



CUSTOMER ADVISORY

Microsoft Dynamics 365 deployments

PHP2Shell

and the portal that calls Dynamics 365.

The vulnerable component is PHP on a server you or your partner operate. Finance and Operations, Dataverse and Business Central online are Microsoft-hosted services. A portal that calls them often holds a long-lived credential.

Patch now. Remove the stored secret. Move the portal onto a platform Microsoft maintains.

26 OCTOBER 2026

Read this against every PHP host that calls Dynamics 365.

01 Patched

21 October 2026. PHP 8.5.12, 8.4.27, 8.3.36 and 8.2.35. PHP 5.4 to 8.1 have no upstream fix.

02 Severity

Two flaws, each CVSS v4.0 9.2. Unauthenticated remote code execution in a default configuration.

03 Scope

The portal, the web shop, or the script that calls Dynamics 365. The product Microsoft hosts is a separate system.

01 Patch

The PHP host, to a fixed release. Confirm the package carries the fix, not only the version number.

02 Rotate

The Entra credential on that host, then the database keys and the gateway keys.

03 Move

The portal onto a platform Microsoft maintains. PHP that remains is a bridge, with a written patch commitment.



— 21 OCTOBER 2026

Two flaws in PHP.

Both were patched after the details leaked, and after exploitation had already begun. The count that follows is of PHP on the internet. It is not a count of Dynamics 365 customers.

DATE	EVENT
17 Oct	Amazon honeypots log an exploit attempt, four days before disclosure.
20 Oct	Details leak. PHP announces an emergency release.
21 Oct	PHP 8.5.12, 8.4.27, 8.3.36 and 8.2.35 ship. Affected: PHP 5.4 through 8.5.11.
22 Oct	Cloudflare reports attempts rising from tens of thousands to millions.
23 Oct	PHP flags a third issue, CVE-2026-115560. It needs an unusual configuration.
25 Oct	Palo Alto Unit 42 counts over 22.6 million potentially exposed PHP instances.

CVE-2026-109618

Memory corruption, triggered by out-of-memory handling.
Code execution in a default configuration.

CVE-2026-109619

A PHP worker can reuse stale state from a freed request.
Code execution in a default configuration.

Described by Datadog, Unit 42 and Google Cloud. Both are CVSS v4.0 9.2.

Is Dynamics 365 itself exposed?

Finance and Operations and Dataverse are Microsoft-hosted services. Business Central online is the same class of service. The flaw sits in PHP on a server you or your partner operate. That server is the customer, vendor or partner portal, the web shop or booking site, or the middleware script that calls Dynamics 365 through OData, a custom service, or the Dataverse Web API. If that host runs a PHP version below the fixed releases, anyone on the internet can attempt to run code on it.

Sources for the timeline and the instance count: the PHP project, Amazon, Cloudflare, Datadog, Google Cloud and Palo Alto Unit 42. Software Dynamics has not independently counted those hosts.



WHAT THE PORTAL HOLDS

The damage equals the identity.

Code execution on the portal lets an attacker act as the portal. This table is analysis of typical integration designs. It is not a record of a published incident. Adjust it to the architecture you actually run.

WHAT SITS ON THE PHP HOST	WHAT AN ATTACKER GAINS
Entra ID app registration client secret or certificate	Token requests as the service principal, from anywhere, until the secret is rotated.
Cached OAuth access tokens	Short-lived Dynamics 365 access, typically about an hour, without the secret.
.env, config.php or web.config	Database credentials, SMTP credentials, and payment gateway keys.
The portal database	Customer and vendor records, order history, and password hashes.
A network position	A trusted route to Dynamics 365, to integration databases, or to an on-premises gateway.

The worst case is a wide role

An integration identity with System Administrator in Dataverse, or a wide security role in Finance and Operations, turns one request to the portal into read and write access to customers, vendors, pricing and open orders. The same host can also use the portal session to impersonate a person who is signed in.

Read the table against three facts.

01 The identity

The Entra application, and the user it maps to in Dynamics 365.

02 The role

The tables, entities and actions that identity is allowed to touch.

03 The secret

The file, vault or certificate that holds the credential tonight.



— CONTAIN

Patch, hunt, rotate.

Treat a portal that was on the internet and unpatched on 17 October as exposed until the logs say otherwise.

01 Patch the runtime

Move to PHP 8.5.12, 8.4.27, 8.3.36 or 8.2.35. On Linux, confirm the distribution package carries the fix. A backport can keep the old version number. PHP 5.4 to 8.1 have no upstream fix. Upgrade those hosts, or take them off the internet.

02 Read the logs

Check web and process logs for PHP workers spawning shells or child processes, and for unusually long filenames in POST bodies.

03 Rotate the credential

Rotate the Entra ID client secret or certificate the portal uses, then database and gateway keys. Revoke sessions where you can.

04 Hunt in the tenant

Review service principal sign-ins for new IP addresses or odd hours. Review Dynamics 365 audit logs for bulk reads, exports or writes by the integration user.

05 Shrink the identity

Give the integration its own service principal and a minimal security role. Restrict it by Conditional Access or IP where your licensing allows it. Prefer a certificate in Key Vault over a secret in a config file.

06 Narrow the path

Add web-application-firewall rules for the published signatures. Limit outbound traffic from the PHP host. Disable session upload progress. That setting mitigates CVE-2026-115560.

Who owns which action

The infrastructure owner patches the host and confirms the package. The identity owner rotates the Entra credential. The Dynamics 365 administrator reads the audit log and cuts the security role. One person should be named for each, today.

FIXED RELEASES, 21 OCTOBER 2026

PHP 8.5 8.5.12	PHP 8.4 8.4.27	PHP 8.3 8.3.36	PHP 8.2 8.2.35
--------------------------	--------------------------	--------------------------	--------------------------

PHP 5.4 to 8.1 have no upstream fix. On Linux, match the package to the fix. A backport can keep the old version number.



— KEY VAULT AND MANAGED IDENTITY

A stolen file should be useless.

Key Vault alone does not stop code running inside the portal from using what the portal can use. Pair the vault with a small role, and with a front door that exposes only the operations the portal needs.

PATTERN	WHAT IT STOPS	WHAT REMAINS
Secret in a config file. The client secret sits in .env or config.php. Common today.	Nothing.	Full theft, and reuse from anywhere.
Secret in Azure Key Vault. The portal reads it at runtime with a managed identity. Access is Azure RBAC. Rotation is on a schedule.	Secrets leaking through file reads and backups. Rotation becomes routine.	A compromised portal can still read the secret while it runs.
Certificate, private key held in Key Vault. The portal asks Key Vault to sign the client assertion. The key is not exported.	Exfiltrating a credential that can be reused later.	The attacker can sign requests only while the portal is compromised.
Managed identity with a federated credential. The app registration trusts the host. No secret exists.	Credential theft, on an Azure-hosted portal.	The attacker can still request tokens from the compromised host.

At the Dynamics 365 end, on every pattern.

- **Dataverse**
An application user with a custom security role, limited to the tables the portal needs. Never System Administrator.
- **Finance and Operations**
Register the Entra application and map it to a service user. The role covers only the required data entities and actions.
- **Entra ID**
Conditional Access for the workload identity, where licensing allows it. Alert on sign-ins from a new location.
- **A front door**
Azure API Management, or an Azure Functions facade, between the portal and Dynamics 365. Only the operations the portal needs, with rate limits and logging. The portal then holds no token that can query an arbitrary entity.

Do these in this order

Rotate the secret that sits in a file. Then cut the role. Then put a front door in front of Dynamics 365. A vault, without the smaller role and the front door, still leaves a portal that can ask for a token.



ASSESS, THEN TRIAGE

Every host that touches Dynamics 365.

Capture eight facts for each portal and each script. Then put the host in one tier.

• PHP version and framework	• Hosting location
• Who owns it, and who patches it	• Which Dynamics 365 identity it uses
• The roles on that identity	• Where the secret lives
• Which data it reads or writes	• When it was last patched or tested

TIER	PROFILE	ACTION WINDOW
1	On the internet. Writes to Dynamics 365, or uses a broad role. Secret in a config file.	Patch and rotate now. A front door and a smaller role within 30 days.
2	On the internet. Read-only, or a narrow role.	Patch now. Move the secret to Key Vault within 60 days.
3	Internal only, or it touches little data.	Patch in the normal cycle. Fold it into the migration backlog.
4	Nobody owns it, it is unpatched, or nobody can say what it is for.	Isolate or retire. Do not carry it forward.

How to weight a host

Give extra weight to a PHP version with no upstream fix, and to a portal that handles payment, pricing or customer master data. A public vendor portal on PHP 8.1, with a client secret in config.php and a System Administrator application user, is Tier 1. An internal read-only price list on a fixed PHP 8.3 build is Tier 3. Retiring a portal that nobody can justify is the cheapest reduction available.

One row per host is enough to start: the name, the PHP version, the identity, the role, where the secret sits, and the tier you assign.



A PLATFORM MICROSOFT MAINTAINS

Most PHP portals repeat Power Platform.

Every line of PHP is code you must patch yourself. Match each portal to the lowest-maintenance option that meets the need.

OPTION	BEST FOR	TRADE-OFF
Power Pages	External customers, partners and vendors. Forms, lists, order or case status, document upload.	No server-side PHP. Custom logic moves to Dataverse plug-ins, Power Automate or Azure Functions. Data must be in Dataverse, so Finance and Operations data needs virtual tables, dual-write or an integration. Access is web roles and table permissions. Licensing is capacity-based. Check it before you commit.
Model-driven apps	Internal staff, and licensed partner users, working with Dynamics 365 records.	The full Dynamics 365 security model, and the audit log. Not for anonymous users, and not for high-volume external users.
Azure Functions or Logic Apps, behind API Management	Headless business-to-business integrations, and bespoke logic the portal must keep.	You own the code, on a Microsoft-managed runtime, with a managed identity, patched by the platform. It needs an API design and monitoring.
Harden, and keep PHP	A complex application that cannot move yet.	You keep the patching burden. Use it only with a front door, Key Vault, a firewall and a written patch commitment. Treat it as a bridge, not a destination.

A practical rule

If a portal mostly shows and edits records, move it to Power Pages or a model-driven app. If it contains real business logic or integrations, extract that logic into a governed API layer first, then retire the PHP front end.

Records

It mostly shows and edits records. Move it to Power Pages, or to a model-driven app for licensed users.

Logic

It holds pricing, payment or an integration. Extract that into the API layer before you retire the PHP.

A bridge

It cannot move this quarter. Harden it, name the owner, and write the date it will be retired.



— ONE HUNDRED AND EIGHTY DAYS

Shrink what holds a trusted identity.

PHP2Shell will not be the last critical flaw in a runtime you operate. The durable fix is less custom, internet-facing code holding a Dynamics 365 identity, and an identity that is cheap to rotate and hard to misuse.

DAYS	0 to 3	Contain Patch, hunt, rotate, and disable session upload progress.
DAYS	3 to 30	Stabilise Finish the inventory. Move secrets into Key Vault. Cut integration roles to the minimum. Turn on Entra sign-in alerts and Dynamics 365 audit alerts.
DAYS	30 to 90	Redesign Put API Management or Azure Functions in front of Dynamics 365. Retire every Tier 4 portal. Pilot the highest-risk external portal on Power Pages.
DAYS	90 to 180	Migrate Move the remaining portals, by tier, to Power Pages, a model-driven app, or the API layer. Any PHP that remains sits on a written patch commitment.

Send the inventory, if you want it read against these tiers, to the office that holds your agreement. Name the owner of each host. We will mark the tier. We will not invent a portal you do not run.

THREE REGISTERED COMPANIES

Johannesburg

South Africa

Software Dynamics SA (Pty) Ltd

Reg. No. 2026/211288/07

37 Willow Avenue, Craigavon, Sandton 2191

+27 72 972 3135

sales@softwaredynamics.co.za

Nairobi

Kenya

Software Dynamics Limited

CPR/2011/62397

Mezzanine Floor, Titan Complex, Chaka Road,
Nairobi

+254 768 341 410

sales@softwaredynamics.co.ke

Melbourne

Australia

Software Dynamics Group Pty Ltd

ACN 696 184 666

5A Springs Road, Donnybrook, VIC 3064

+61 423 605 291

sales@softwaredynamics.com.au

Group enquiries sales@softwaredynamicsgroup.com