132 https://www.zdnet.com/article/microsofts-printnightmare-patch-is-now-causing-problems-for-some-printers/

noodzaak tot *p*atchen. Hier wordt verder op ingegaan in paragraaf 4.2. Het uitbrengen van een mitigatiemaatregel voordat een patch verschijnt kan een *good practice* zijn omdat afnemers deze in de regel wel gelijk toepassen na publicatie ervan. Zo zorgt een fabrikant ervoor dat de software van afnemers veilig is. Nadeel is dat sommige afnemers de noodzaak van patchen dan nog minder gaat inzien.

Het aantal kwetsbaarheden in software groeit en de gevolgen van aanvallen worden groter. Kwetsbaarheden spelen een steeds grotere rol bij cyberaanvallen en kunnen door aanvallers worden gebruikt als startpunt voor het opzetten van een aanval. Dit onderstreept het belang van tijdig patchen. Maar patchen en mitigeren vormen tegelijkertijd een risico omdat dit kan leiden tot verstoringen of de introductie van nieuwe kwetsbaarheden. De organisatie moet het besluit om te patchen daarom goed doordenken vanuit het ICT-landschap van de organisatie. De publicatie van een kwetsbaarheid kan de opmaat vormen naar wijdverbreide aanvallen.

4.1.4 Prikkels voor veiligere software

Naast de meer intrinsieke factoren, die direct betrekking hebben op het ontwikkelproces bij de fabrikant, spelen ook factoren ten aanzien van regulering en aansprakelijkheid een rol bij het ontstaan van kwetsbaarheden.

Overheden en organisaties die software gebruiken hebben op dit momenteel weinig mogelijkheden om softwarefabrikanten te verplichten cybersecurity in hun producten te borgen. Daarmee komen problemen die ontstaan als gevolg van kwetsbaarheden grotendeels te liggen bij de afnemer van een product. Afnemers moeten hierop extra bedacht zijn bij het aanschaffen van software. Eenmaal aangeschaft kan een afnemer weinig meer doen om te controleren of een product veilig is.

Positie afnemer ten opzichte van fabrikant

Sommige (grote) afnemers, zoals overheden en vitale bedrijven, zijn in staat om met behulp van geavanceerde software en uitgebreide analyses zelf kwetsbaarheden in software op te sporen. Maar niet alle afnemers bevinden zich in de positie om zelf de software te kunnen testen of ontleden (reverse engineering), of om autonoom een volledige risicobeoordeling te kunnen doen (zie ook paragraaf 4.2 over informatie-asymmetrie en referentiekader transparantie). Uit interviews blijkt bovendien dat niet alle organisaties weten hoe ze eisen moeten stellen en af moeten dwingen om een fabrikant verantwoording af te laten leggen. Fabrikanten laten in overeenkomsten doorgaans vastleggen dat zij beperkt aansprakelijk zijn voor de gevolgen van eventuele kwetsbaarheden in de software. Daarmee wordt een kwetsbaarheid een probleem van de afnemer, en niet van de fabrikant.

Verder verbieden fabrikanten met de voorwaarden die zij verbinden aan de aanschaf en het gebruik van hun software dat afnemers het product ‘openmaken’ om te kijken hoe het werkt en welke componenten het bevat. Dit doen fabrikanten vanuit het oogpunt van bedrijfsgeheim. Deze afspraken werken belemmerend voor afnemers om het product aan een eigen onderzoek te onderwerpen en om kwetsbaarheden te melden die uit zo’n

onderzoek volgen. Ook bepalen fabrikanten via de voorwaarden dat zij niet aansprakelijk kunnen worden gesteld voor de gevolgen van kwetsbaarheden in de software.[133]

Wettelijke eisen

Los van het stellen van eisen aan een softwareproduct door afnemers worden er ook nauwelijks eisen gesteld door de overheid voor het op de markt brengen van software, het onderhoud tijdens de levenscyclus en de rol van de fabrikant tijdens incidentbestrijding. De Wbni[134] verplicht aanbieders van essentiële diensten tot het nemen van beveiligingsmaatregelen met betrekking tot hun netwerk en informatiesystemen (bijvoorbeeld het melden van cybersecurityincidenten), maar dit geldt niet voor softwarefabrikanten. Bovenstaande toont aan dat in het stelsel van partijen, met name op het gebied van wet- en regelgeving, een lacune zit aan de kant van de fabrikanten.

Nationale initiatieven

Er zijn diverse initiatieven om met wet- en regelgeving te komen voor het op de markt brengen van software. Zo zijn het ministerie van Economische Zaken en Klimaat en het ministerie van Justitie en Veiligheid zijn in de vorm van de *roadmap* digitaal veilige hard- en software (*roadmap* DVHS) gekomen met een initiatief waarin ze een pakket aan meetregelen voorstellen om onveiligheden in hard- en software te voorkomen, kwetsbaarheden te detecteren en de gevolgen daarvan te mitigeren.[135] De maatregelen in de *roadmap* zijn gericht op zowel preventie, detectie als mitigatie en bestaan onder meer uit wettelijke eisen en het aansprakelijk stellen van fabrikanten voor schade als gevolg van digitale onveiligheid. Zorg om aansprakelijkheid moet een prikkel vormen voor fabrikanten om voorzorgsmaatregelen te nemen of schade te beperken. De maatregelen zijn met name gericht op kleinere apparaten (IoT[136]), maar zijn universeel toepasbaar op andere typen software. Die vraag die rijst is in hoeverre deze maatregelen ook toegepast moeten worden op alle veiligheidskritische software en software in de algemene zin.

Internationale initiatieven

Verschillende internationale overheden nemen initiatief om de lacune in wet- en regelgeving aan te pakken. Op 27 juni 2019 is de Europese *Cybersecurity Act* in werking getreden.[137] Met deze nieuwe regels voor cyberbeveiliging worden onder meer het mandaat van ENISA[138] versterkt en een cybersecurity certificeringskader geïntroduceerd. Een ander recent voorbeeld van een initiatief op het gebied van wetgeving is de cyberwetgeving in de VS, waarmee eisen worden gesteld aan software die door de overheid wordt aangekocht.[139] Australië heeft ook plannen om regulering voor

133 Cyber Security Raad (CSR), *Integrale aanpak cyberweerbaarheid*, 2021.; Tjong Tjin Tai, E. en Knoops, B., Zorplichten tegen cybercrime (*Nederlands Juristenblad* 24-04-2015, afl. 16), 2015; Anderson, R., *Security Engineering*, 2020.

134 Wet Beveiliging Netwerk- en Informatiesystemen (Wbni) voor digitale dienstverleners, zie https://wetten.overheid.nl/BWBR0041515/2021-07-01

135 Ministerie van Economische Zaken en Klimaat en ministerie van Justitie en Veiligheid, *Roadmap digitaal veilige hard- en software*, 2018.

136 *Internet-of-Things*, bijvoorbeeld een smart TV, slimme koelkast, verbonden temperatuursensoren, et cetera.

137 https://ecer.minbuza.nl/-/europese-cyber-security-act-van-kracht; https://digital-strategy.ec.europa.eu/en/policies/cybersecurity-act

138 Het Europees agentschap voor netwerk- en informatiebeveiliging.

139 https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/, https://www.nytimes.com/2021/05/12/us/politics/biden-cybersecurity-executive-order.html

cybersecurity te verbeteren.[140] De nadruk bij dit voorstel ligt op IoT en organisaties die persoonlijke informatie verwerken. Ten aanzien van softwareveiligheid zet Australië in op sterkere afspraken over *responsible disclosure* als prikkel voor fabrikanten om kwetsbaarheden sneller te patchen. In China wordt het misbruiken van kwetsbaarheden strafbaar en komen er sancties voor fabrikanten die nalaten patches uit te brengen voor gemelde kwetsbaarheden.[141]

De Cyber Security Raad (CSR) concludeert in zijn laatste adviesrapport dat er ondanks een aantal belangrijke initiatieven, zowel binnen de Europese Unie als Nederland, nog geen sluitend mechanisme is van verantwoordelijkheid voor veilige hard- en software.[142] Volgens de CSR moeten fabrikanten meer verantwoordelijk worden gehouden voor economische schade als gevolg van het verzaken van zorgplicht op het gebied van cybersecurity. Deze zorgplicht moet gaan bijdragen aan het beschermen van burgers en bedrijven tegen cybercrime.

Handhaving
Als handhaving van wettelijke eisen wordt ingevuld middels certificering van software, dan is er een risico op perverse effecten. Zo heeft de certificeringsinstantie een verdienmodel richting degene die gecertificeerd wil worden en kan een softwarefabrikant bij het certificering van software gaan voor de weg van de minste weerstand. Concurrentie tussen verschillende certificeringsinstanties leidt niet altijd tot betere standaarden en kan ook uitmonden in een race naar de bodem (het principe van '*maximum complacency*', de fabrikant kiest ervoor om certificering door één enkele certificeerder te laten bekrachtigen en verzet zich tegen pogingen om hem ertoe te brengen het product te verbeteren).[143]

140 Commonwealth of Australia, *Strengthening Australia's cyber security regulations and incentives*, 2021.
141 https://therecord.media/chinese-government-lays-out-new-vulnerability-disclosure-rules/
142 CSR, *Integrale aanpak cyberweerbaarheid*, 2021.
143 Anderson, R., *Security Engineering*, 2020.

Ervaringen uit het verleden: Common criteria, ISO 27001 en BitSight

Common Criteria
De *Common Criteria for Information Technology* is een internationale standaard voor computerbeveiliging. Deze standaard kent een aantal problemen: zo zijn de certificeringskosten hoog, de standaard generiek beschreven (de techniek is er buiten gelaten, inclusief *usability*, een belangrijke randvoorwaarde voor veiligheid), de standaard kan niet goed omgaan met snelle ontwikkelingen in de praktijk/ toepassing, de standaard wordt heel verschillend toegepast (bijvoorbeeld streng in Duitsland, erg los in Nederland) en vormt aansprakelijkheid geen onderdeel van de standaard.

ISO 27001
ISO 27001[144] werkt voor bedrijven vooral als manier om geld te verdienen. Certificering kost veel geld en is een bron van inkomsten voor de certificeringsinstanties. Bij de aanvraag van een certificaat door een bedrijf is de certificeringsinstantie afhankelijk van informatie die door het bedrijf wordt aangeleverd. Zo is het mogelijk voor de aanvrager om aan te geven dat bepaalde beveiligingsmaatregelen zijn genomen, terwijl die in de praktijk niet zijn geïmplementeerd. Er vind geen daadwerkelijke onafhankelijke toetsing plaats. Bijna alle grote lekken zijn voorgekomen bij bedrijven die met 27001 gecertificeerd zijn.[145]

BitSight
BitSight, anders dan ISO 27001 een initiatief van de private sector, is een bedrijf die het internet afspeurt op zoek naar servers van bedrijven en overheidsinstellingen. Servers die zijn ontdekt worden gescand en krijgen een beveiligingsscore toegekend (bijvoorbeeld op basis van hoeveel servers (niet) zijn gepatcht). Ze zijn daarmee niet afhankelijk van informatie aangeleverd door bedrijven (bij ISO 27001 de aanvrager) en komen op basis van de scans tot een score. Maar dit heeft ook nadelige effecten. Zo zijn bedrijven huiverig voor het opzettelijk verbinden van kwetsbare servers met het internet (bijvoorbeeld voor het trainen van werkgevers, studenten, e.a.). Zodra deze server opgemerkt wordt opgemerkt door BitSight heeft dit negatieve invloed op de beveiligingsscore van het bedrijf.

Handhaving kan alleen als fabrikanten verplicht worden transparant te zijn over hoe de software werkt en zodat derden de veiligheid ervan kunnen beoordelen. De *Executive Order* ter verbetering van de cyberveiligheid in de VS gaat hier op in en noemt dat er dringend behoefte is aan strengere en meer voorspelbare mechanismen om ervoor te zorgen dat producten veiliger en overeenkomstig hun beoogde werking functioneren.[146]

144 Een ISO standaard voor informatiebeveiliging. Zie https://www.iso.org/isoiec-27001-information-security.html.
145 Anderson, R., *Security Engineering*, 2020.
146 https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

Economische prikkels

De voorbeelden in dit onderzoek laten zien dat softwareproducten dynamisch zijn. Ze worden namelijk regelmatig aangepast voor nieuwe functionaliteit en het verhelpen van kwetsbaarheden. Tegelijkertijd kennen de producten vaak een lange geschiedenis omdat kan worden voortgebouwd op bestaande onderdelen. Dit maakt dat het voor fabrikanten een grote investering kan zijn om de fundamentele oorzaken (*root causes*) van onveiligheid, zoals beschreven in paragraaf 4.1.1, aan te pakken. Dat zou namelijk betekenen dat zij software dat het resultaat is van tientallen jaren van ontwikkeling opnieuw zouden moeten opbouwen.

Tegenover deze investering staan weinig economische prikkels. Verzekeraars verzekeren niet alleen organisaties die software gebruiken, maar ook fabrikanten die software maken. In die laatste rol eisten zij van fabrikanten dat zij de aansprakelijkheid voor de gevolgen van onveilige software afwentelen op de afnemers van de software. De Cyber Security Raad schrijft daarover dat verzekeraars idealiter eisen stellen aan zowel de fabrikant als aan de organisatie die de software gebruikt.[147]

Een fabrikant kan ook een economische prikkel ondervinden door waardedaling als gevolg van onveiligheid (aandeelhouders). Zo hebben aandeelhouders van SolarWinds het bedrijf aangeklaagd: volgens de aandeelhouders hebben de *private equity* bedrijven die eigenaar zijn van het bedrijf gekozen voor korte termijn winst boven cybersecurity ('*goldrush* bij investeerders in SaaS business').[148]

Daarnaast kan de materiële verplichting om de onveiligheid weg te nemen zorgen voor een economische prikkel voor een fabrikant om meer te doen om te voorkomen dat er software met kwetsbaarheden op de markt zijn. Op dit moment ligt deze economische prikkel alleen bij de afnemers van de software.

147 Cyber Security Raad (CSR), *Integrale aanpak cyberweerbaarheid*, 2021.
148 https://www.scmagazine.com/home/solarwinds-hack/solarwinds-lawsuit-claims-private-equity-owners-sacrificed-cybersecurity-to-boost-short-term-profits/

Traceren en terugroepen in de voedselsector[149]

In de voedselsector zijn voedselbedrijven verplicht om te kunnen traceren aan wie ze voedselproducten hebben geleverd. Deze plicht geldt voor de gehele voedselketen, van primaire productie (zoals landbouw, veeteelt en visserij) tot de consument die het voedsel opeet. In elke schakel van deze keten moet een voedselbedrijf kunnen traceren waar hun grondstoffen vandaan komen en aan wie zij hun producten hebben geleverd. Deze verplichting wordt traceerbaarheid genoemd. Als een voedselbedrijf tot de ontdekking komt dat ze onveilig voedsel op de markt hebben gebracht, moet het binnen vier uur een distributielijst met alle afnemers[150] en afgenomen producten moeten samenstellen en desgewenst aanleveren aan de autoriteiten.

Voedselbedrijven zijn ook verplicht om dit voedsel uit eigen initiatief of op last van de autoriteiten terug te roepen (*recall*). In de praktijk vinden autoriteiten het voldoende als een voedselbedrijf zich beperkt tot een publicatie in een dagblad en/of op de eigen website, maar een 'zuivere *recall*' betekent dat het voedselbedrijf zijn afnemers zo direct mogelijk waarschuwt en oproept de producten terug te brengen en eventueel de producten zelf ophaalt bij de afnemer. Dat laatste gebeurt bijvoorbeeld bij *recalls* van personenauto's, wanneer het veiligheidsprobleem zo ernstig is dat de personenauto geen gebruik mag maken van de openbare weg.

Regulering en aansprakelijkheid spelen ook een rol bij het ontstaan van kwetsbaarheden. Op dit moment zijn er weinig mogelijkheden voor overheden en organisaties om fabrikanten te verplichten cybersecurity in hun producten te borgen. Afnemers weten niet altijd hoe ze eisen moeten stellen en een fabrikant verantwoording moeten laten afleggen. Daarmee wordt de kwetsbaarheid een probleem van de afnemer.

Er zijn nu nauwelijks regels voor het op de markt brengen van software. De huidige marktwerking van softwareproducten dwingt te weinig af dat veiligheidsrisico's goed worden beheerst. Kwetsbaarheden opsporen is tijdrovend, kost veel menskracht en is daarmee duur. In sommige gevallen kan het nodig zijn om een product opnieuw op te bouwen om het echte (veiligheids)probleem aan te pakken. De afwezigheid van economische prikkels verklaart dat fabrikanten deze afweging op dit moment niet maken.

149 Vanuit het idee dat er een keten is van producent via een aantal tussenstappen tot de consument, geldt de verplichte traceerbaarheid voor elk voedselbedrijf één stap terug en één stap vooruit in de keten (exclusief de stap naar de eindgebruiker ofwel de consument). Bron: Artikel 18, lid 1 van Verordening (EG) nr. 178/2002 in: Richtsnoeren voor de tenuitvoerlegging van de artikelen 11, 12, 14, 17, 18, 19 en 20 van Verordening (EG) nr. 178/2002 betreffende de algemene levensmiddelenwetgeving (26 januari 2010).

150 Voor de laatste schakel (de eindverbruiker, consument) geldt de verplichting om te traceren niet, maar sommige retailers registreren leveringen aan consumenten (deels) wel (online bestellingen, klantenkaarten et cetera).