Nationaal Cyber Security Centrum
Ministerie van Veiligheid en Justitie

Help! Mijn website is kwetsbaar voor SQL-injectie

Controleer uw website en tref maatregelen

Factsheet FS-2014-05
versie 1.0 | 9 oktober 2014

SQL-injectie is een populaire en veel toegepaste aanval op websites waarvan kwaadwillenden gebruik maken voor het buitmaken van grote hoeveelheden (klant)informatie. Hoewel voor het buitmaken van deze informatie ook andere typen aanvallen bestaan, blijkt SQL-injectie in veel gevallen het gebruikte middel.

Een SQL-injectiekwetsbaarheid ontstaat wanneer een kwaadwillende in staat is om de verzoeken die een website aan een database verstuurt te beïnvloeden. Hierdoor kan deze kwaadwillende via bijvoorbeeld een simpele zoekopdracht informatie aan de database onttrekken of de inhoud van de database wijzigen. Een SQL-injectiekwetsbaarheid kan op deze manier zowel de integriteit als de vertrouwelijkheid van de informatie achter de website in gevaar brengen.

In deze factsheet geven wij aan hoe u kunt handelen op het moment dat uw website kwetsbaar is voor SQL-injectie en succesvol is aangevallen. Daarnaast beschrijft de factsheet welke maatregelen u kunt treffen om ervoor te zorgen dat SQL-injectie u niet overkomt.

Doelgroep
Deze factsheet richt zich op ontwikkelaars en technische beheerders van websites. Bent u eigenaar van een kwetsbare website maar heeft u geen kennis van de techniek achter deze website? Stuur deze factsheet dan door naar uw ontwikkelaar of beheerder.

De belangrijkste feiten

- Een SQL-injectiekwetsbaarheid treedt op als gebruikersinvoer onveilig wordt gebruikt in de verzoeken die een website verstuurt aan de database.
- Via SQL-injectie kan een aanvaller willekeurige informatie aan een database onttrekken. In sommige gevallen kan hij ook informatie uit deze database verwijderen of juist informatie toevoegen.
- Slachtoffers van SQL-injectie raden wij aan om:
  - de schade en oorzaak van het incident te onderzoeken;
  - communicatie hierover voor te bereiden en uit te voeren;
  - tijdelijke maatregelen te treffen tijdens het onderzoek;
  - de integriteit van de database te herstellen;
  - de kwetsbaarheid weg te nemen door code aan te passen of de laatste patches van software te installeren, en
  - de website weer online te brengen zodra is vastgesteld dat de kwetsbaarheid afdoende is verholpen.
- SQL-injectie detecteren kan door:
  - logbestanden te controleren,
  - de integriteit van de database te controleren, en
  - publieke uitlatingen over uw website te monitoren.
- De kans op een SQL-injectiekwetsbaarheid kan geminimaliseerd worden door:
  - gebruik te maken van geparameteriseerde queries;
  - gebruikersinvoer te normaliseren, te valideren en te filteren op ongewenste invoer;
  - database-accounts met beperkte rechten te gebruiken;
  - gevoelige informatie in de database onleesbaar te maken;
  - de laatste updates van software te installeren;
  - voorzichtig te zijn met CMS-plug-ins van derden;
  - regelmatig beveiligingsscans uit te voeren, en
  - de mogelijkheid te bieden om op verantwoorde wijze kwetsbaarheden aan u te melden (responsible disclosure).

Achtergrond

Veel websites communiceren met achterliggende databases in een taal die Structured Query Language (SQL) heet. Websites maken gebruik van een database om daarin allerlei gegevens op te slaan. Denk bijvoorbeeld aan gebruikersnamen en wachtwoorden voor besloten gedeelten van de website of nieuwsberichten. Om met de

database te kunnen communiceren, formuleert de website een verzoek aan de database als een verzoek in SQL.

Wat is SQL-injectie?

SQL-injectie is het manipuleren van de structuur van SQL-verzoeken die de website aan de database richt via malafide gebruikersinvoer. Vaak kan een bezoeker van een website een SQL-verzoek direct of indirect beïnvloeden via bijvoorbeeld een zoekterm, een formulier of zelfs de waarde van een cookie of de string die de browser identificeert (de 'useragent'). Dit is geen probleem als de bezoeker alleen de inhoud van het verzoek kan beïnvloeden. Wanneer hij echter ook de structuur van het verzoek kan veranderen, biedt dat de bezoeker ongeautoriseerde toegang tot de database.

Stel bijvoorbeeld dat een organisatie via een website de mogelijkheid biedt om nieuwsberichten te doorzoeken op basis van een zoekterm. De ontwikkelaar van de website heeft daarvoor het volgende SQL-verzoek opgenomen in de code:

```
SELECT titel, omschrijving
FROM   nieuws
WHERE  omschrijving LIKE '%zoekterm%'
```

Dit SQL-verzoek haalt de titel en de omschrijving van nieuwsberichten op uit de database. Het verzoek beperkt het resultaat daarbij op basis van een door de bezoeker ingevoerde zoekterm. Het probleem is echter dat een bezoeker door het manipuleren van de zoekterm ook de structuur van het SQL-verzoek kan manipuleren. Stel dat een kwaadwillende de zoekterm '`zoekterm' UNION SELECT gebruikersnaam, wachtwoord FROM gebruikers; --`' aanbiedt aan de website. Het SQL-verzoek ziet er plotseling als volgt uit:

```
SELECT titel, omschrijving
FROM   nieuws
WHERE  omschrijving LIKE '%zoekterm'
UNION
SELECT gebruikersnaam, wachtwoord
FROM   gebruikers; --%'
```

Dit SQL-verzoek doet twee dingen. Ten eerste zorgt het ervoor dat nieuwsberichten nog steeds worden opgehaald. Ten tweede leidt het er toe dat de website een lijst van gebruikersnamen en wachtwoorden uit een ander deel van de database opvraagt. Kortom, de kwaadwillende heeft ervoor gezorgd dat de website uiteindelijk een andere vraag heeft gesteld aan de database dan eigenlijk de bedoeling was. Daarmee is het in dit voorbeeld mogelijk om via een simpele zoekopdracht alle gebruikersnamen en wachtwoorden uit de database lekken. Afhankelijk van de informatie die uw website opslaat in de database, kan hiermee ook andere informatie in handen komen van kwaadwillenden zoals creditcardnummers, medische informatie en documenten.

Wat kan er gebeuren?

SQL-injectie biedt de mogelijkheid om informatie uit de database op te vragen, te manipuleren en te verwijderen en, in sommige gevallen, hier informatie aan toe te voegen. De voorgaande paragraaf biedt hier een voorbeeld van. Daarnaast kan SQL-injectie gebruikt worden om de logica van de website te beïnvloeden waardoor bijvoorbeeld een authenticatiemechanisme wordt omzeild.

Hoe weet ik dat mijn website gehackt is?

U kunt achterhalen of uw website is aangevallen op basis van SQL-injectie door het monitoren van uw systemen en het monitoren van publieke uitlatingen over uw website. Daarnaast bestaat de mogelijkheid dat een externe organisatie of ontwikkelaar u op de hoogte brengt van de aanwezigheid van de kwetsbaarheid of het misbruik ervan. In het kader 'SQL-injectie detecteren' vindt u meer informatie over de monitoringsmechanismen die u kunt inrichten.

SQL-injectie detecteren

- *Controleer logbestanden op verdachte meldingen*
  Controleer of logbestanden van de webserver, de website, de applicatieserver, de database, het IPS/IDS, of de firewall verdachte zaken bevatten. Denk hierbij aan:
  - de aanwezigheid van SQL-commando's in URL's uit het webserverlog; denk aan commando's als SELECT, INSERT, DROP, UNION en UPDATE[1],
  - meldingen van het IDS over waargenomen pogingen tot SQL-injectie, en
  - foutmeldingen van de database zoals mislukte SQL-verzoeken uitgevoerd door de website.
- *Controleer de integriteit van de database*
  Controleer of de database verdachte zaken bevat zoals nieuwe, onbekende of vreemde gebruikersnamen of tabellen.
- *Monitor publieke uitlatingen over uw website*
  Zorg ervoor dat u via mechanismen als Google Alerts[2] en Twitter zoekopdrachten automatisch bericht krijgt over uitlatingen over uw website in combinatie met bijvoorbeeld 'dump', 'hack' of 'sql injection'.

Wat kan ik doen als ik gehackt ben?

Zodra uw website succesvol is aangevallen, betekent dit dat informatie uit de database in handen is gekomen van kwaadwillenden. Uiteraard is het niet mogelijk dit ongedaan te maken, maar het is wel zaak om direct stappen te ondernemen om de impact van de aanval zoveel mogelijk te beperken en misbruik in de toekomst te voorkomen.

[1] Let hierbij ook op gecodeerde versies van deze sleutelwoorden zoals '&#68&#82&#79&#80' als heximaal gecodeerde versie van het sleutelwoord 'DROP'. Zie hiervoor bijvoorbeeld de "Character Encoding Calculator" op http://ha.ckers.org/sqlinjection/

[2] https://www.google.com/alerts

Bij (het vermoeden van) een aanval op basis van SQL-injectie raden we aan de volgende maatregelen te treffen:

1. Onderzoek of er daadwerkelijk sprake is van een incident. Maak hiervoor gebruik van de detectiemogelijkheden uit het kader 'SQL-injectie detecteren' en voer een geautomatiseerde beveiligingsscan uit op de website[3] om zelfstandig SQL-injectiekwetsbaarheden te vinden.

2. Bepaal wat de impact is van het incident. Zijn er bijvoorbeeld 'alleen' email-adressen gelekt of ook wachtwoorden en andere gevoelige informatie?

Uit voorgaande stappen kan blijken dat er daadwerkelijk sprake is van een incident. Is dit het geval, onderneem dan de volgende stappen:

3. Bepaal wie u gaat inlichten, op welke manier u dat gaat doen en wat u gaat communiceren. Denk daarbij aan interne partijen zoals juridische zaken en de helpdesk, maar ook aan externe partijen zoals de bezoekers van uw website, klanten, leveranciers, afnemers, toezichthouders en de pers.

4. Bepaal wat u met uw website doet gedurende de afhandeling van het incident. Het is duidelijk dat u de website niet zonder meer in de lucht kunt houden. Mogelijke tijdelijke oplossingen zijn:
   - het online brengen van een tijdelijke statische website ter vervanging van de normale website, en
   - het uitschakelen van een deel van de functionaliteit van de website.

   Zet nooit zomaar een back-up terug van het gehele systeem of de database. De kans is namelijk groot dat u hiermee een kwetsbaar systeem terugzet dat daarna opnieuw gehackt wordt.

5. Herstel de integriteit van de database als blijkt dat aanvallers niet alleen informatie uit de database hebben ontvreemd maar daarnaast ook ongeautoriseerde wijzigingen op de database hebben doorgevoerd. Plaats bijvoorbeeld een back-up terug waarvan u zeker weet dat deze voor de hack is gemaakt.

6. Verhelp de kwetsbaarheid:

   - Maakt u gebruik van een standaard softwareproduct (zoals een CMS), installeer dan in ieder geval de laatste updates van deze software. Is het probleem daarin nog niet verholpen? Meld dit probleem dan zo snel mogelijk bij de leverancier. Deze kan dan alsnog een update uitbrengen.
   - Wanneer u de website zelf heeft ontwikkeld (of heeft laten ontwikkelen), dient u zelf aanpassingen aan uw website te (laten) doen om de kwetsbaarheid weg te nemen. Evalueer de complete website, niet alleen de aangevallen delen. Maak hierbij gebruik van de maatregelen uit de paragraaf 'Hoe kan ik SQL-injectie in mijn website voorkomen?'.

7. Laat een grondige penetratietest uitvoeren op de website. Laat hierbij in ieder geval kijken naar SQL-injectiekwetsbaarheden, maar liefst ook naar andere typen kwetsbaarheden. Verhelp de gevonden kwetsbaarheden.

8. Herstel de functionaliteiten van uw website zodra duidelijk is dat de kwetsbaarheden afdoende zijn verholpen.

Blokkeren en herstellen toegang

Indien ook inloggegevens van uw gebruikers zijn gelekt, is het van belang om niet alleen de gebruikers hierover in te lichten maar ook misbruik hiervan te voorkomen. Overweeg daarom om initieel de toegang voor gebruikers te blokkeren. Na het herstellen van de functionaliteiten kunt u vervolgens de mogelijkheid bieden om inloggegevens op een veilige wijze te wijzigen (bijvoorbeeld via e-mailverificatie) zodat gebruikers zich weer toegang kunnen verschaffen tot uw website.

Hoe kan ik SQL-injectie in mijn website voorkomen?

Uiteraard is het altijd beter om aanwezigheid van SQL-injectiekwetsbaarheden in uw website voorkomen. De 'ICT-Beveiligingsrichtlijnen voor webapplicaties' van het NCSC bevatten, naast maatregelen voor het voorkomen van SQL-injectiekwetsbaarheden, ook maatregelen voor het voorkomen van allerlei andere kwetsbaarheden. De onderstaande maatregelen, die voor het grootste deel afkomstig zijn uit deze richtlijnen, zijn van belang voor het voorkomen van SQL-injectiekwetsbaarheden (tussen haakjes vindt u de verwijzing naar de specifieke maatregel uit de richtlijnen):

- Maak gebruik van geparameteriseerde queries bij het opstellen van een SQL-verzoek (richtlijn B3-5). Dit houdt in dat een SQL-verzoek niet wordt opgebouwd door dynamisch allerlei strings aan elkaar te plakken, maar door een statisch SQL-verzoek te definiëren en hierin later parameters te 'plakken'. Alle moderne programmeertalen ondersteunen dit concept.

- Normaliseer en valideer alle invoer van gebruikers (richtlijnen B3-3 en B3-6). Dit houdt in dat de website controleert of de invoer die hij ontvangt overeen komt met de verwachte invoer. Zo moet een ingevoerde postcode altijd bestaan uit vier cijfers en twee letters. De website zou invoer die niet van die vorm is, bij voorbaat al niet moeten accepteren

[3] Er zijn diverse tools beschikbaar om geautomatiseerd vast te stellen op uw website een SQL-injectiekwetsbaarheid bevat. Een populair voorbeeld hiervan is sqlmap (http://sqlmap.org/).

De belangrijkste stappen bij SQL-injectie op een rijtje

1. Onderzoek of er daadwerkelijk sprake is van een incident.
2. Bepaal wat de impact van het incident is. Welke informatie is er gelekt?
3. Bepaal wie u gaat inlichten, op welke manier u dat gaat doen en wat u gaat communiceren.
4. Bepaal wat u met uw website doet gedurende de afhandeling van het incident. Schakel bijvoorbeeld tijdelijk functionaliteiten uit.
5. Herstel de integriteit van de database.
6. Verhelp de kwetsbaarheid door het installeren van updates of het (laten) wijzigen van eigen ontwikkelde software.
7. Laat een grondige penetratietest uitvoeren om ook andere potentiele kwetsbaarheden in uw website op te sporen.
8. Herstel de functionaliteiten zodra duidelijk is dat de kwetsbaarheden afdoende zijn verholpen.

» Filter op ongewenste invoer (richtlijn B3-1). Na het normaliseren en valideren van de invoer, controleert de website hierbij of er toch nog ongewenste invoer overblijft zoals bepaalde 'gevaarlijke' sleutelwoorden. Zo duidt het gebruik van de woorden SELECT en DROP in de invoer op een mogelijke poging tot SQL-injectie.

» Zorg ervoor dat de website gebruik maakt van een databaseaccount met beperkte rechten (richtlijn B0-12). Wanneer de website bijvoorbeeld alleen informatie mag raadplegen en niet wijzigen, kan men via SQL-injectie geen wijzigingen doorvoeren in de database. Let er op dat deze maatregel wel effect heeft op de integriteit van de database maar niet op de vertrouwelijkheid.

» Versleutel gegevens in de database (richtlijn B5-3). Het versleutelen of op een andere manier onleesbaar maken (hashen) van gegevens in de database verkleint niet de kans op aanwezigheid van een SQL-injectiekwetsbaarheid, maar verkleint wel de schade op het moment dat deze aanwezig is.

» Installeer altijd de laatste versies van de software waarvan de website gebruik maakt (richtlijn B0-7). Daarbij gaat het niet alleen om de webserver en de website maar ook om bijvoorbeeld databases, bibliotheken en plug-ins.

» Wees voorzichtig met het gebruik van allerlei plug-ins wanneer u gebruik maakt van een content management systeem (CMS). Vaak ontwikkelen derde partijen (anders dan de CMS-ontwikkelaars) allerlei plug-ins voor dit soort systemen en is er niet altijd voldoende aandacht besteed aan de beveiliging ervan. Wees u ervan bewust dat dergelijke plug-ins vaak ook directe toegang tot de database hebben en daarmee een SQL-injectiekwetsbaarheid kunnen introduceren die gevolgen kunnen hebben voor de gehele website.

» Voer regelmatig penetratietesten en beveiligingsscans uit op de website om kwetsbaarheden vast te stellen (richtlijnen B0-8 en B0-9).

» Zorg er tot slot voor dat kwetsbaarheden in uw website op verantwoorde wijze bij u gemeld kunnen worden. Door het publiceren van een zogenoemde responsible disclosure policy op uw website, maakt u duidelijk hoe personen en organisaties kwetsbaarheden, zoals SQL-injectiekwetsbaarheden, bij u kunnen melden en hoe het proces na melding verloopt. U kunt gebruik maken van de 'Leidraad voor responsible disclosure'[4] van het NCSC voor het opstellen van een dergelijke policy.

Tot slot

SQL-injectie is slechts één van de kwetsbaarheden die in een website aanwezig kunnen zijn. Een veilige website is daarom niet alleen beschermd tegen deze kwetsbaarheid, maar ook tegen allerlei andere kwetsbaarheden. Om organisaties hierin te helpen heeft het NCSC de "ICT-Beveiligingsrichtlijnen voor webapplicaties" opgesteld. Zie voor meer informatie over en de inhoud van deze richtlijnen de website van het NCSC via https://www.ncsc.nl[5].

[4] https://www.ncsc.nl/actueel/nieuwsberichten/leidraad-responsible-disclosure.html
[5] https://www.ncsc.nl/dienstverlening/expertise-advies/kennisdeling/whitepapers/ict-beveiligingsrichtlijnen-voor-webapplicaties.html

Uitgave van **Nationaal Cyber Security Centrum**
Turfmarkt 147 | 2511 DP Den Haag
Postbus 117 | 2501 CC Den Haag
www.ncsc.nl | info@ncsc.nl | T 070-751 55 55 | F 070-322 25 37
Publicatienr: FS-2014-05 | Aan deze informatie kunnen geen rechten worden ontleend.