Patchen zonder verrassingen

Oracle Critical Patch Updates in de praktijk

Impact, testen, koppelingen en een terugvalplan dat verder gaat dan een back-up

---

Paul de Boer

Training & Consultancy

pauldeboeronline.nl

Patchen zonder verrassingen: Oracle Critical Patch Updates in de praktijk

Paul de Boer

Oracle brengt vier keer per jaar een Critical Patch Update (CPU) uit, in januari, april, juli en oktober, en daarnaast zo nodig extra beveiligingsupdates. Voor een organisatie die met Oracle EBS werkt, is dat telkens een afweging. Wachten met installeren vergroot het risico dat bekende kwetsbaarheden worden misbruikt. Installeren zonder goede test kan de financiële administratie verstoren. In de praktijk zit het verschil vooral in de voorbereiding.

Eerst de impact bepalen

Elke CPU begint met de vraag voor welke modules en systemen kwetsbaarheden worden opgelost, en hoe ernstig die zijn. Stel op basis daarvan een korte rapportage op voor het management: waarom deze patch van belang is, op welke testomgeving wordt getest, hoe lang het testen duurt en wat de totale doorlooptijd is, van het klonen van de omgeving en het installeren van de patch tot het testen en de uiteindelijke installatie op productie. Zo kan het management een onderbouwde beslissing nemen, en weet iedereen vooraf wat er van hem of haar wordt verwacht.

Testen met het eigen maatwerk en de koppelingen

Oracle test een patch op de standaardsoftware, niet op het maatwerk van een organisatie. Stel daarom voor elke patch een specifiek testplan op, waarin naast de standaardprocessen ook het maatwerk en de koppelingen met andere systemen zijn opgenomen.

Die koppelingen zijn vaak het lastigste deel. Vrijwel elke organisatie heeft koppelingen voor inkomende en uitgaande facturen en voor betalingen, bijvoorbeeld via Apro. Daarnaast zijn er koppelingen met branchespecifieke software, met een servicebus of datawarehouse, en soms met portalen waarin klanten of leveranciers zelf gegevens inzien of aanleveren. Niet elke testomgeving beschikt over al deze koppelingen. Een testplan moet daarom vermelden wat op welke omgeving wel en niet kan worden getest, en hoe het resterende risico wordt afgedekt.

Als het toch misgaat

Een terugvalplan blijft vaak beperkt tot het terugzetten van een back-up van de database. Dat is zelden voldoende. Zodra de database teruggaat naar een eerder moment, lopen de aanpalende systemen daar niet meer mee gelijk. Of de oorzaak nu een mislukte patch, een storing of een beveiligingsincident is: de vraag is niet alleen hoe snel de back-up kan worden teruggezet, maar ook of alle aanpalende applicaties daarna weer met elkaar in de pas lopen en met elkaar kunnen communiceren.

Een goed terugvalplan beschrijft daarom vooraf onder meer:

- Naar welk moment wordt teruggezet, en hoe per aanpalend systeem wordt bepaald of en hoe het op dat moment kan aansluiten.
- Welke facturen in werkelijkheid al zijn betaald. Dat volgt uit de bankafschriften die vanaf het herstelmoment opnieuw moeten worden ingelezen. Een betaling die al is uitgevoerd, mag niet nog een keer worden verstuurd.
- Hoe wordt omgegaan met de goedkeuringsworkflow: moeten gebruikers hun acties opnieuw uitvoeren, of is er een andere oplossing?
- Welke gegevens in welke volgorde opnieuw moeten worden verwerkt, vast te stellen met queries op bijvoorbeeld transactie-ID en tijdstip.
- Wie de beslissingen neemt.

Dat overzicht maak je niet tijdens een incident, maar daarvoor.

Waar AI helpt

Ook hier vervangt AI de ervaring niet, maar het kan het voorbereidende werk wel versnellen:

De beschrijving van een CPU naast het overzicht van maatwerk en koppelingen leggen, en aangeven welke onderdelen mogelijk worden geraakt.

Op basis van eerdere testplannen een eerste opzet maken voor het testplan van een nieuwe patch.

Ondersteunen bij het opstellen en controleren van herstelqueries, waarbij een beheerder het resultaat beoordeelt.

Het overzicht van koppelingen en gegevensstromen beschrijven en actueel houden.

Wat ik organisaties aanraad

Houd een actueel overzicht bij van alle koppelingen met de ERP-omgeving, en weet welke testomgeving over welke koppelingen beschikt.

Leg voor elke CPU vast wat de impact is, wie wat test en wat de doorlooptijd is.

Toets of de periodieke back-up niet alleen snel kan worden teruggezet, maar ook zo is ingericht dat de aanpalende applicaties daarna weer met elkaar in de pas lopen.

Stel het terugvalplan op voordat het nodig is, inclusief de aanpalende systemen.

Elke organisatie en elke ERP-omgeving is anders. Wilt u weten wat dit voor uw situatie betekent? Neem dan gerust contact op via pauldeboeronline.nl.