6

Rijksoverheld

Datum
19 sept 2022

# memo

Open Source Security - advies en handreiking

## Management Samenvatting

- Het gebruik en de ontwikkeling van open source software (OSS) kent diverse voordelen en (daardoor) een groeiende belangstelling. Naast voordelen kent de inzet van OSS echter ook specifieke risico's en uitdagingen. Die zijn deels anders dan die gelden voor commerciële/closed source software (CSSI). Dit advies gaat over de security risico's van OSS en over de daarbij behorende afwegingen en beheersmaatregelen.
- Het blijkt dat het open, transparante karakter van OSS niet direct maakt dat het ook vrij is van kwetsbaarheden, dit in tegenstelling tot wat vaak wordt aangenomen. Het overgrote deel van de OSS (componenten) die in gebruik zijn bevat meerdere kwetsbaarheden, en regelmatig ook ernstige. OSS is daarmee niet inherent veiliger dan CSS. Het open karakter van OSS kan juist ook bepaalde nieuwe risico's introduceren.
- Veilige OSS vraagt om professionele softwareontwikkeling met behulp van moderne, geautomatiseerde ontwikkelvoorzieningen, zoals een geautomatiseerde CI/CD-straat met daaraan gekoppeld technische voorzieningen voor Software Composition Analysis (SCA), Static-, Interactive- en Dynamic Application Security Testing (respectievelijk SAST, IAST en DAST), vulnerability scanning, Software Bills of Materials (SBOM) en pentesten. Gebruikers van OSS zouden moeten vereisen dat die OSS op een dergelijke, professionele en moderne manier wordt ontwikkeld.
- De inzet van OSS kent verder specifieke continuïteitsrisico's. Er is namelijk veelal niet een specifieke (professionele) partij verantwoordelijk voor, of aanspreekbaar op, de continuïteit, het beheer en het tijdig patchen van de OSS. Dit beheer gebeurt vaak door een collectief van (vrijwillige) ontwikkelaars. Niet zelden is die groep klein en daarmee kwetsbaar. Het vraagt van OSS-gebruikers dat zij de ontwikkeling, professionaliteit en robuustheid van de betreffende OSS-community continu blijven volgen. En op basis hiervan afwegen of zij de betreffende OSS al dan niet willen (blijven) inzetten.
- Het ontbreken van een aanwijsbare eindverantwoordelijke partij voor de ontwikkeling en het beheer van OSS compliceert ook de security governance en compliance borging.
- De specifieke security risico's die gelden bij OSS vragen om een goede risico afweging. De inzetbaarheid zal o.a. afhangen van de robuustheid en professionaliteit van de betreffende OS-community, hun werkwijze en het inzetscenario voor de betreffende OSS. Dat kan betekenen dat OSS soms wel, en soms ook niet, toepasbaar is. Om bij deze afweging te helpen staat achterin dit advies een 'handreiking veilig gebruik open source software'.

5.1(2e) We vinden het om te beginnen een goed stuk! We hebben daarom niet zozeer suggesties voor de inhoud, maar met name ideeën voor de doelgroep, het herstructureren van de huidige tekst en de varianten m.b.t. OSS. Dit lichten we in de onderstaande comments verder toe.

**Met opmaak:** Engels (Verenigde Staten)

**Met opmaak:** Engels (Verenigde Staten)

**Met opmaak:** Engels (Verenigde Staten)

5.1(2e) Is de management samenvatting geschreven voor een technische doelgroep? Of bestuurders of beleidsmakers die in actie moeten komen? Wie is de doelgroep voor dit stuk? Ons advies is eerst de doelgroep te bepalen en op basis hiervan de tekst te herstructureren.

Omdat veel van de risico's ten aanzien van OSS niet inherent een probleem van OSS zijn, maar eerder te maken hebben met hoe de governance wordt ingericht (continuïteit/ onderhoud / support) en het vertrouwen in de partij die software levert, adviseren wij de doelgroep van bestuurders en/of beleidsmakers wel aan te spreken.

Zie de suggesties op pagina 2 hoe we adviseren om hier vorm aan te geven.

5.1(2e) Open Source Software (OSS) wordt in het stuk voornamelijk beschreven als 'OSS ontwikkeld door een community'. Dit is wellicht de meest bekende variant. Is er een specifieke reden waarom alleen deze variant wordt beschreven en niet de andere varianten van OSS?

## Aanleiding voor dit advies en open source afwegingskader

- OSS heeft de afgelopen decennia wereldwijd een belangrijke stempel gedrukt op de ontwikkeling van digitale dienstverlening. Op dit moment bestaat het grootste deel van de digitale producten en diensten zelfs uit één of meerdere OSS-componenten. Open source is daarmee overal. Tegelijkertijd groeit de interesse om OSS nog breder in te zetten. Dit geldt ook voor de overheid die zichzelf tot doel heeft gesteld om meer OSS in te zetten en zelf ontwikkelde software als open source vrij te geven via het 'open, tenzij' principe.
- Die keuze om het gebruik van open source binnen de overheid verder te stimuleren is gebaseerd op een aantal overwegingen. Zo moet de inzet van OSS onder ander de afhankelijkheid van commerciële IT-leveranciers beperken, de risico's van *vendor lock-in* verkleinen, het innovatief vermogen versterken, bijdragen aan een meer open en transparante digitale overheid en ertoe leiden dat IT-voorzieningen die bekostigd worden met publieke middelen ook vervolgens voor eenieder (her)bruikbaar zijn.
- Het ontwikkelen, en in zekere zin ook het inzetten, van OSS kent een andere dynamiek en andere uitdagingen dan die gelden voor meer 'reguliere' commercieel verkrijgbare IT-diensten, waar de overheid en private partijen vooral ervaring mee hebben. Een bredere inzet van OSS vraagt daarmee om andere werkwijzen en andere beheersafwegingen dan welke gebruikersorganisaties, zoals bij de overheid, gewend zijn en kennen. Om (overheids)organisaties hierbij te helpen is recent o.a. de 'toolbox open source' ontwikkeld.
- Eén van de punten waar de inzet en ontwikkeling van OSS verschilt van die van commerciële aanbieders is het risicomanagement. OSS kent andere risico's en vraagt om ander risicomanagement dan organisaties gewend zijn met closed source software (CSS) - ook op het gebied van security.
- Dit memo beschrijft een aantal van deze security risico's en aandachtspunten die samenhangen met de inzet en ontwikkeling van OSS, en geeft adviezen hoe (overheids)organisaties hier het beste mee zouden kunnen omgaan. Deze zijn vervolgens aan het einde van dit stuk samengebracht in een 'afwegingskader veilige inzet van open source software'. Hiermee poogt dit stuk een concrete bijdrage te leveren aan (een adequaat risicomanagement bij) de groeiende inzet van OSS. Gelet op het feit dat OSS-gebruik toeneemt en er een ambitie ligt om dit ook verder te versterken, is het adequaat kunnen beheersen van de bijbehorende security risico's van steeds groter belang.
- (Overheids)organisaties kunnen op verschillende manieren betrokken zijn bij OSS. Zo kunnen zij elders ontwikkelde OSS zelf inzetten, OSS zelf laten ontwikkelen of zelf ontwikkelde software, wanneer dit gereed is, als OSS ter beschikking stellen. Dit advies gaat niet specifiek in op ieder van deze scenario's maar behandelt een aantal generieke risico's en vraagstukken die in zekere mate voor al deze situaties gelden.

## Open source software; veiliger dan closed source?

- De ambitie om (overheidsbreed) meer OSS in te willen zetten, steunt~~, naast de hierboven genoemde overwegingen, ook~~ op de veronderstelling dat OSS veiliger is dan CSS omdat – zo is de aanname – eenieder die dat wil de source code van OSS kan inzien, kan inspecteren en zo nodig kan (laten) verbeteren mochten hierin kwetsbaarheden worden aangetroffen. Het open, transparante karakter van OSS zou zo moeten leiden tot het voorkomen dan wel sneller bekend worden van kwetsbaarheden, een sneller herstel daarvan en zo uiteindelijk tot een veiliger softwareproduct.
- Uit (wetenschappelijk) onderzoek blijkt vooralsnog echter weinig concrete steun voor de hypothese dat OSS (inherent) veiliger is dan CSS. Diverse

Datu
30 ju
2022

5.1(2e) Deze aanleiding zou een heel mooie basis voor de management samenvatting vormen! Als dit stuk samengevoegd wordt met de management samenvatting hierboven, dan zou dat een goede beschrijving van het wat, waarom en hoe vormen.

We adviseren om in de management samenvatting hoogover te beginnen zodat bestuurders de aanleiding, problematiek en noodzaak zien om hier binnen de organisatie mee aan de slag te gaan. Vervolgens zou op het eind van de management samenvatting benoemd kunnen worden dat dit stuk een meer gedetailleerde uiteenzetting en handreiking bevat voor beleidsmakers en IT-professionals om te helpen bij een verdere en concretere uitwerking.

5.1(2e) Dit is een heel mooie opening om de management samenvatting mee te beginnen!

5.1(2e): Wederom een sterke formulering om in de management samenvatting hierboven te zetten. Als de doelgroep bijvoorbeeld bestuurders zijn, dan kan de vraag 'Waarom Moet Ik Dit NU Doen?' in een tekst helpen om tot actie aanzetten. We willen dat mensen in beweging komen naar aanleiding van dit stuk.

Na 'belang' nog een zin toevoegen: waarom is het adequaat kunnen beheersen van deze

5.1(2e)]: We adviseren om verderop in het stuk, waar meer in detail wordt ingegaan op OSS en CSS, een overzicht (bijvoorbeeld een kwadrant) te maken met daarin deze varianten beschreven, de bijbehorende voor- en nadelen en risico's (e.g. continuiteit, aansprakelijkheid, patchbaarheid, vendor lock-in) en aansluitend adviezen. Dit maakt het voor een lezer in één oogopslag visueel duidelijk wat de verschillende varianten, risico's en aanbevelingen zijn.

5.1(2e) We adviseren om dit onderdeel de aanleiding te laten zijn als de bovenstaande aanleiding en management samenvatting samengevoegd worden.

Onderaan kan dan bijvoorbeeld nog afsluitend een punt toegevoegd worden als:

- Zowel OSS als CSS kennen beiden voor – en nadelen. Omdat het de ambitie is om overheidsbreed meer OSS in te zetten wordt

onderzoeken[1] laten zien dat er in de praktijk weinig verschil zit in het aantal kwetsbaarheden tussen enerzijds OSS en anderzijds CSS. De kwaliteit van de code lijkt vooral te worden bepaald door professionaliteit van het ontwikkelteam en de manier waarop zij code ontwikkelen, los van of dit OSS of CSS is. Uit recent onderzoek blijkt dat het overgrote deel van de OSS ten minste één of meerdere bekende kwetsbaarheden bevat; de helft bevat één of meer kwetsbaarheden waarvan bekend is dat die actief misbruikt (kunnen) worden. Het gemiddelde aantal kritieke kwetsbaarheden per OS-project varieert tussen de 2.6 voor .Net codes en 9.5 voor OSS in Java. Het totale aantal kwetsbaarheden per OS-project, inclusief de minder ernstige, ligt een stuk hoger, namelijk op gemiddeld 23 voor OS in .Net en 90 voor OS in Java.

- Het feit dat de code van OSS voor eenieder inzichtelijk is, leidt in de praktijk dus niet noodzakelijkerwijs tot minder kwetsbaarheden. Mogelijk heeft dit ermee te maken dat het open en transparant maken van code nog niet automatisch betekent dat die code ook daadwerkelijk door anderen systematisch wordt gereviewd en verbeterd. Een andere reden, die verderop nader wordt toegelicht, heeft ermee te maken dat sommige OSS die wordt (her)gebruikt al voor langere tijd niet meer is bijgewerkt en geüpdate. In de tussentijd bekend geworden kwetsbaarheden zijn niet gepatcht en zijn blijven bestaan. Uit recent onderzoek blijkt dat 88% van de geanalyseerde OSS, onderdelen bevatte die 2 jaar of langer daarvoor voor het laatst waren geüpdate.
- Het open en transparant maken van OSS brengt daarnaast zelf overigens ook een aantal risico's met zich mee. Het maakt het voor kwaadwillenden bijvoorbeeld eenvoudiger om zelf (nog onbekende) kwetsbaarheden in de OSS van de overheid te vinden en te misbruiken, in plaats van die kwetsbaarheden te melden. De (vaak collectieve) wijze waarop OSS wordt ontwikkeld en gedistribueerd kan het ook eenvoudiger maken voor kwaadwillenden om gericht en al dan niet verhuld nieuwe kwetsbaarheden of malware toe te voegen aan de OSS-code. In het geval van CSS is de productie- en distributieketen van software meer afgeschermd en afgesloten voor anderen dan het kern ontwikkelteam en doet dit risico zich minder voor. Het recente voorbeeld van node.ipc laat ook zien dat het gericht toevoegen van kwaadaardige code aan OSS geen theoretisch risico is, maar één met in potentie verstrekkende gevolgen. Andere voorbeelden zijn die van Linux Mint (2016) en Linux Gentoo (2018). De problematiek rondom de Solarwinds hack laat zien dat CSS hier overigens ook niet helemaal immuun voor is. Het risico dat kwaadwillende misbruik maken van het open karakter van OSS door zelf malware toe te voegen aan de source code is in zekere mate te beheersen door eisen te stellen aan code signing en het gebruik van multifactor authenticatie (MFA) door ontwikkelaars.
- Transparantie kan daarnaast nog een ander meer algemeen risico met zich mee brengen wanneer de open source code betrekking heeft op technische voorzieningen voor defensie doeleinden, vitale infrastructuur of nationale veiligheid. Vreemde mogendheden kunnen dan via de OSS-code direct inzicht verkrijgen in technische capabilities en (on)mogelijkheden van (overheids)organisaties die in deze domeinen actief zijn. Dit kan betekenen dat inzet van OSS hier onwenselijk is of in ieder geval inzet van specifieke aanvullende maatregelen vereist.

[1] Fagundez et al, 2018, A literature review about the difference in security for open source and proprietary source software – and its influence in Open Science, Proceedings Mere 2018 Conference, Barcelona.
Schryen, 2011, Is Open Source Security a Myth. CACM, 54:5, 130-139

**Het vraagstuk van continuïteit, support en security governance**

- Eén van de belangrijkste belemmerende factoren in het gebruik en de adoptie van OSS binnen de private en publieke sector heeft te maken met de continuïteitsrisico's die samenhangen met OSS. OSS wordt over het algemeen ontwikkeld en beheerd door een community van vrijwilligers, geïnteresseerden en gebruikers van de betreffende software. Het kan zijn dat gedurende een bepaalde periode een specifieke partij, zoals een overheidsorganisatie of bedrijf, de ontwikkeling en het beheer ervan (co)financiert of regisseert, maar ook dan zal vaak het streven zijn om de ontwikkelde OSS uiteindelijk te laten landen in een brede community die voor de verdere doorontwikkeling en het beheer zorg draagt. De community van bouwers en beheerders biedt die OSS gratis, en veelal in open source licentie, aan voor eenieder die die software wil gebruiken. De keerzijde hiervan is echter ook een zekere vrijblijvendheid. Gebruikers kunnen er namelijk niet op rekenen; er niet vanuit gaan, dat de betreffende OSS gedurende de gehele gebruiksperiode adequaat ondersteund en onderhouden blijft, of dat kwetsbaarheden en andere securityproblemen altijd tijdig verholpen en gepatched worden. Er kunnen zich forse problemen voordoen in een situatie waarin een bedrijf of overheidsorganisatie een bepaalde OSS-voorziening heeft geïmplementeerd voor een kritiek bedrijfsproces, om er vervolgens achter te komen dat de ontwikkel- en beheer community die zich over de OSS had ontfermd, uit elkaar is gevallen en eigenlijk weinig tijd meer steekt in het goede beheer en tijdig patchen van die OSS. De organisatie moet vervolgens zo snel mogelijk op zoek naar een alternatief voor de OSS-voorziening en deze, vanwege o.a. security risico's die voortkomen uit gebrekkig of onzeker onderhoud, snel uitfaseren.
- Het ontbreken van garanties, van de zekerheid dat OSS goed wordt onderhouden; dat er geen enkele partij echt aanspreekbaar is op, dan wel aansprakelijk is voor, adequaat onderhoud, maakt dat veel professionele organisaties terughoudend zijn met het gebruik van OSS, zeker in kritieke bedrijfsprocessen. De mate waarin dit risico speelt hangt af van de robuustheid, omvang en professionaliteit van de betrokken OSS community en of er bijvoorbeeld een partij of bedrijf is die zich op enigerlei wijze heeft ontfermd over de betreffende OSS ontwikkeling. Uit recent onderzoek van de Linux Foundation blijkt bijvoorbeeld dat bij een aanzienlijk deel van de OSS projecten de software wordt ontwikkeld en beheerd door een kleine groep, en soms één of enkele, ontwikkelaars. Dit maakt die OSS-projecten kwetsbaar, niet alleen vanuit continuïteitsoverwegingen maar ook omdat bij kleinere teams de kans groter is dat er minder volgens professionele standaarden wordt gewerkt en kwetsbaarheden minder snel worden verholpen. Alvorens een OSS-voorziening te implementeren is het daarmee verstandig een beeld te krijgen van de omvang, samenstelling en werkwijze van de betrokken community en om tijdens het gebruik ook te blijven monitoren hoe de community activiteiten zich ontwikkelen. Het gemiddeld aantal repo commits per maand kan een goede graadmeter zijn voor de omvang en activiteit van de community door de tijd heen. Blijf weg bij OSS die al langere tijd niet is geüpdate. De kans is groot dat kwetsbaarheden niet tijdig zijn en worden verholpen.
- Het gebrek aan SLA's en bijbehorende zekerheden is ook de reden waarom bedrijven en (overheids)organisaties er vaak voor kiezen om OSS alleen in bedrijfskritische processen in te zetten wanneer daaraan een gedegen (betaald) support contract is te koppelen met SLA's. Dit is bijvoorbeeld het geval bij RedHat Linux. Veel OSS kent dergelijke beheercontracten niet. In

Datu 30 ju 2022

5.1(2e) Op sommige plekken wordt aansluitend aan de vraagstukken een advies gegeven en op sommige plekken weer niet.

We adviseren, zoals bovenstaand genoemd, om een overzicht van de verschillende OSS en CSS variaties te maken, de voor - en nadelen, de risico's en bijbehorend de adviezen. Dan kunnen hier eerst de vraagstukken en aandachtspunten geschetst worden die meer duiding geven aan dit overzicht wat daarna een opsomming zou geven.