YellowKey BitLocker Bypass via Windows Recovery Environment

Technical advisory and mitigation playbook for an unpatched zero-day affecting Windows 11, Windows Server 2022 and Windows Server 2025 in default TPM-only configuration.

Field	Value
Advisory ID	SA-2026-05-17
Revision	v1.1 — May 22, 2026 (corrections applied)
Prepared by	nLogic AS Kim Hansen, Security Advisor
Original publication	May 17, 2026
Classification	Internal reference / customer deliverable
CVE	Not assigned (as of publication)
Patch status	No patch available (as of publication)
Affected systems	Windows 11 (all versions), Windows Server 2022, Windows Server 2025
Not affected	Windows 10
Effective mitigation	TPM+PIN pre-boot authentication; UEFI hardening; reagentc /disable where appropriate

Revision History

▲ About this revision

Version 1.1 applies three technical corrections identified during a post-publication verification pass against Microsoft Learn (May 22, 2026). All revised sections are clearly marked with an amber ”UPDATED v1.1” badge throughout the document. The strategic guidance, threat model, CIS/NIST/NSM mapping and decision matrices from v1.0 remain unchanged.

Changes in v1.1:

Section	Change	Reason
2.2 — Hunt 4	KQL query rewritten	Original used ActionType has "Usb" and has_any ("Boot", "Startup", "Reboot") — these values do not exist in DeviceEvents. Replaced with documented PnP ActionType values and DeviceInfo-based boot proxy.
2.2 — Hunt 5	Best-effort note added	DeviceEvents rarely contains explicit BitLocker ActionType values. Added pointer to Windows Event Log channel Microsoft-Windows-BitLocker/BitLocker Management for full coverage.
3.3 — DMA check	PowerShell corrected	Comment "5 = DMA protection" was wrong. Value 5 in SecurityServicesRunning is Kernel-mode Hardware-enforced Stack Protection. Replaced with AvailableSecurityProperties -contains 3 and Get-ComputerInfo approach.
3.2 — Settings Catalog	TPM Startup row clarified	Dropdown values in Intune Settings Catalog are Allow TPM / Do not allow TPM / Require TPM, not "Required". Row updated to reflect actual UI options.
3.2 — TPM+PIN protector	PowerShell alternative added	Added Add-BitLockerKeyProtector with -TpmAndPinProtector for unattended rollout, alongside the existing interactive manage-bde command.
3.4 — reagentc note	Caveat added	Note added that reagentc /disable may block new BitLocker enablement; devices with BitLocker already active are unaffected.

1. Executive Summary

YellowKey is a publicly disclosed zero-day weakness that bypasses BitLocker protection on Windows 11, Windows Server 2022 and Windows Server 2025 when configured in the default TPM-only mode. The attack requires physical access to the device and a prepared USB stick. Microsoft has not issued a CVE or a patch as of the publication date.

The vulnerability does not break BitLocker's cryptography. It exploits a cross-volume bug in the Transactional NTFS log replay performed by the Windows Recovery Environment (WinRE). A specially crafted FsTx structure on an attacker-controlled volume causes WinRE to delete winpeshl.ini on the BitLocker volume, removing the recovery shell restriction and dropping the attacker to cmd.exe with full read access to the already-mounted, TPM-unlocked disk.

Preconditions for attack

- Physical access to the device
- USB stick with a prepared \System Volume Information\FsTx structure (alternatively, write access to the EFI partition)
- Ability to boot the device into WinRE
- BitLocker configured in TPM-only mode (default on Windows 11)

Effective mitigations available today

- TPM+PIN (pre-boot authentication) — reliably blocks the published PoC
- Disable USB boot in UEFI and set a UEFI administrator password
- reagentc /disable on endpoints with centralized recovery (Autopilot, MECM, SCCM)

2. Detection — Microsoft Defender XDR Advanced Hunting

2.1 Detection limitations

YellowKey executes before the operating system boots. Standard EDR/AV agents (Defender, CrowdStrike, SentinelOne and similar) are not active in WinRE and cannot detect the exploit itself. Detection must therefore focus on:

- Preparatory activity (FsTx structures placed on internal or removable volumes)
- Indirect indicators (unexpected WinRE boots, USB activity around reboot events)
- Forensic indicators on devices suspected of compromise

2.2 KQL queries

Hunt 1 — FsTx structures across volumes

```
// Suspicious FsTx folders in System Volume Information
DeviceFileEvents
| where Timestamp > ago(30d)
| where FolderPath has @"System Volume Information\FsTx"
    or FolderPath has @"\FsTx\"
| where ActionType in ("FileCreated", "FileModified", "FileRenamed")
| project Timestamp, DeviceName, ActionType, FileName, FolderPath,
    InitiatingProcessFileName, InitiatingProcessCommandLine,
    InitiatingProcessAccountName
| order by Timestamp desc
```

Hunt 2 — winpeshl.ini manipulation

winpeshl.ini is the file WinRE uses to constrain the recovery shell. Deletion or modification outside servicing operations is a strong indicator.

```
DeviceFileEvents
| where Timestamp > ago(30d)
| where FileName =~ "winpeshl.ini"
| where ActionType in ("FileDeleted", "FileModified", "FileRenamed", "FileCreated")
| project Timestamp, DeviceName, ActionType, FolderPath,
    InitiatingProcessFileName, InitiatingProcessCommandLine,
    InitiatingProcessAccountName
| order by Timestamp desc
```

Hunt 3 — reagentc.exe execution

reagentc.exe is Microsoft's WinRE administration utility. Legitimate executions occur during patching and servicing — baseline tracking helps detect anomalies.

```
DeviceProcessEvents
| where Timestamp > ago(30d)
| where FileName =~ "reagentc.exe"
| project Timestamp, DeviceName, AccountName, AccountDomain,
    ProcessCommandLine, InitiatingProcessFileName, InitiatingProcessCommandLine,
    InitiatingProcessParentFileName
| order by Timestamp desc
```

Hunt 4 — USB activity around boot events

▲ UPDATED v1.1 • Hunt 4 KQL replaced. The previous query used non-existent ActionType values ("Usb", "Boot", "Startup", "Reboot"). The revised query uses documented PnP ActionType values and DeviceInfo first-seen heartbeats as a boot proxy.

Why the change matters: KQL has is case-insensitive but values like "Usb" and "Boot" are not substrings of any documented DeviceEvents ActionType. The original query would have returned zero rows in production Defender XDR.

Correlates USB connections with subsequent reboots. Requires manual interpretation, does not catch the attack alone but supports forensic investigation.

```
// Hunt 4 (v1.1) — PnP activity correlated with device boot proxy
let PnpEvents = DeviceEvents
    | where Timestamp > ago(7d)
    | where ActionType in ("PnPDeviceConnected", "PnPDeviceAllowed",
                           "PnPDeviceBlocked", "RemovableStoragePolicyTriggered")
    | project PnpTime = Timestamp, DeviceName, PnpAction = ActionType, AdditionalFields;
let BootProxy = DeviceInfo
    | where Timestamp > ago(7d)
    | summarize FirstSeen = min(Timestamp) by DeviceName, bin(Timestamp, 1h)
| project BootTime = FirstSeen, DeviceName;
PnpEvents
| join kind=inner BootProxy on DeviceName
| where datetime_diff('minute', BootTime, PnpTime) between (-30 .. 30)
| project DeviceName, PnpTime, BootTime, PnpAction, AdditionalFields
| order by BootTime desc
```

For higher-fidelity boot detection, ingest Windows Event Log IDs 6005 / 6006 / 6008 (System log) into Sentinel and join against this query — DeviceEvents does not surface explicit boot events.

Hunt 5 — BitLocker-related events

▲ UPDATED v1.1 • Best-effort note added. DeviceEvents rarely surfaces explicit BitLocker ActionType strings; high-fidelity BitLocker telemetry lives in the Microsoft-Windows-BitLocker/BitLocker Management Windows Event Log channel.

```
DeviceEvents
| where Timestamp > ago(30d)
| where ActionType has "BitLocker"
    or AdditionalFields has "BitLocker"
    or AdditionalFields has "VolumeMaster"
| project Timestamp, DeviceName, ActionType, AdditionalFields,
    InitiatingProcessFileName
| order by Timestamp desc
```

Supplement with the Microsoft-Windows-BitLocker/BitLocker Management event channel (e.g., Event IDs 24580, 24586). Ingest via Azure Monitor Agent or Sentinel data connector for full coverage.

Hunt 6 — Suspicious cmd.exe executions from recovery context

After a successful bypass, subsequent OS boots may show traces of modified files or installed payloads. Look for processes created by unexpected parents.

```
DeviceProcessEvents
| where Timestamp > ago(30d)
| where FileName in ("cmd.exe", "powershell.exe", "regedit.exe")
| where InitiatingProcessParentFileName !in
    ("explorer.exe", "svchost.exe", "services.exe", "userinit.exe",
     "winlogon.exe", "RuntimeBroker.exe", "mmc.exe", "Code.exe",
     "WindowsTerminal.exe", "wt.exe")
| where AccountName != "system"
| project Timestamp, DeviceName, FileName, ProcessCommandLine,
    InitiatingProcessFileName, InitiatingProcessParentFileName, AccountName
| order by Timestamp desc
```

2.3 PowerShell — local scanning for FsTx structures

Run on an endpoint to identify suspicious structures on local or mounted volumes. Suitable for Intune Proactive Remediations or scripted rollout. Script must run as SYSTEM to read System Volume Information (Proactive Remediations does this by default).

```
# Scan every accessible filesystem volume for FsTx structures
$findings = @()
Get-PSDrive -PSProvider FileSystem | ForEach-Object {
    $path = Join-Path $_.Root 'System Volume Information\FsTx'
    if (Test-Path -Path $path -ErrorAction SilentlyContinue) {
        $findings += [PSCustomObject]@{
            Drive     = $_.Root
            Path      = $path
            FileCount = (Get-ChildItem -Force $path -ErrorAction SilentlyContinue).Count
            ScanTime  = Get-Date
        }
    }
}

if ($findings.Count -gt 0) {
    $findings | Format-Table -AutoSize
    Write-Warning "Suspicious FsTx structures found. Verify legitimacy manually."
    exit 1
} else {
    Write-Output "No FsTx structures found."
    exit 0
}
```

Note: the path System Volume Information\FsTx may also exist legitimately on system volumes as part of Transactional NTFS functionality. Findings must be verified against a baseline. Particular attention should be paid to occurrences on removable media.

3. Mitigation Playbook

3.1 Phase 1 — Inventory (Day 0—2) |

Goal: Identify which endpoints run TPM-only BitLocker.

Intune reporting:

- Endpoint Security → Disk encryption → Reports
- Filter on devices without a ”Startup PIN” protector

PowerShell on individual devices:

```
Get-BitLockerVolume | Select-Object MountPoint, ProtectionStatus, EncryptionMethod,
KeyProtector
manage-bde -protectors -get C:
reagentc /info
```

Rollout priority:

1. Travelling employees (laptops that leave the office)
2. Privileged users (IT admins, leadership, finance)
3. Devices with access to sensitive data (GDPR, health records, client data)
4. General fleet

3.2 Phase 2 — TPM+PIN via Intune (Day 2—14)

Settings catalog configuration (recommended approach):

Devices → Configuration → Create → New Policy → Windows 10 and later → Settings catalog

▲ UPDATED v1.1 • Intune Settings Catalog table — TPM Startup row clarified. The Settings Catalog drop-down values are Allow TPM / Do not allow TPM / Require TPM, not ”Required”.

Relevant settings under the BitLocker category:

Setting (Settings Catalog UI name)	Value
Require Device Encryption	Enabled
Require additional authentication at startup	Enabled
Configure TPM startup:	Do not allow TPM (forbids TPM-only — explicit)
Configure TPM startup PIN:	Require startup PIN with TPM
Configure minimum PIN length for startup	8 (minimum; consider 10+)
Allow enhanced PINs for startup	Enabled (alphanumeric)
Configure use of passwords for operating system drives	Disabled

Equivalent outcome with Configure TPM startup: "Allow TPM" + Configure TPM startup PIN: "Require startup PIN with TPM" — the PIN requirement alone is sufficient to enforce TPM+PIN. The explicit "Do not allow TPM" makes the intent unambiguous and prevents future drift if the PIN setting is relaxed.

Equivalent CSP paths (for reference):

```
./Device/Vendor/MSFT/BitLocker/SystemDrivesRequireStartupAuthentication
./Device/Vendor/MSFT/BitLocker/SystemDrivesMinimumPINLength
```

Equivalent Group Policy (locally or via GPMC):

```
Computer Configuration
  → Administrative Templates
    → Windows Components
      → BitLocker Drive Encryption
        → Operating System Drive
          → Require additional authentication at startup
```

Modifying existing protectors (per device):

```
# Interactive — prompts the user for the PIN
manage-bde -protectors -add C: -TPMAndPIN
# Verify
manage-bde -protectors -get C:
# Remove the pure TPM protector once TPMAndPIN is confirmed active
manage-bde -protectors -delete C: -type TPM
```

▲ UPDATED v1.1 • PowerShell alternative added for unattended rollout (replaces interactive manage-bde prompt).

```
# Unattended alternative using PowerShell BitLocker module
$pin = ConvertTo-SecureString -String "<choose-strong-pin>" -AsPlainText -Force
Add-BitLockerKeyProtector -MountPoint "C:" -TpmAndPinProtector -Pin $pin

# Verify
Get-BitLockerVolume -MountPoint "C:" | Select-Object -ExpandProperty KeyProtector

# Remove pure TPM protector (after TPM+PIN is confirmed active)
$tpmId = (Get-BitLockerVolume -MountPoint "C:").KeyProtector |
    Where-Object { $_.KeyProtectorType -eq 'Tpm' } |
    Select-Object -ExpandProperty KeyProtectorId
Remove-BitLockerKeyProtector -MountPoint "C:" -KeyProtectorId $tpmId
For production rollout, source the PIN from a secure pipeline (Key Vault / SCEPman / per-device secret) rather than hard-coding.
Avoid logging the secure string in transcripts.
```

User communication — important:

- Explain that a PIN must be entered at every boot
- Establish a process for forgotten PINs (Self-Service Password Reset or helpdesk flow)
- Consider Intune Proactive Remediation to monitor PIN status

3.3 Phase 3 — UEFI/BIOS hardening

These measures block the delivery mechanism for the published PoC:

8. Set a UEFI administrator password — prevents the attacker from changing boot order
9. Disable USB boot — removes the simplest attack vector
10. Confirm Secure Boot is enabled and functioning — verify with Confirm-SecureBootUEFI
11. Enable Kernel DMA Protection (requires VT-d / AMD-Vi capable hardware) — blocks DMA-based side channels

Verification script:

▲ UPDATED v1.1 • DMA Protection check corrected. In Win32_DeviceGuard, value 5 in SecurityServicesRunning is Kernel-mode Hardware-enforced Stack Protection — not DMA Protection.

```
# Secure Boot status (unchanged)
Confirm-SecureBootUEFI

# v1.1 — Correct Kernel DMA Protection check
$dg = Get-CimInstance -Namespace root\Microsoft\Windows\DeviceGuard `
                      -ClassName Win32_DeviceGuard

# (a) Is DMA Protection AVAILABLE on this hardware?
$dmaAvailable = $dg.AvailableSecurityProperties -contains 3
"DMA Protection available: $dmaAvailable"

# (b) Boot DMA Protection actual status — most reliable via Get-ComputerInfo
$info = Get-ComputerInfo |
    Select-Object DeviceGuardSecurityServicesConfigured,
                  DeviceGuardSecurityServicesRunning,
                  DeviceGuardAvailableSecurityProperties
$info | Format-List

# (c) Boot DMA Protection direct via the MDM CSP node
#     ./Vendor/MSFT/DeviceStatus/DMA/BootDMAProtectionStatus  (1 = Enabled, 2 = Disabled)
```

Correct value map for SecurityServicesRunning (Win32_DeviceGuard): 0 = none, 1 = Credential Guard, 2 = Memory Integrity (HVCI), 3 = System Guard Secure Launch, 4 = SMM Firmware Measurement, 5 = Kernel-mode Hardware-enforced Stack Protection, 6 = Stack Protection in Audit mode, 7 = Hypervisor-Enforced Paging Translation. In AvailableSecurityProperties (a different array on the same object), value 3 = DMA Protection.

3.4 Phase 4 — WinRE decision

reagentc /disable removes the attack vector entirely but has operational consequences.

▲ UPDATED v1.1 • Caveat added regarding the interaction between reagentc /disable and BitLocker enablement.

Microsoft documents that BitLocker cannot be newly enabled while WinRE is disabled. Devices that already have BitLocker active are unaffected — BitLocker continues to operate normally. Re-enable WinRE (reagentc /enable) before provisioning new devices.

Decision matrix:

Criterion	reagentc /disable?
Central reset/reimaging via Autopilot / MECM / SCCM	Yes, strongly consider
Kiosk or locked-down device (display, prod-PC)	Yes
Field and travel equipment with high sensitivity	Yes
General office fleet with local recovery	No — emphasize TPM+PIN instead
Dev/test machines that frequently need Boot Options	No
Servers (Windows Server 2022/2025)	Evaluate case by case

Implementation (per device):

```
# Check status first
reagentc /info

# Disable WinRE
reagentc /disable

# Re-enable if needed
reagentc /enable
```

Intune rollout: package as a Win32 app with a detection rule on the reagentc /info output. Alternatively, deploy as a PowerShell script in a configuration profile.

3.5 Phase 5 — Compensating controls

For devices where pre-boot PIN is not feasible (kiosk mode, headless servers, exception-approved users):

- Physical security: lockable laptop cases, cable locks at fixed workstations, asset tracking
- Travel policy: specific guidance for travel outside the EEA — recommend ”burner laptops” for high-risk countries
- Tampering detection: enable Secured-Core PC capabilities where hardware supports it
- Credential isolation: minimize local AD/Entra credential cache (Credential Guard enabled)
- Data minimization: reduce what is actually stored locally — push to OneDrive/SharePoint with Known Folder Move

4. CIS Controls v8 — Risk Assessment

4.1 Directly affected controls

Ctrl	IG	Status	Consequence
3.6	IG1	Assumption broken	"BitLocker enabled" is no longer sufficient as a stand-alone control.
4.1	IG1	Must be updated	Endpoint baseline configuration must explicitly require pre-boot authentication.
4.5	IG1	Not directly	—
4.7	IG2	Indirectly	—
8.2	IG1	Affected	Standard EDR logging does not capture pre-boot activity — supplement with UEFI logs.
10.1	IG1	Limited effect	AV/EDR does not run in WinRE.
13.7	IG2	Limited effect	Same reason as 10.1.

4.2 Recommended baseline updates

For IG1 baseline (minimum level):

- Retain the BitLocker requirement (3.6)
- New requirement: UEFI administrator password enabled
- New requirement: USB boot disabled on laptops
- Extend the existing central logging requirement to include USB events

For IG2 baseline (recommended for most organizations):

- TPM+PIN mandatory on all laptops with access to classified data
- Minimum PIN length: 8 alphanumeric characters
- Kernel DMA Protection enabled where hardware supports it
- Defender XDR Advanced Hunting queries from section 2 implemented as custom detections

For IG3 baseline (high security):

- TPM+PIN on all endpoints, including desktops
- reagentc /disable on all non-developer devices
- Secured-Core PC required for new hardware procurement
- Physical security and chain-of-custody procedures for travelling devices

4.3 Relation to other frameworks

NSM Basic Principles for ICT Security 2.0:

- 2.1.3 (Establish a secure ICT architecture) — pre-boot authentication requirement
- 2.3.1 (Protect devices and software) — directly affected
- 3.3.4 (Analyse data from security monitoring) — supplement with custom detections

NIST Cybersecurity Framework 2.0:

- PR.DS-01 (The confidentiality, integrity, and availability of data-at-rest are protected) — assumption chal-lenged
- PR.PS-01 (Configuration management practices are established) — baseline must be updated

5. Timeline and Follow-up

Date	Event
May 12, 2026	PoC published on GitHub by the researcher Nightmare-Eclipse
May 13, 2026	Confirmed by Will Dormann (Tharros Labs) and Kevin Beaumont
May 14, 2026	Broader media coverage; no official Microsoft response
May 17, 2026	Advisory v1.0 issued
May 22, 2026	Advisory v1.1 issued — technical corrections (this revision)
June 2026 (expected)	Microsoft Patch Tuesday — researcher has signalled a "big surprise"

6. References

The Hacker News — Windows Zero-Days Expose BitLocker Bypasses And CTFMON Privilege Escalation
https://thehackernews.com/2026/05/windows-zero-days-expose-bitlocker.html

BleepingComputer — Windows BitLocker zero-day gives access to protected drives
https://www.bleepingcomputer.com/news/security/windows-bitlocker-zero-day-gives-access-to-protected-drives-poc-released/

Cyber Unit — Windows BitLocker Zero-Day (YellowKey): What the WinRE Bypass Means
https://cyberunit.com/insights/windows-bitlocker-zero-day-yellowkey-winre-bypass/

Hive Security — YellowKey: The BitLocker Bypass Hidden in Windows Recovery
https://hivesecurity.gitlab.io/blog/yellowkey-bitlocker-bypass-winre-windows-11/

Patch Window — BitLocker zero-day: no patch, PoC is public
https://patchwindow.serverdigital.net/brief/bitlocker-yellowkey-zero-day-no-patch

Cybernews — BitLocker bypass zero-day exploit released
https://cybernews.com/security/researcher-releases-bitlocker-bypass-and-privilege-escalation-exploit/

Microsoft Learn — BitLocker countermeasures
https://learn.microsoft.com/en-us/windows/security/operating-system-security/data-protection/bitlocker/countermeasures

Microsoft Learn — BitLocker CSP
https://learn.microsoft.com/en-us/windows/client-management/mdm/bitlocker-csp

CIS Critical Security Controls v8
https://www.cisecurity.org/controls/v8

NSM Basic Principles for ICT Security 2.0
https://nsm.no/regelverk-og-hjelp/rad-og-anbefalinger/grunnprinsipper-for-ikt-sikkerhet/

7. About nLogic

nLogic AS is a trusted Nordic partner for high-performance network and security solutions. We help organizations secure critical infrastructure through architectures grounded in Zero Trust principles, the CIS Critical Security Controls, NSM's Basic Principles for ICT Security, and the NIST Cybersecurity Framework.

This advisory was prepared by Kim Hansen, Security Advisor, nLogic AS. For tailored guidance on applying these recommendations to your environment, please contact your nLogic representative.

Disclaimer

This document is a technical advisory, not a legal or compliance statement. Changes to BitLocker, WinRE or firmware configuration must be validated against your organisation's risk profile and servicing routines before rollout.

Concerned about your current security posture and potential risks? Contact nLogic for an initial conversation.

Thomas Brodersen, IT Security Advisor, nLogic
thomas.brodersen@nlogic.no
+47 958 30 108

Powered by Knowledge. Driven by Trust.

nLogic