Overzicht

Dit document is zowel een analyse en uitleg over de WordPress core software ontwikkeling en zijn gerelateerde veiligheidsprocessen, als een onderzoek naar de inherente veiligheid die direct in de software is ingebouwd. Individuen die WordPress als een content management systeem of als een web applicatie framework evalueren kunnen dit document gebruiken voor hun analyse en hun keuzeproces. Ontwikkelaars kunnen dit document gebruiken wanneer ze zich willen verdiepen in de veiligheidscomponenten en best practices van de software.

De informatie in dit document is up-to-date met de meest recente en stabiele versie van de software; WordPress 4.1. Dit betekent echter niet dat het niet relevant is voor de minder recente versies van de software, dit vanwege de sterke focus vanuit het WordPress development team op achterwaartse compatibiliteit. Bepaalde veiligheidsmaatregelen en veranderingen die aan de core software in specifieke releases toegevoegd zijn zullen genoteerd worden. Het is aan te raden om altijd de laatste stabiele versie van WordPress te gebruiken om zo een zo veilig mogelijke ervaring te garanderen.

Management Samenvatting

WordPress is een dynamisch open-source content management systeem dat voor miljoenen websites, web applicaties en blogs gebruikt wordt. Het onderhoudt op dit moment 23% van de top 10 miljoen websites op het web. De gebruiksvriendelijkheid van WordPress, de mogelijkheid tot uitbreiding en zijn geavanceerde en gevorderde development community maakt het een populaire en veilige keuze voor veel verschillende websites.

Sinds de oprichting in 2003, wordt WordPress zijn core software steeds meer aangesterkt. Dit onderhoud is nodig om de software bestand te houden tegen veel voorkomende veiligheidsbedreigingen. Hieronder valt ook de top 10 met meest voorkomende veiligheidsbedreigingen van ‘The Open Web Application Security Project (OWASP) die later in dit document verder besproken zullen worden.

Het beveiligingsteam van WordPress, het WordPress core leadership team en de global community van WordPress werken allemaal samen om de mogelijke veiligheidsbedreigingen van de core software die gebruikt wordt voor distributie en installatie op WordPress.org, te identificeren en op te lossen. Daarnaast houden de teams zich bezig met het aanbevelen en documenteren van best practices omtrent veiligheid voor externe plug-ins en thema auteurs.

Om veel voorkomende kwetsbaarheden tegen te gaan moeten website developers en beheerders extra rekening houden met het correct gebruiken van core APIs, met de onderliggende server configuratie en ze moeten ervoor zorgen dat gebruikers sterke wachtwoorden voor WordPress implementeren.

Een overzicht van WordPress

WordPress is een gratis open source content management systeem (CMS). Het is de meest gebruikte CMS software ter wereld en onderhoudt meer dan 23% van de top 10 miljoen websites op het web.[[1]](#footnote-1) Dit geeft het naar schatting een marktwaarde van 60% over alle websites die een CMS gebruiken.

WordPress valt onder de General Public License (GPLv2 of later) die vier core ‘rechten’ vrijgeeft. Deze worden gezien als de WordPress “Bill of rights”:

1. Het recht om het programma voor welk doeleinde dan ook te gebruiken.
2. Het recht om het programma te bestuderen en aan te passen waar (persoonlijk) nodig.
3. Het recht tot (her)distributie.
4. Het recht tot distributie van persoonlijk gemodificeerde versies.

**Het WordPress Core Leadership Team**

Het WordPress project is een meritocratie dat gerund wordt door een core leadership team en door zijn mede-bedenker (co-creator) en hoofd developer Matt Mullenweg. Het team overziet alle aspecten van het project. Hier horen onder andere core development, WordPress.org en community initiatieven bij.

Het core leadership team bestaat uit Matt Mullenweg, vijf hoofd developers en meer dan tien core developers met permanente toegang. Deze developers hebben het laatste woord als het gaat om technische beslissingen en ze leiden architecture discussies en implementatie efforts(?)

WordPress heeft meerdere developers die bijdragen aan de ontwikkeling van de software. Developers die daadwerkelijk de code mogen wijzigen en toevoegen aan het project zijn de ‘committers’. Sommige van hen zijn voormalige committers, andere zijn huidige committers en weer andere zijn mogelijk toekomstige committers. Deze developers zijn betrouwbare en ervaren ontwikkelaars van WordPress die daardoor veel respect van hun naasten(/collega´s) (peers) ontvangen. WordPress heeft ook gast committers, dit zijn individuen die tijdelijk toegang krijgen tot bijvoorbeeld een specifiek component om daar mee aan het werk te gaan.

De core- en bijdragende developers zijn voornamelijk verantwoordelijk voor de ontwikkeling van WordPress. Bij elke versie dragen honderden developers code bij aan WordPress. Deze core bijdragers zijn vrijwilligers die op verschillende manieren hun bijdrage leveren aan de core codebasis.

**De WordPress Release Cyclus**Elke WordPress release cyclus wordt door één of meerdere core WordPress developers geleid. Een release cyclus duurt meestal vanaf de eerste meeting (scoping) tot de lancering ongeveer 4 maanden.

Een release cyclus doorloopt de volgende fases[[2]](#footnote-2):

* Fase 1: Het plannen en vaststellen van teambegeleiders. Dit gebeurt in de #core chat room op Slack. De ‘releasebegeleider’ bespreekt features voor de volgende release van WordPress. WordPress bijdragers worden in deze discussie meegenomen. De releasebegeleider zal voor elke feature een aparte begeleider aanstellen.
* Fase 2: Het begin van de ontwikkelingsfase. De teambegeleiders verzamelen teams en werken aan hun desbetreffende features. Regelmatige communicatie verzekerd een goed doorlopende ontwikkeling.
* Fase 3: Beta. Betas worden uitgegeven en beta-testers worden gevraagd om op zoek te gaan naar bugs. Bijdrages voor verbeteringen of feature aanvragen worden vanaf dit punt niet meer meegenomen. Third party plug-ins en thema auteurs worden aangemoedigd om hun code aan de hand van de nieuwe veranderingen te testen.
* Fase 4: Release Candidate(?) Vanaf dit punt worden vertaalbare strings niet meer veranderd. Het werk richt zich uitsluitend op regressies en blockers.
* Fase 5: Lancering. De WordPress versie wordt gelanceerd en beschikbaar gemaakt in de WordPress admin voor updates.

**Versie nummering en Veiligheidsreleases**

Een grote WordPress versie wordt gespecificeerd door het gebruik van twee sequences. (A major WordPress version is dictated by the first two sequences?). Bijvoorbeeld, 3.5 is een grote release, 3.6, 3.7 en 4.0 ook. Er bestaat geen ‘WordPress 3’ of ‘WordPress 4’. Elke grote release wordt aangeduid aan de hand van zijn eigen nummering, bijv. “WordPress 3.9”.

Grote releases kunnen nieuwe gebruikersfeatures en developer APIs bevatten. Een grote release wordt echter over het algemeen in de software wereld gezien als een manier om achterwaartse compatibiliteit te verbreken. WordPress streeft ernaar om dit nooit te hoeven doen. Achterwaartse compatibiliteit is een van de belangrijkste filosofieën van WordPress, met als doel om updates een stuk makkelijker te maken voor zowel gebruikers als ontwikkelaars.

Een kleinere WordPress versie wordt gespecificeerd door het gebruik van een derde sequence. Versies als 3.5.1 en 3.4.2 zijn kleinere releases. Kleinere versies worden alleen gebruikt om bedreigingen in veiligheid en gevaarlijke bugs mee te specificeren en op te lossen. Omdat er zo vaak nieuwe WordPress versies worden uitgebracht – het doel is, om de 4-5 maanden voor een grote release en kleinere releases wanneer nodig – zijn er alleen grote en kleinere releases nodig.

**Versie Achterwaartse Compatibiliteit**

Het WordPress project hecht veel waarde aan achterwaartse compatibiliteit. Achterwaartse compatibiliteit betekend dat thema’s, plug-ins en gepersonaliseerde code nog steeds werken wanneer de core software van WordPress ge-update wordt. Dit spoort website beheerders aan om hun WordPress versie up-to-date te houden met de laatste veilige release.

WordPress en Beveiliging

**Het WordPress beveiligingsteam**

Het WordPress beveiligingsteam bestaat uit ongeveer 25 experts, waaronder hoofd developers en beveiliging researchers. Ongeveer de helft hiervan zijn werknemers van Automattic ( de makers van WordPress.com, het eerste en grootste hosting platform op het web.) Een ander deel werkt in het web beveiligingsveld. Het team is continue in overleg met wel bekende en vertrouwde beveiligingsonderzoekers en hosting bedrijven[[3]](#footnote-3).

Het WordPress beveiligingsteam werkt vaak met andere beveiligingsteams samen om gedeelde zwaktepunten aan te pakken zoals het PHP XML parser(?) dat door de XML-RCP API gebruikt wordt en met WordPress meelift in WordPress 3.9.2.[[4]](#footnote-4) (the PHP XML parser, used by the XML-RPC API that ships with WordPress, in WordPress 3.9.2). Deze oplossing was het resultaat van een succesvolle samenwerking tussen het Drupal en het WordPress team.

**WordPress Veiligheidsrisico’s, Processen, en Geschiedenis**

Het WordPress team gelooft in ‘Responsible Disclosure’, oftewel verantwoordelijke openbaring. Hierbij wordt het beveiligingsteam meteen geïnformeerd wanneer er een mogelijk veiligheidsrisico optreedt. Mogelijke veiligheidsrisico’s kunnen meteen via [security@wordpress.org](mailto:security@wordpress.org)[[5]](#footnote-5) doorgemaild worden. Het beveiligingsteam communiceert onderling door middel van gesloten emaillijsten. Voor het traceren, testen en oplossen van bugs en beveiligingsproblemen maken ze gebruik van een ‘walled-off’, privé ‘Trac’.

Elke veiligheidsmelding (security report) die binnen komt wordt erkent en meegenomen door het beveiligingsteam om zo de juiste verificatie door te kunnen voeren en om te kijken hoe zwaar de bedreiging is. Wanneer dit vastgesteld is plant het beveiligingsteam een moment om het probleem op te lossen. Dit kan meegenomen worden in een eerstvolgende release van de WordPress software of het kan meteen als een beveiligingsrelease uitgegeven worden. Dit is afhankelijk van de zwaarte (severity) van het probleem.

Bij een directe beveiligingsrelease wordt er door het team ook een handleiding uitgegeven. Deze handleiding wordt via de WordPress.org News site[[6]](#footnote-6) gedeeld met het nieuws over de release en de details omtrent de veranderingen. Een dankwoord voor de verantwoordelijke openbaring van een bedreiging wordt in de handleiding vermeld om op deze manier meer individuen aan te moedigen om in de toekomst mogelijke bedreigingen te melden.

Beheerders van de WordPress software zien een melding op het dashboard van hun website wanneer er een nieuwe versie is uitgebracht en er dus een upgrade doorgevoerd kan worden. Na de manual upgrade worden de gebruikers doorgestuurd naar de ‘About WordPress’ pagina waar de details omtrent de veranderingen vermeld staan. Wanneer beheerders de functie ‘automatic background updates’, ofwel automatische achtergrond updates aan hebben staan ontvangen ze een email nadat de upgrade is doorgevoerd.

**Automatische Achtergrond Updates voor Beveiligingsreleases**

Vanaf versie 3.7 kwam WordPress met de automatische achtergrond update voor alle kleinere releases[[7]](#footnote-7), zoals 3.7.1 en 3.7.2. Het beveiligingsteam van WordPress kan geautomatiseerde beveiligingsverbeteringen identificeren, repareren en doorvoeren zonder dat de website beheerder iets hoeft te doen. De beveiligingsupdate installeert zich automatisch.

Wanneer een beveiligingsupdate voor de meest recente release van WordPress wordt doorgevoerd, dan zal het core team voor alle releases die de achtergrond update functie ondersteunen ook een beveiligingsupdate doorvoeren. (Sinds WordPress 3.7). Op deze manier ontvangen oudere, maar nog steeds recente versies van WordPress ook de beveiligingsverbeteringen.

Particuliere website eigenaars kunnen in hun configuratiebestand de automatische achtergrond update functie uitzetten. Het is echter sterk aanbevolen om de functie aan te hebben staan om de meest recente release van WordPress te gebruiken.

**2013 OWASP Top 10**

De Open Web Application Security Project (OWASP) is een online community die zich bezig houdt met web applicatie beveiliging. De OWASP top 10 lijst[[8]](#footnote-8) richt zich op het identificeren van de gevaarlijkste veiligheidsrisico’s voor een grote groep organisaties. De top 10 items worden geselecteerd en geprioriteerd in combinatie met schattingen naar overeenkomsten van exploiteerbaarheid, aantoonbaarheid en het effect van impact.

Het volgende onderdeel zal ingaan op APIs, middelen en het beleid dat WordPress voert om de core software, third party plug-ins en thema’s te versterken en beschermen tegen mogelijke veiligheidsrisico’s.

**A1 - Injection**

In WordPress zijn er verschillende APIs en functies beschikbaar die developers beschermen tegen onbevoegde code en die helpen bij het valideren en ‘ontsmetten’ van data. Er zijn best practices en documentatie beschikbaar[[9]](#footnote-9) over het gebruik van APIs voor de bescherming, validering of ontsmetting van in- en output data in HTML, URLs en HTTP headers en over de interactie met de database en het filesysteem. Beheerders kunnen ook restricties op bepaalde type files leggen die via filters ge-upload kunnen worden.

**A2 - Broken Verificatie en Sessie Management**

WordPress zijn core software beheert gebruikersaccounts, verificaties en details. Specificaties zoals een user ID, naam, wachtwoord en verificatie cookies (authentication cookies) worden beheert op de server-side (on the server side). Wachtwoorden worden in de database beschermd door middel van standaard ‘salting en stretching’ technieken. Bestaande sessies worden verwijderd bij het uitloggen bij WordPress versies ouder dan versie 4.0.

**A3 - Cross Site Scripting (XSS)**

WordPress biedt een bepaald aantal functies aan die assisteren bij het veilig stellen van user-supplied data.[[10]](#footnote-10) ‘Vertrouwde’ gebruikers, zoals beheerders en editors op een enkele WordPress installatie en sitebeheerders van WordPress multisites kunnen ongefilterd HTML of Javascript in een post of op een pagina posten. ‘Niet vertrouwde’ gebruikers en content die door gebruikers is aangeleverd wordt standaard gefilterd om mogelijk gevaarlijke entiteiten te verwijderen. Dit wordt met behulp van de KSES bibliotheek en aan de hand van van de wp\_kses functie gedaan.

Om een voorbeeld te noemen, het WordPress core team kwam er vóór de release van WordPress 2,3 achter, dat de functie the\_search\_query() door de meeste thema auteurs, die de functie output voor het gebruik in HTML niet konden ontsnappen, misbruikt werd(As an example, the WordPress core team noticed before the release of WordPress 2.3 that the functionthe\_search\_query() was being misused by most theme authors, who were not escaping the function's output for use in HTML.)

Als uitzondering op de regel met betrekking tot achterwaartse compatibiliteit werd de output van de functie in WordPress 2.3 veranderd to be pre-escaped.?

**//A4 - Insecure Direct Object Reference// 🡪 Weet hier te weinig vanaf om het goed te vertalen**

WordPress often provides direct object reference, such as unique numeric identifiers of user accounts or content available in the URL or form fields. While these identifiers disclose direct system information, WordPress' rich permissions and access control system prevent unauthorized requests.

**A5 – Beveiliging Misconfiguratie**

De meerderheid van de WordPress beveiligingsconfiguratie operaties/ondernemingen/aspecten(?)(operations) worden beperkt tot een enkele gemachtigde beheerder. Standaard WordPress settings worden continue door het WordPress core team geëvalueerd. Daarnaast levert het core team ook documentatie en best practices op om zo de beveiliging voor de server configuratie van een lopende website te verbeteren.[[11]](#footnote-11)

**A6 - Gevoelige Data Exposure**

De wachtwoorden van de WordPress gebruikersaccounts zijn ‘salted en hashed’ op basis van het portable PHP Password Hashing Framework.[[12]](#footnote-12) WordPress zijn toestemmingssysteem (permission system) wordt gebruikt om de toegang tot privé informatie zoals geregistreerde gebruikers PII’s, email adressen van mensen die comments achterlaten en privé gepubliceerde content te bepalen. In WordPress 3.7 werd er een wachtwoord sterktemeter in de core software toegevoegd. Deze sterktemeter hielp gebruikers bij het instellen van hun wachtwoorden aan de hand van hints en aan de hand van extra informatie. WordPress heeft ook een optionele configuratie setting voor requiring(?) HTTPS.

**A7 - Ontbrekende Functie Level Access Control**

WordPress controleert altijd of de autorisatie en toestemmingen (permissions) van welke functie level toegang aanvraag dan ook kloppen, voordat deze actie wordt uitgevoerd. (WordPress checks for proper authorization and permissions for any function level access requests prior to the action being executed.) Toegang en de mogelijkheid tot inzage van administratieve URL’s, menu’s en pagina’s zijn nauw verbonden met en goed geïntegreerd in het authenticatie systeem om zo de toegang van ongewenste (unauthorized) gebruikers tegen te gaan.

**A8 - Cross Site Request Forgery (CSRF)**

WordPress maakt gebruik van cryptografische penningen (tokens?), ookwel nonces[[13]](#footnote-13) genoemd, om de intentie van de aanvragen van de geautoriseerde gebruikers te valideren en zo deze te beschermen tegen mogelijke CSRF bedreigingen. WordPress stelt APIs beschikbaar voor het genereren, creëren en verifiëren van tijdelijke penningen. Deze penningen behoren tot een specifieke gebruiker, een specifieke actie en een specifieke tijdperiode. Deze kunnen zo nodig aan de formulieren en URLS worden toegevoegd. Alle nonces worden ongeldig verklaard bij het uitloggen.

**A9 – Componenten met bekende kwetsbaarheden gebruiken**

De kleine hoeveelheid bibliotheken en frameworks die door WordPress in de core functionaliteiten geïntegreerd zijn, worden door het team zorgvuldig gemonitord. In het verleden heeft het core team meerdere bijdragen aan third party componenten geleverd om deze veiliger te maken, bijvoorbeeld de update om een cross-site zwakte in TinyMCE in WordPress 3.5.2 te repareren.[[14]](#footnote-14)

Wanneer het team het nodig acht, mag het core team besluiten om kritieke externe componenten te ‘forken’ of weg te halen. Een voorbeeld hiervan is de SWF Upload bibliotheek. De SWF Upload bibliotheek is officieel vervangen door de Plupload in 3.5.2 en een veilige fork van SWF Upload werd beschikbaar gesteld door het beveiligingsteam[[15]](#footnote-15) voor de plug-ins die nog steeds op korte termijn gebruik maakten van SWF Upload.

**A10 - Ongevalideerde Redirects en Forwards**

WordPress zijn interne access control (access control) en authenticiteit systeem bieden bescherming tegen de pogingen die gedaan worden om gebruikers naar ongewilde bestemmingen te sturen en tegen automatische redirects (?). Deze functionaliteit wordt ook beschikbaar gesteld voor plug-in ontwikkelaars via de API, wp\_safe\_redirect()[[16]](#footnote-16)

**Verdere veiligheidsrisico’s en bezorgdheden**

**XXE (XML eXternal Entity) processing aanvallen**

Wanneer XML geprocessed wordt, schakelt WordPress het laden van custom XML entiteiten uit om zo External Entity en Entity Expansion aanvallen tegen te gaan. Los van de PHP core functionaliteit, levert WordPress geen extra veilige XML processing API voor plug-in auteurs.

**SSRF (Server Side Request Forgery) aanvallen**

HTTP aanvragen van WordPress worden gefilterd om zo de toegang tot loopback en privé IP adressen tegen te gaan. Daarnaast wordt toegang alleen mogelijk gemaakt voor bepaalde standaard http portalen (ports?).

WordPress Plugin and Thema Beveiliging

**Het Default Thema**

WordPress heeft als vereiste dat een thema ingeschakeld enabled (?) moet zijn om content die visueel zichtbaar aan de voorkant is, te renderen (?). Het default thema dat meegezonden wordt met core WordPress (op dit moment, “Twenty Fifteen”) is intensief gereviewed en getest voor veiligheidsredenen door en het team van thema developers en het core development team.

Het default thema kan dienen als een beginpunt voor custom theme development en biedt de mogelijk voor site developers om een child theme te maken waar wat aangepaste aspecten in kunnen zitten, maar die alsnog op het default thema terug kunnen vallen voor de meeste functionaliteiten en beveiliging. Het default thema kan gemakkelijk door een beheerder verwijderd worden als deze niet nodig is.

**WordPress.org Thema en Plugin Repositories**

Er staan ongeveer 30,000 plug-ins en 2,000 thema’s op de WordPress.org site vermeld. Deze thema’s en plug-ins worden vóór opname in de software handmatig door vrijwilligers gereviewed voordat ze beschikbaar worden gemaakt in de repository.

De opname van plug-ins en thema’s in de repository betekend niet automatisch dat deze 100% veilig zijn. Richtlijnen worden vrijgegeven zodat plug-in auteurs deze vóór de daadwerkelijke opname in de repository kunnen raadplegen [[17]](#footnote-17). Daarnaast wordt er op de WordPress.org site ook uitgebreide documentatie beschikbaar gesteld over WordPress thema ontwikkeling[[18]](#footnote-18).

Elk plug-in thema heeft de mogelijkheid om continue verder ontwikkeld te worden door de plug-in of thema eigenaar. Latere reparaties of feature ontwikkelingen kunnen ge-upload worden naar de repository en beschikbaar gesteld worden - met een omschrijving van de veranderingen - voor gebruikers die de desbetreffende plug-in of het desbetreffende thema hebben geïnstalleerd.

Websitebeheerders ontvangen een notificatie wanneer een plug-in geupdate moet worden via hun beheerdersdashboard.

Wanneer er een beveiligingsrisico door het WordPress security team gevonden wordt, dan wordt de plug-in auteur benaderd en wordt er samen gewerkt om reparaties te maken en om een veiligere versie van de plug-in uit te brengen. Wanneer de plug-in auteur niet reageert of wanneer het beveiligingsrisico dusdanig groot is, dan wordt de plug-in/ het thema uit de publiekelijke directory gehaald en in sommige gevallen gerepareerd en meteen ge-update door het beveiligingsteam.

**Het Thema Review Team**

Het Thema Review Team is een groep vrijwilligers die begeleid worden door belangrijke leden van de WordPress community. Zij reviewen en keuren thema’s goed die voor opname in het officiële WordPress thema directory aangedragen worden. Het Thema Review Team onderhoudt de officiële Thema Review Guidelines[[19]](#footnote-19), de Thema Unit Test Data[[20]](#footnote-20) en de Thema Check Plug-in[[21]](#footnote-21) en probeert om de WordPress developer community te stimuleren en onderwijzen omtrent development best practices. Opname in de groep wordt gemodereerd door core committers van het WordPress development team.

De Rol van de Hosting Provider in WordPress Beveiliging

WordPress kan op een hoop verschillende platforms geïnstalleerd worden. Naast de meerdere bepalingen(?) provisions van de WordPress core software voor het werken met veilige web applicaties, zijn de configuraties van het operating systeem en de onderliggende web server die de software host, even belangrijk om de WordPress applicaties beveiligd te houden.

**Een aantekening over WordPress.com en WordPress beveiliging**

WordPress.com is de grootste WordPress installatie ter wereld en is eigendom van en wordt geleid door Matt Mullenweg, de eigenaar van Automattic,Inc. Wordpress.com runt op de core WordPress software en heeft zijn eigen beveiligingsprocessen, risico´s en oplossingen.[[22]](#footnote-22) Dit document refereert naar beveiliging omtrent de self-hosted, te downloaden open source WordPress software die beschikbaar wordt gesteld door WordPress.org en overal ter wereld geïnstalleerd kan worden.

Bijlage

**Core WordPress APIs**

De WordPress Core Application Programming Interface (API) is opgemaakt uit verschillende individuele APIs[[23]](#footnote-23). Deze APIs houden zich bezig met alle functies die gebruikt worden door, of betrokken zijn bij, een bepaalde set functionaliteiten. Samen vormen deze APIs de project interface die het mogelijk maakt voor plug-ins en thema’s om samen te werken, veranderd te worden en om de WordPress core functionaliteit veilig uit te kunnen breiden.

Naast dat elke WordPress API best practices en gestandaardiseerde methodes voor interactie en de uitbreiding van WordPress core software aanbiedt, zijn de volgende WordPress APIs het meest relevant voor het aansturen en voor de verharding van WordPress beveiliging:

**Database API**

De Database API[[24]](#footnote-24) die in WordPress 0.71 is toegevoegd, zorgt voor de juiste methode voor het ophalen van data als named values die in de database layer zijn opgeslagen.

**Filesystem API**

De Filesystem API[[25]](#footnote-25) die in WordPress 2.6[[26]](#footnote-26) is toegevoegd was in de eerste instantie ontwikkeld voor WordPress zijn eigen automatische update features. De Filesystem API selecteert (abstract out) de functionaliteit die nodig is voor het - op een veilige manier en op een verschillende hoeveelheid host types- lezen en schrijven van local files richting het file systeem.

Het doet dit aan de hand van de WP\_Filesystem\_Base  class, en meerdere subclasses die verschillende manieren van verbinding, afhankelijk van individuele host support, naar het lokale filesystem implementeren. Elk thema of elke plug-in die files lokaal moet schrijven zou het WP\_Filesystem family of classes moeten gebruiken.

**HTTP API**

De http API[[27]](#footnote-27) die in WordPress 2.7[[28]](#footnote-28) is toegevoegd en vervolgens verder uitgebreid is naar WordPress 2.8 standaardiseert de http aanvragen voor WordPress. De API regelt de cookies, gzip encoding en decoding, chunk decoding (indien http 1.1) en nog meer andere http protocol implementaties. De API standaardiseert aanvragen, test elke methode voor verzending, en gebruikt, gebaseerd op de server configuratie, de passende methode voor een aanvraag.

**Permissions and current user API**

De permissions en current user API[[29]](#footnote-29) is een set functies die helpt bij de verificatie van de huidige gebruiker zijn toestemmingen en autoriteit om aangevraagde taken uit te voeren. Daarnaast kan het ook verdere bescherming bieden tegen ongeautoriseerde gebruikers die proberen om functies buiten hun toegestane vermogen proberen te bereiken en uit te voeren.

**White paper content License**

De tekst in dit document (los van het WordPress logo of trademark) is gelicentieerd onder CC0.1.0 UniversAL (cc0.1.0) Public Domain Dedication. Er mogen zonder toestemming, kopieën gemaakt worden, het mag gedistribueerd worden en het mag voor commerciële doeleinden gebruikt worden.

*Nogmaals dank voor Drupal zijn security whitepaper voor de inspiratie*

**Additional Reading**

* WordPress News <https://wordpress.org/news/>
* WordPress Security releases <https://wordpress.org/news/category/security/>
* WordPress Developer Resources <https://developer.wordpress.org/>

1. <http://w3techs.com/>, vanaf Maart 2015 [↑](#footnote-ref-1)
2. <https://make.wordpress.org/core/handbook/how-the-release-cycle-works/> [↑](#footnote-ref-2)
3. Andrew Nacin, WordPress lead developer, <http://vip.wordpress.com/security> [↑](#footnote-ref-3)
4. <http://wordpress.org/news/2014/08/wordpress-3-9-2/> [↑](#footnote-ref-4)
5. <http://codex.wordpress.org/Security_FAQ> [↑](#footnote-ref-5)
6. <http://wordpress.org/news/> [↑](#footnote-ref-6)
7. <https://wordpress.org/news/2013/10/basie/> [↑](#footnote-ref-7)
8. <https://www.owasp.org/index.php/Top_10_2013-Top_10> [↑](#footnote-ref-8)
9. <https://developer.wordpress.org/plugins/security/> [↑](#footnote-ref-9)
10. <http://codex.wordpress.org/Data_Validation#HTML.2FXML> [↑](#footnote-ref-10)
11. <http://codex.wordpress.org/Hardening_WordPress> [↑](#footnote-ref-11)
12. <http://www.openwall.com/phpass/> [↑](#footnote-ref-12)
13. <https://developer.wordpress.org/plugins/security/nonces/> [↑](#footnote-ref-13)
14. <https://wordpress.org/news/2013/06/wordpress-3-5-2/> [↑](#footnote-ref-14)
15. <https://make.wordpress.org/core/2013/06/21/secure-swfupload/> [↑](#footnote-ref-15)
16. <https://developer.wordpress.org/reference/functions/wp_safe_redirect/> [↑](#footnote-ref-16)
17. <https://wordpress.org/plugins/about/guidelines> [↑](#footnote-ref-17)
18. <https://developer.wordpress.org/themes/getting-started/> [↑](#footnote-ref-18)
19. <http://codex.wordpress.org/Theme_review> [↑](#footnote-ref-19)
20. <http://codex.wordpress.org/Theme_Unit_Test> [↑](#footnote-ref-20)
21. <https://wordpress.org/plugins/theme-check/> [↑](#footnote-ref-21)
22. <http://automattic.com/security/> [↑](#footnote-ref-22)
23. <https://codex.wordpress.org/WordPress_APIs> [↑](#footnote-ref-23)
24. <https://codex.wordpress.org?Database_API> [↑](#footnote-ref-24)
25. <https://codex.wordpress.org/Filesystem_API> [↑](#footnote-ref-25)
26. <http://codex.wordpress.org/Version_2.6> [↑](#footnote-ref-26)
27. <https://codes.wordpress.org/HTTP_API> [↑](#footnote-ref-27)
28. <https://codex.wordpress.org/Version_2.7> [↑](#footnote-ref-28)
29. [http://codex.wordpress.org/Function\_Reference/current­\_user\_can](http://codex.wordpress.org/Function_Reference/current_user_can) [↑](#footnote-ref-29)