VERTROUWELIJK • VOORBEELDRAPPORT

Rapportage penetratietest

Grey box test, webapplicatie en API

Klantportaal van Voorbeeld B.V.

pentestingservice.nl

Opdrachtgever	Voorbeeld B.V. (fictief)
Scope	portaal.voorbeeld-bv.nl en bijbehorende API
Type test	Grey box, webapplicatie en API
Testperiode	3 tot en met 7 maart 2025
Rapportdatum	12 maart 2025
Versie	1.0 (definitief)
Uitgevoerd door	{{TESTER: naam, certificering zoals OSCP}}

Documentgegevens

Dit is een voorbeeldrapport. De opdrachtgever, de bevindingen en alle gegevens zijn fictief en dienen om te laten zien hoe een rapportage van pentestingservice.nl is opgebouwd. Een echt rapport bevat de resultaten van jouw eigen omgeving.

Versiebeheer

Versie	Datum	Auteur	Wijziging
0.1	10-03-2025	{{TESTER}}	Concept ter interne review
0.9	11-03-2025	{{TESTER}}	Concept gedeeld met opdrachtgever
1.0	12-03-2025	{{TESTER}}	Definitieve versie na terugkoppeling

Distributie en classificatie

Classificatie	Vertrouwelijk. Deel dit rapport alleen met betrokkenen die de bevindingen herstellen.
Ontvangers	Contactpersoon opdrachtgever, verantwoordelijke voor de applicatie, DevOps-team
Bewaartermijn	Volgens de afspraken in de opdrachtbevestiging

Managementsamenvatting

In maart 2025 voerde pentestingservice.nl een grey box penetratietest uit op het klantportaal van Voorbeeld B.V. en de onderliggende API. Het doel: in kaart brengen welke risico's een aanvaller met een gebruikersaccount kan misbruiken, en welke gegevens daarbij bereikbaar zijn.

De test bracht zes bevindingen aan het licht. Twee daarvan vragen om snel handelen. Via een zwakke plek in de inlogfunctie kon de tester gegevens uit de database benaderen. Daarnaast kon een gewone gebruiker de bestellingen van andere klanten inzien, doordat de autorisatie per object ontbrak. Beide risico's raken persoonsgegevens en vragen om herstel op korte termijn.

> **Risico-oordeel:** het portaal biedt op onderdelen een degelijke basis. De twee zwaarste bevindingen maken op dit moment toegang tot klantgegevens mogelijk. Na herstel van de kritieke en hoge bevindingen daalt het risico sterk. Een hertest bevestigt of het herstel werkt.

Bevindingen naar risiconiveau

Risiconiveau	Aantal	Toelichting
Kritiek	1	Direct herstel gewenst
Hoog	2	Herstel binnen twee weken
Middel	2	Herstel binnen dit kwartaal

Laag	1	Meenemen in regulier onderhoud

Scope en aanpak

De test richtte zich op de webapplicatie en de API achter de login. De opdrachtgever leverde vooraf testaccounts voor twee rollen: een gewone klant en een beheerder. Daarmee kon de tester de omgeving vanuit het perspectief van een ingelogde aanvaller onderzoeken.

Binnen scope

- portaal.voorbeeld-bv.nl, inclusief inlog, dashboard en accountbeheer
- De REST-API op api.voorbeeld-bv.nl, voor zover bereikbaar vanuit het portaal
- Autorisatie tussen rollen en tussen klanten onderling

Buiten scope

- Onderliggende infrastructuur en netwerkcomponenten
- Fysieke toegang en social engineering
- Denial-of-service en belastingtesten

Werkwijze

De test volgde de OWASP Web Security Testing Guide en de fasering van PTES, in lijn met NIST SP 800-115. Het onderzoek bestond voor het grootste deel uit handmatig werk, aangevuld met tooling voor verkenning en verificatie. Elke bevinding is met de hand bevestigd, zodat het rapport geen valse meldingen bevat.

Risicoclassificatie

De risiconiveaus volgen de CVSS-methodiek. Het niveau weegt de kans op misbruik en het gevolg voor Voorbeeld B.V. samen.

Niveau	CVSS-score	Betekenis
Kritiek	9.0 tot 10.0	Direct misbruik mogelijk, groot gevolg voor data of systemen
Hoog	7.0 tot 8.9	Reëel misbruik, gevoelige gegevens in het geding
Middel	4.0 tot 6.9	Misbruik onder voorwaarden, beperkt gevolg
Laag	0.1 tot 3.9	Klein risico, vaak informatie voor een aanvaller

Bevindingen op een rij

ID	Bevinding	Niveau	Status
F-01	SQL-injectie in de inlogfunctie	Kritiek	Open

F-02	Kapotte objectautorisatie in de order-API	Hoog	Open
F-03	Verouderde component met bekende kwetsbaarheid	Hoog	Open
F-04	Zwak wachtwoordbeleid, aanmelding te vaak te proberen	Middel	Open
F-05	Ontbrekende beveiligingsheaders en verouderde TLS	Middel	Open
F-06	Informatielekkage via gedetailleerde foutmeldingen	Laag	Open

Detailbevindingen

F-01 SQL-injectie in de inlogfunctie

Kritiek **CVSS:** 9.1 **Locatie:** /login (e-mailveld) **Status:** Open

Beschrijving. Het inlogformulier verwerkt de invoer van het e-mailveld zonder deze te scheiden van de databasequery. Daardoor kon de tester de query beïnvloeden en gegevens uit onderliggende tabellen benaderen zonder geldige inloggegevens.

Gevolg. Een aanvaller kan klantgegevens uitlezen en de inlogcontrole omzeilen. Dat raakt persoonsgegevens en de vertrouwelijkheid van het portaal.

Bewijs. De tester benaderde via het inlogscherm records uit de gebruikerstabel. De schermafbeeldingen staan in bijlage B (in dit voorbeeld weggelaten).

Aanbeveling. Werk met geparametriseerde queries of prepared statements, zodat invoer nooit als code wordt uitgevoerd. Valideer invoer server-side en beperk de rechten van het databaseaccount tot het minimum.

F-02 Kapotte objectautorisatie in de order-API

Hoog **CVSS:** 8.1 **Locatie:** /api/orders/{id} **Status:** Open

Beschrijving. De order-API controleert wel of een gebruiker is ingelogd, en niet of het opgevraagde ordernummer bij die gebruiker hoort. Door het ID in de aanvraag aan te passen, kreeg de tester de bestellingen van een andere klant te zien.

Gevolg. Elke ingelogde klant kan de ordergegevens van anderen inzien, inclusief naam, adres en bestelhistorie. Dit risico vind je alleen wanneer je met meerdere accounts naast elkaar test.

Bewijs. Met het account van klant A vroeg de tester een order van klant B op. De API gaf de gegevens terug in plaats van een foutmelding.

Aanbeveling. Controleer bij elke aanvraag of het object bij de ingelogde gebruiker hoort. Voer deze autorisatie server-side uit, per object, en dek de hele API af met dezelfde check.

F-03 Verouderde component met bekende kwetsbaarheid

Hoog **CVSS:** 7.5 **Locatie:** applicatieframework (backend) **Status:** Open

Beschrijving. De applicatie draait op een frameworkversie waarvoor een bekende kwetsbaarheid bestaat. Voor deze kwetsbaarheid is publiek een herstel beschikbaar.

Gevolg. Een aanvaller kan de bekende zwakke plek gericht misbruiken. Bekende kwetsbaarheden vormen een dankbaar doelwit, omdat de aanpak vrij verkrijgbaar is.

Bewijs. De tester stelde de versie vast via de reactieheaders van de server.

Aanbeveling. Werk het framework bij naar een versie met herstel. Richt een proces in dat afhankelijkheden periodiek controleert op bekende kwetsbaarheden.

F-04 Zwak wachtwoordbeleid en herhaalde aanmeldpogingen

Middel **CVSS:** 5.9 **Locatie:** /login **Status:** Open

Beschrijving. Het portaal staat korte wachtwoorden toe en beperkt het aantal aanmeldpogingen niet. Daardoor kan een aanvaller geautomatiseerd wachtwoorden aflopen.

Gevolg. Accounts met een zwak wachtwoord lopen risico op overname. In samenhang met F-01 vergroot dit de kans op toegang.

Aanbeveling. Stel een minimumlengte in, controleer wachtwoorden tegen bekende gelekte lijsten, en begrens het aantal pogingen. Bied tweestapsverificatie aan voor klantaccounts.

F-05 Ontbrekende beveiligingsheaders en verouderde TLS

Middel **CVSS:** 5.3 **Locatie:** portaal.voorbeeld-bv.nl **Status:** Open

Beschrijving. De server stuurt een deel van de aanbevolen beveiligingsheaders niet mee, en accepteert een verouderd TLS-protocol. Daardoor is het portaal gevoeliger voor aanvallen op het transport en in de browser.

Gevolg. Een aanvaller op het netwerk kan verkeer sneller onderscheppen of manipuleren dan bij een strakke configuratie.

Aanbeveling. Zet de headers voor transportbeveiliging, inhoudstype en framing aan. Schakel verouderde TLS-versies uit en houd alleen de actuele versies aan.

F-06 Informatielekkage via gedetailleerde foutmeldingen

Laag **CVSS:** 3.7 **Locatie:** diverse endpoints **Status:** Open

Beschrijving. Bij fouten toont de applicatie technische details, zoals padnamen en versies. Die informatie helpt een aanvaller bij de verkenning.

Gevolg. Een aanvaller leert sneller hoe de omgeving is opgebouwd, wat de aanloop naar een aanval verkort.

Aanbeveling. Toon gebruikers een neutrale foutmelding en log de technische details alleen aan de serverkant.

Herstel en hertest

De volgorde hieronder helpt je het risico snel te verlagen. Pak eerst de kritieke en hoge bevindingen op, daarna de rest.

Prioriteit	Bevinding	Advies voor doorlooptijd
1	F-01 SQL-injectie	Direct
2	F-02 Objectautorisatie	Binnen twee weken
3	F-03 Verouderde component	Binnen twee weken
4	F-04 en F-05	Binnen dit kwartaal
5	F-06	Regulier onderhoud

> **Hertest.** Na herstel toetst pentestingservice.nl of de zwakke plekken daadwerkelijk zijn verholpen. Het hertestrapport dient als bewijs richting je bestuur, auditors en toezichthouders.

Bijlage A Methodologie en tooling

De test volgde een vaste fasering: verkenning, in kaart brengen, actief testen, verificatie en rapportage. De onderstaande kaders vormden de basis.

- OWASP Web Security Testing Guide, voor de dekking van webapplicaties
- PTES, voor de fasering van de test
- NIST SP 800-115, als technisch kader

Voor verkenning en verificatie zetten de testers een vaste set gereedschappen in, waaronder een onderscheppende proxy, een poortscanner en scripts voor gerichte controles. Elke geautomatiseerde melding is met de hand nagelopen.

Verantwoording

Dit voorbeeldrapport dient om de opbouw van een rapportage te tonen. De opdrachtgever, de gegevens en de bevindingen zijn fictief. Een echte pentest wordt uitgevoerd binnen een vooraf afgesproken scope, met toestemming van de eigenaar van de systemen. Een rapport geeft de situatie weer op het moment van testen. Wijzigingen na de testperiode vallen buiten de rapportage. pentestingservice.nl voert testen uit volgens vastgelegde werkwijzen en met gecertificeerde testers.