real world technology solutions

INCIDENT RESPONSE GUIDE

M365 OneDrive Phishing Incident Response

1.0

A step-by-step guide for IT teams whose Microsoft 365 users have been hit by the SharePoint sharing phishing campaign. Triage, contain, investigate,

AUTHOR

Andrew Yager, RWTS

DATE

14 May 2026

About this guide

This guide is for IT staff and admins responsible for a Microsoft 365 tenant where a user has clicked a OneDrive phishing link. It is the same playbook we use internally at Real World Technology Solutions when we respond to this incident for our clients — published so anyone in the same position can work through it.

The current campaign hitting Sydney organisations — particularly Christian church networks — is unusual in that the bait email is sent through a real, compromised Microsoft 365 account via SharePoint's own file-sharing notifications. The email passes SPF, DKIM, and DMARC because it is a legitimate Microsoft sharing email. Your spam filter will not catch it. The mechanics are described below.

Work through the phases in order. Don't skip containment to go investigating — every minute you spend hunting is another minute the attacker has valid tokens, active rules, and ongoing access.

> **NOTE**
>
> If your tenant has Huntress Identity Threat Detection and Response (ITDR) or a similar identity-focused security tool deployed, check that portal first. It will likely have already alerted on the inbox rule creation and risky sign-ins, and may have auto-isolated the affected account. Use this guide to verify the tool's actions, or to work the incident if you don't have one of those tools.

Threat summary

The mechanic

One mailbox in a Microsoft 365 tenant is compromised, usually because someone earlier in the chain clicked the same OneDrive phish. The attacker uses the tenant's own SharePoint to share a file with everyone in the contact list. The sharing notification email is sent by Microsoft and passes SPF, DKIM, and DMARC because it is a legitimate Microsoft sharing email. The shared "file" links to a credential-harvest page imitating OneDrive. Anyone who logs in to view it hands over their M365 password.

Post-compromise, the attacker:

- Adds Outlook inbox rules that hide replies, non-delivery reports, and security alerts (move to "RSS Feeds", Archive, Deleted Items, or Junk; mark as read).
- Sets mailbox-level forwarding to an external address in some cases.
- Waits, sometimes days, before sending the next round of phish — defeating "unusual sending pattern" detection.
- Uses the newly compromised account to send the next round of sharing notifications, and the cycle repeats.

Indicators of compromise

Email indicators (inbound to your users):

- Subject contains "shared a file with you", "shared '...' with you", or "A protected File was sent to you".
- From: a legitimate Microsoft-sent sharing notification on behalf of someone the user knows.
- Body: a sharing card with an "Open" or "Read the message" button.
- The link does not resolve to `*.sharepoint.com` or `*.onedrive.com`.
- The bait landing page has Microsoft branding but security-vendor "file security" badges (ESET, Bitdefender, etc.) in the footer — a giveaway the page isn't really from Microsoft.

Tenant indicators (compromised accounts):

- Inbox rules with names like a single letter, "..", or generated GUIDs.
- Rules that move messages to RSS Feeds, Archive, Deleted Items, Junk, or Conversation History.
- Rules that forward externally.
- Sign-ins from countries the organisation does not operate in.
- Sign-ins from data-centre IP ranges. Look up the IP — Azure, AWS, OVH, and DigitalOcean are common attacker infrastructure.
- A burst of SharePoint sharing events from a single user inside a short window.
- Mailbox forwarding set on accounts that shouldn't have it.

Initial assessment

Before you touch anything, get the facts. If the affected user reported the incident, walk through these questions with them. If you noticed it through monitoring, work out as many as you can from logs first.

1. Who is the affected user? What's their UPN, their role, and do they have any privileged access (Global Admin, Exchange Admin, SharePoint Admin)?
2. What exactly did they click? Get the URL if possible, or a screenshot of the email.
3. What did they enter on the page? Just password, or password and MFA code?
4. When did they click? Time is critical for audit-log queries.
5. Have they noticed anything unusual since? Missing emails, replies they didn't send, password change notifications, sign-in alerts.
6. Has anyone else in the organisation reported the same email?
7. What MFA is enforced for this user? App, SMS, hardware key, or none.

> **WARNING**
>
> If the user entered an MFA code on the fake page, assume the attacker has a valid session token — not just the password. Containment must include revoking sign-in sessions, not just resetting the password. Password reset alone does not invalidate existing tokens.

Containment (do these first, within minutes)

Connect to the relevant Microsoft services first. You'll need all three for the full workflow:

```
Connect-ExchangeOnline -UserPrincipalName admin@yourtenant.onmicrosoft.com
Connect-IPPSSession    -UserPrincipalName admin@yourtenant.onmicrosoft.com
Connect-MgGraph -Scopes
"User.ReadWrite.All","Directory.AccessAsUser.All","AuditLog.Read.All"
```

Then run containment in this order:

1. Reset the affected user's password to a strong temporary value and force change on next sign-in.
2. Revoke all active sign-in sessions. This is what kills the attacker's existing tokens.
3. If the user is non-essential for the next hour, disable the account briefly to force full token re-issue.
4. Block the user's sign-in from unfamiliar locations via Conditional Access if you have Entra ID P1 or above and no policy already exists.
5. Tell the affected user not to sign back in until you say so. Communicate by phone, not email — their inbox is the problem.

Reset password and revoke sessions:

```
$UPN = "user@yourtenant.com.au"

# Generate a strong temporary password
Add-Type -AssemblyName System.Web
$tempPassword = [System.Web.Security.Membership]::GeneratePassword(20,4)
Write-Host "Temp password (give to user via phone, not email): $tempPassword"

# Reset password via Graph
$passwordProfile = @{
    Password = $tempPassword
    ForceChangePasswordNextSignIn = $true
}
Update-MgUser -UserId $UPN -PasswordProfile $passwordProfile

# Revoke all sign-in sessions (invalidates refresh tokens)
Revoke-MgUserSignInSession -UserId $UPN

# Optional: bounce the account to force a clean state (60-second window)
Update-MgUser -UserId $UPN -AccountEnabled:$false
Start-Sleep -Seconds 60
Update-MgUser -UserId $UPN -AccountEnabled:$true
```

Investigation

Once contained, hunt for what the attacker did while they were in. Work through the per-user checks first, then the tenant-wide scan if you have any reason to suspect more than one account is compromised — and if one user clicked, others probably did too.

Per-user: inbox rules

```
$UPN = "user@yourtenant.com.au"

Get-InboxRule -Mailbox $UPN |
    Select-Object Name, Description, Enabled,
        RedirectTo, ForwardTo, ForwardAsAttachmentTo,
        MoveToFolder, DeleteMessage, MarkAsRead |
    Format-List
```

Look for any rule the user can't explain. Specifically: single-character or nonsense names, rules that move mail to Deleted Items / RSS Feeds / Archive / Junk, rules that mark mail as read, and rules that forward externally.

Per-user: mailbox-level forwarding

```
Get-Mailbox -Identity $UPN |
    Select-Object DisplayName, UserPrincipalName,
        ForwardingSmtpAddress, ForwardingAddress,
        DeliverToMailboxAndForward
```

Per-user: recent sign-ins

```
$startDate = (Get-Date).AddDays(-30).ToString('yyyy-MM-ddTHH:mm:ssZ')

Get-MgAuditLogSignIn -Filter "userPrincipalName eq '$UPN' and createdDateTime ge $startDate"
-All |
    Select-Object CreatedDateTime,
        IpAddress,
        @{N='Location';E={"$($_.Location.City), $($_.Location.CountryOrRegion)"}},
        @{N='App';E={$_.AppDisplayName}},
        @{N='Client';E={$_.ClientAppUsed}},
        @{N='Status';E={$_.Status.ErrorCode}} |
    Sort-Object CreatedDateTime -Descending |
    Format-Table -AutoSize
```

Flag sign-ins from countries you don't operate in, sign-ins from data-centre ASNs (use `whois` or ipinfo.io on the IP), and sign-ins from non-Outlook clients you don't expect (PowerShell, "Office 365 Exchange Online", or "Microsoft Office" from an odd location).

Tenant-wide: suspicious inbox rules

If multiple users are likely compromised, scan the whole tenant. This takes a while on large tenants — run it in a screen or tmux session.

```
$mailboxes = Get-Mailbox -ResultSize Unlimited
$suspicious = foreach ($mbx in $mailboxes) {
    Get-InboxRule -Mailbox $mbx.UserPrincipalName -ErrorAction SilentlyContinue |
        Where-Object {
            $_.ForwardTo -or
            $_.ForwardAsAttachmentTo -or
            $_.RedirectTo -or
            ($_.MoveToFolder -and $_.MoveToFolder -match 'Deleted|Archive|RSS|Junk|
Conversation') -or
            $_.DeleteMessage -eq $true -or
            $_.MarkAsRead -eq $true
        } |
        Select-Object @{N='Mailbox';E={$mbx.UserPrincipalName}},
            Name, Description, Enabled,
            ForwardTo, RedirectTo, MoveToFolder,
            DeleteMessage, MarkAsRead
}

$suspicious | Export-Csv -Path .\suspicious-inbox-rules.csv -NoTypeInformation
$suspicious | Format-Table -AutoSize
```

Tenant-wide: mailbox forwarding

```
Get-Mailbox -ResultSize Unlimited |
    Where-Object { $_.ForwardingSmtpAddress -or $_.ForwardingAddress } |
    Select-Object DisplayName, UserPrincipalName,
        ForwardingSmtpAddress, ForwardingAddress,
        DeliverToMailboxAndForward |
    Export-Csv -Path .\mailbox-forwarding.csv -NoTypeInformation
```

Tenant-wide: audit log for inbox rule creation

```
$startDate = (Get-Date).AddDays(-30)
$endDate   = Get-Date

Search-UnifiedAuditLog `
    -StartDate $startDate -EndDate $endDate `
    -Operations "New-InboxRule","Set-InboxRule","UpdateInboxRules","Set-Mailbox" `
    -ResultSize 5000 |
    Export-Csv -Path .\audit-rule-events.csv -NoTypeInformation
```

Tenant-wide: SharePoint sharing activity for the compromised user

```
Search-UnifiedAuditLog `
    -StartDate (Get-Date).AddDays(-14) -EndDate (Get-Date) `
    -Operations
"SharingSet","SharingInvitationCreated","AnonymousLinkCreated","SecureLinkCreated" `
```

```
    -UserIds $UPN -ResultSize 5000 |
    Export-Csv -Path .\sharing-events.csv -NoTypeInformation
```

This tells you what the attacker shared and with whom. Those recipients need to be warned — and if they're inside the same tenant, their mailboxes need investigating too.

Remediation

Remove malicious inbox rules

For a single rule you've already identified:

```
Remove-InboxRule -Mailbox $UPN -Identity "RuleName" -Confirm:$false
```

For everything matching the CSV from the tenant-wide scan, manually review the file first, then remove in bulk:

```
$confirmed = Import-Csv .\suspicious-inbox-rules-reviewed.csv  # after manual review
foreach ($r in $confirmed) {
    Remove-InboxRule -Mailbox $r.Mailbox -Identity $r.Name -Confirm:$false
}
```

Remove mailbox-level forwarding

```
Set-Mailbox -Identity $UPN `
    -ForwardingSmtpAddress $null `
    -ForwardingAddress $null `
    -DeliverToMailboxAndForward $false
```

Purge the bait emails tenant-wide

Use a Content Search in Purview to find every copy of the phish across every mailbox in the tenant, preview the results, then purge. This is the single most useful action you can take — it stops more users from clicking the link.

```
# Connect-IPPSSession first if you haven't already

# Build the search
New-ComplianceSearch -Name "OneDrivePhish-2026-05" `
    -ExchangeLocation All `
    -ContentMatchQuery '(subject:"shared a file with you" OR subject:"protected File was
sent to you") AND received>=2026-04-30'

Start-ComplianceSearch -Identity "OneDrivePhish-2026-05"

# Poll until complete
do {
    Start-Sleep -Seconds 10
    $s = Get-ComplianceSearch -Identity "OneDrivePhish-2026-05"
    Write-Host "Status: $($s.Status) — Items: $($s.Items)"
} while ($s.Status -ne "Completed")

# Preview matches
$s.SuccessResults
# Or via the Purview portal for a richer view
```

> **WARNING**
>
> Preview the search results before purging. You can only purge messages you've identified accurately. Tune the `ContentMatchQuery` if it's pulling in legitimate sharing notifications — narrow it by adding the suspect sender domain, the bait URL pattern, or a date window.

Once you're confident the search only matches the phish, purge:

```
# Soft delete (recoverable for 14 days)
New-ComplianceSearchAction -SearchName "OneDrivePhish-2026-05" -Purge -PurgeType SoftDelete

# Or hard delete (unrecoverable)
New-ComplianceSearchAction -SearchName "OneDrivePhish-2026-05" -Purge -PurgeType HardDelete
```

SoftDelete is the right default. Use HardDelete only if the bait clearly contains malicious payloads you want gone permanently.

Notify external recipients of the original phish

The recipients of phish that went out from your compromised user need to be warned. Pull the recipient list from the SharePoint sharing audit (see Investigation) and send a templated warning. Factual and short works best:

> **SUGGESTED WORDING**
>
> Subject: Possible phishing email from us — please disregard
>
> Hi — Our team has identified that a phishing email was sent from one of our Microsoft 365 accounts on [date]. If you received a message that looked like a OneDrive "shared a file with you" notification from [name], please do not click the link. If you've already clicked and entered your password, please change it immediately, sign out of all devices, and check your inbox rules. Sorry for the inconvenience — we've contained the original compromise and are working with our IT team to make sure it doesn't happen again.

Hardening (post-incident)

Don't close the ticket without doing these. Otherwise you'll be back inside 30 days.

1. Enforce MFA on every account in the tenant. App-based or hardware key. SMS is better than nothing but increasingly phishable.
2. Create a Conditional Access policy blocking legacy authentication (IMAP, POP, Basic Auth, "Other clients"). Requires Entra ID P1 or above.
3. Create a Conditional Access policy requiring MFA from outside Australia, or block sign-ins from countries you don't operate in.
4. Enable Unified Audit Log if it isn't already on. It's on by default for new tenants but old ones may have it off.
5. Consider deploying identity threat detection such as Huntress ITDR. The licensing is per-mailbox and the detection-to-remediation time is the difference between this being a 4-hour cleanup and a 4-week one.
6. Run a phishing-awareness brief for your users. The blog post that accompanies this guide is written for a non-technical audience — share it.

Communicating with affected parties

Three groups need to hear from you:

- **The compromised user.** Phone first — their inbox is the problem. Tell them what happened, what you've done, and what they need to do (sign back in with the new password, set up MFA properly, review their sent items for anything they didn't send).
- **Other users in your organisation.** A short notice telling them what the phish looks like, what to do if they think they clicked it, and where to report suspected phishing internally. Send it from a different sender than the compromised account.
- **External recipients of the phish.** The template above. Send from a clean account, not the compromised one.

Escalation and external help

- **Huntress ITDR support:** 24/7 SOC. Use them if your tenant has Huntress and you need help interpreting an alert chain or want them to take action.
- **Microsoft 365 support:** open a Pro or Premier support case if you see evidence of broader tenant compromise — admin account takeover, persistent app registrations, OAuth grant abuse, or you've lost access to admin accounts. Microsoft Defender for Office 365 P2 customers can request Customer Incident Response Team (CIRT) involvement.
- **ACSC (Australian Cyber Security Centre):** report through ReportCyber at cyber.gov.au/report. Recommended for any business email compromise, mandatory for essential infrastructure and government bodies. Reporting helps the ACSC track campaigns and warn others.
- **Your cyber insurance carrier:** notify them early. Many policies require notification within a specific window for cover to apply, and the carrier may have their own preferred incident response provider they want involved.

If you'd rather not work through this alone

Real World Technology Solutions does this kind of incident response work for Sydney businesses, churches, and not-for-profits. If you need an extra pair of hands on your tenant — or you'd rather hand the whole thing over — call us on **1300 798 718** or email **support@rwts.com.au**.

If you'd rather not need us next time, the same team can audit your Microsoft 365 tenant against the hardening checklist above, lock down Conditional Access and MFA, and deploy ongoing identity monitoring.

Appendix: PowerShell quick reference

All the commands above in a single block, ready to copy-paste once you've set `$UPN`:

```
# --- Connect ---
Connect-ExchangeOnline -UserPrincipalName admin@yourtenant.onmicrosoft.com
Connect-IPPSSession    -UserPrincipalName admin@yourtenant.onmicrosoft.com
Connect-MgGraph -Scopes
"User.ReadWrite.All","Directory.AccessAsUser.All","AuditLog.Read.All"

$UPN = "user@yourtenant.com.au"

# --- Contain ---
Add-Type -AssemblyName System.Web
$tempPassword = [System.Web.Security.Membership]::GeneratePassword(20,4)
$passwordProfile = @{ Password = $tempPassword; ForceChangePasswordNextSignIn = $true }
Update-MgUser -UserId $UPN -PasswordProfile $passwordProfile
Revoke-MgUserSignInSession -UserId $UPN
"Temp password: $tempPassword"

# --- Investigate (per user) ---
Get-InboxRule -Mailbox $UPN | Format-List Name, Description, Enabled, RedirectTo, ForwardTo,
MoveToFolder, DeleteMessage, MarkAsRead
Get-Mailbox -Identity $UPN | Format-List ForwardingSmtpAddress, ForwardingAddress,
DeliverToMailboxAndForward
$start = (Get-Date).AddDays(-30).ToString('yyyy-MM-ddTHH:mm:ssZ')
Get-MgAuditLogSignIn -Filter "userPrincipalName eq '$UPN' and createdDateTime ge $start" -
All |
    Select-Object CreatedDateTime, IpAddress, @{N='Location';E={"$($_.Location.City), $
($_.Location.CountryOrRegion)"}}, AppDisplayName, ClientAppUsed |
    Sort-Object CreatedDateTime -Descending | Format-Table

# --- Investigate (tenant) ---
$mbx = Get-Mailbox -ResultSize Unlimited
$suspect = foreach ($m in $mbx) {
    Get-InboxRule -Mailbox $m.UserPrincipalName -ErrorAction SilentlyContinue |
        Where-Object { $_.ForwardTo -or $_.RedirectTo -or ($_.MoveToFolder -match 'Deleted|
Archive|RSS|Junk') -or $_.DeleteMessage -or $_.MarkAsRead } |
        Select-Object @{N='Mailbox';E={$m.UserPrincipalName}}, Name, ForwardTo, RedirectTo,
MoveToFolder, DeleteMessage
}
$suspect | Export-Csv .\suspicious-inbox-rules.csv -NoTypeInformation

# --- Remediate ---
Remove-InboxRule -Mailbox $UPN -Identity "RuleName" -Confirm:$false
Set-Mailbox -Identity $UPN -ForwardingSmtpAddress $null -ForwardingAddress $null -
DeliverToMailboxAndForward $false

# --- Purge bait emails ---
New-ComplianceSearch -Name "OneDrivePhish-2026-05" -ExchangeLocation All `
    -ContentMatchQuery '(subject:"shared a file with you" OR subject:"protected File was
sent to you") AND received>=2026-04-30'
Start-ComplianceSearch -Identity "OneDrivePhish-2026-05"
# wait for status Completed, preview, then:
New-ComplianceSearchAction -SearchName "OneDrivePhish-2026-05" -Purge -PurgeType SoftDelete
```

Document control

Version	1.0
Author	Andrew Yager, Real World Technology Solutions
First issued	14 May 2026
Audience	IT professionals managing Microsoft 365 tenants
Licence	Share freely with attribution. Not legal or formal cyber-incident advice — for that, engage a qualified provider.
Companion	Plain-English blog: Microsoft 365 phishing is now coming from real accounts. Here's how to spot it.