Barracuda Managed XDR™

Barracuda XDR PoC
Threat Simulation Guide

Barracuda Managed XDR™

Table of Contents

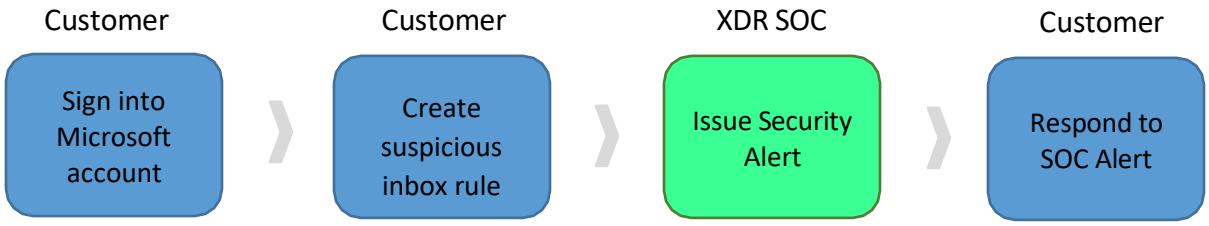
XDR Cloud Security: Suspicious Inbox Rule

Rule:
Microsoft365 Suspicious New Inbox Rule Created

Purpose:
This detection monitors Microsoft365 accounts for new suspicious inbox rule creation. After gaining access to a victim's account, threat actors will often create an inbox rule inside their mailbox to maintain stealthy access. This inbox rule can do anything a normal inbox rule could but is usually used to forward emails matching sensitive keywords, like 'invoice' or 'payment', to an external email address controlled by the attacker. It also is usually created as a hidden inbox rule. These tactics allow the threat actor to intercept financial information pertaining to the compromised user/organization and their contacts, all without the impacted user's knowledge.

Objective:
Test detection of a suspicious hidden inbox rule creation.

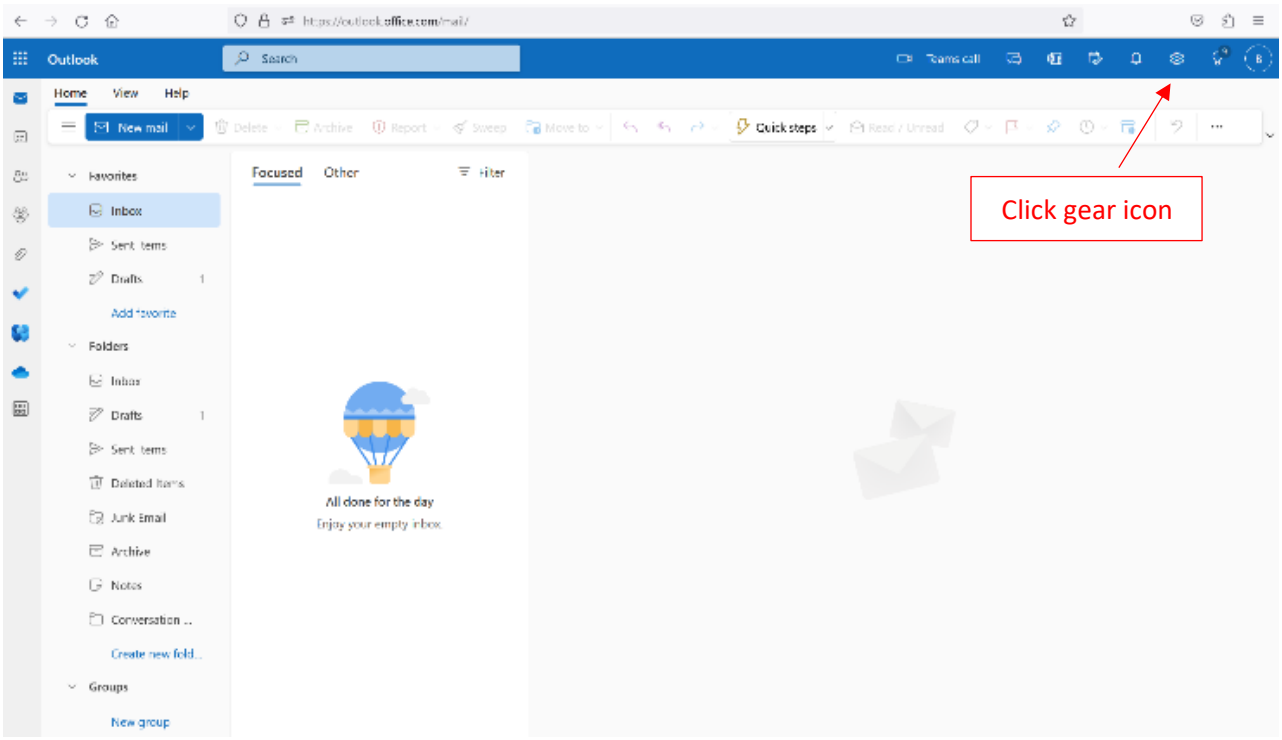
Test Workflow:



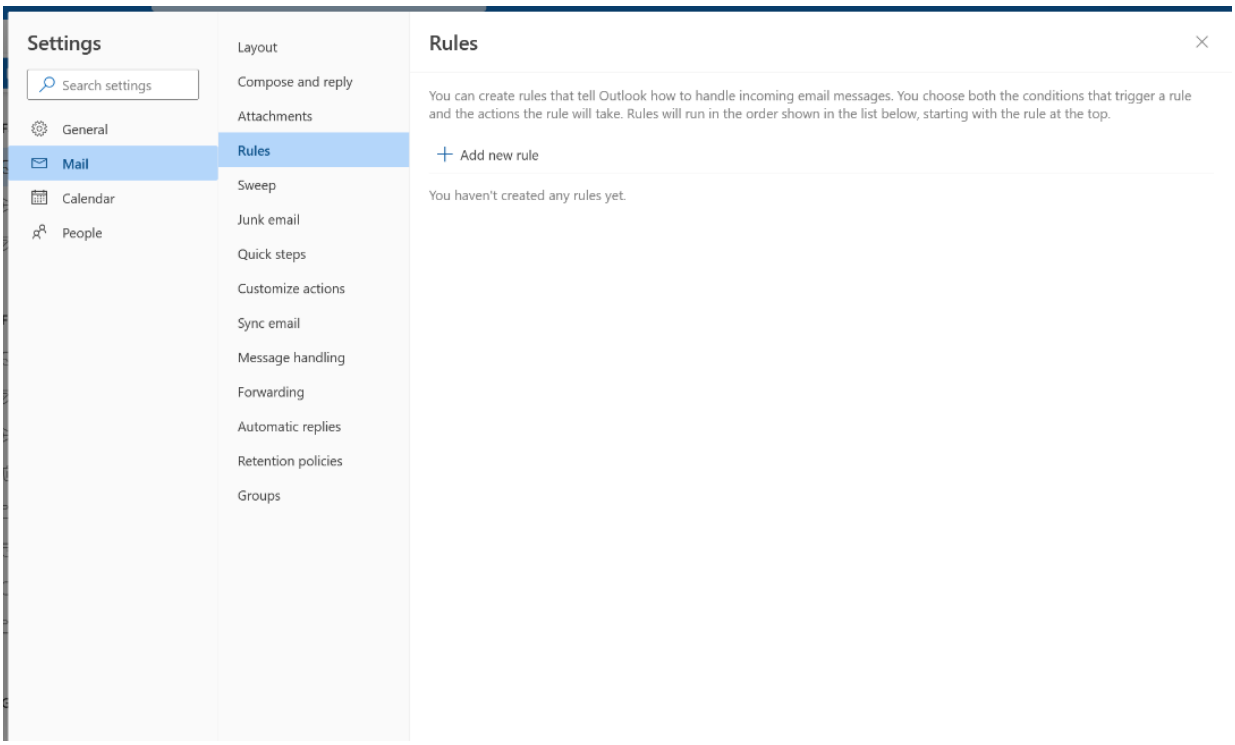
How to test:

1. Navigate to Outlook.com and sign in.

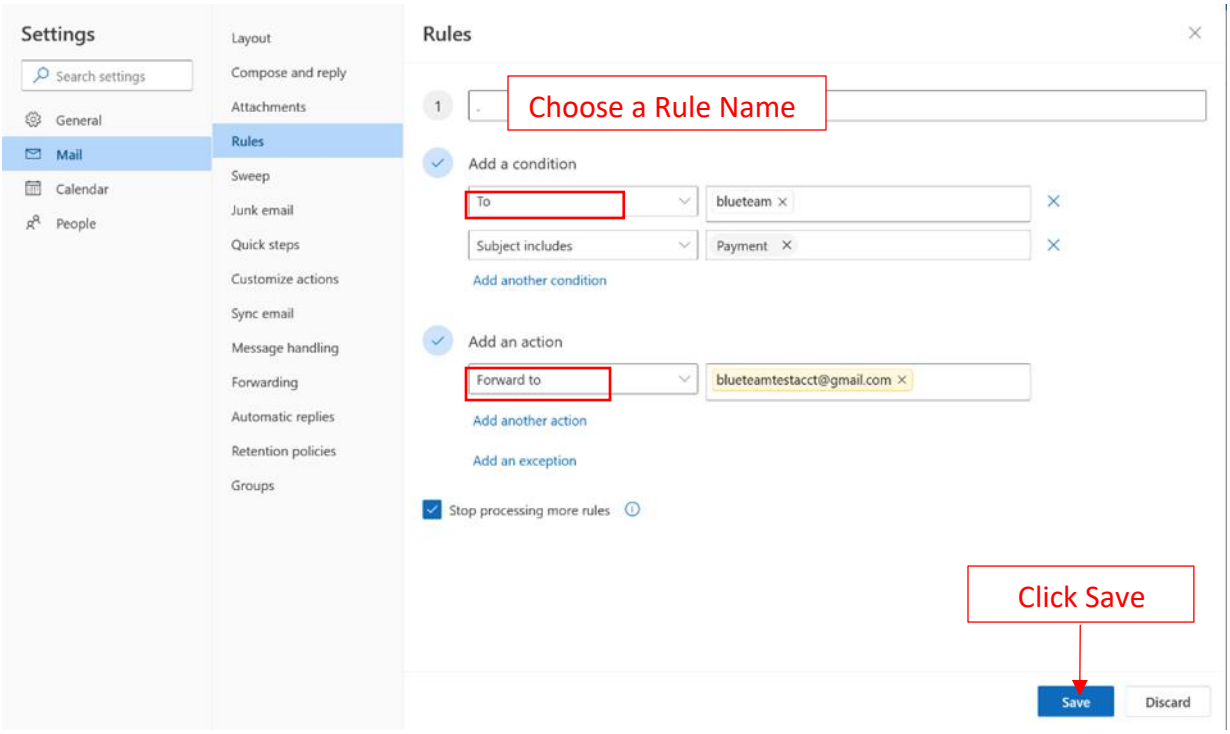
2. Click the gear icon in the top right corner.



3. Click “Mail” in the left-hand menu.
4. Choose “Rules” under the Mail section.
5. Click “+ Add new rule.”



6. Give your rule a name, such as “.” in order to make this a hidden inbox rule.
7. Set the conditions for your rule to look for messages containing keywords such as “payment”.
8. Specify the actions to be performed on matched emails, such as forwarding to another account.
9. Click “Save” to add the rule.



10. A Barracuda XDR alert will trigger from the SOC describing this impossible travel scenario. The alert can be viewed via the Barracuda XDR Security Dashboard.
11. We request that you reply to the security alert stating that the reported activity was associated with authorized security testing.
12. The SOC team will close the incident, marking the conclusion of this threat simulation test.

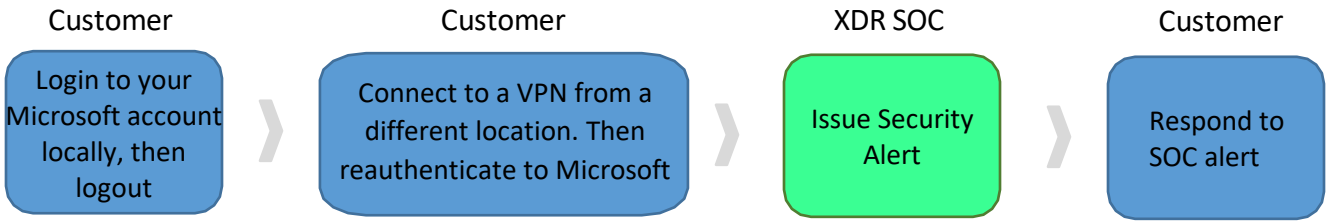
XDR Cloud Security: Impossible Travel

Rule:
Microsoft365 Impossible Travel

Purpose:
The impossible travel detection identifies two user activities (in a single or multiple sessions) originating from geographically distant locations within a period shorter than the time it would have taken the user to travel from the first location to the second, indicating that an unauthorized actor is accessing the same account. This is a key indicator that an account has been compromised.

Objective:
Detect impossible travel between two geolocations within a short period of time.

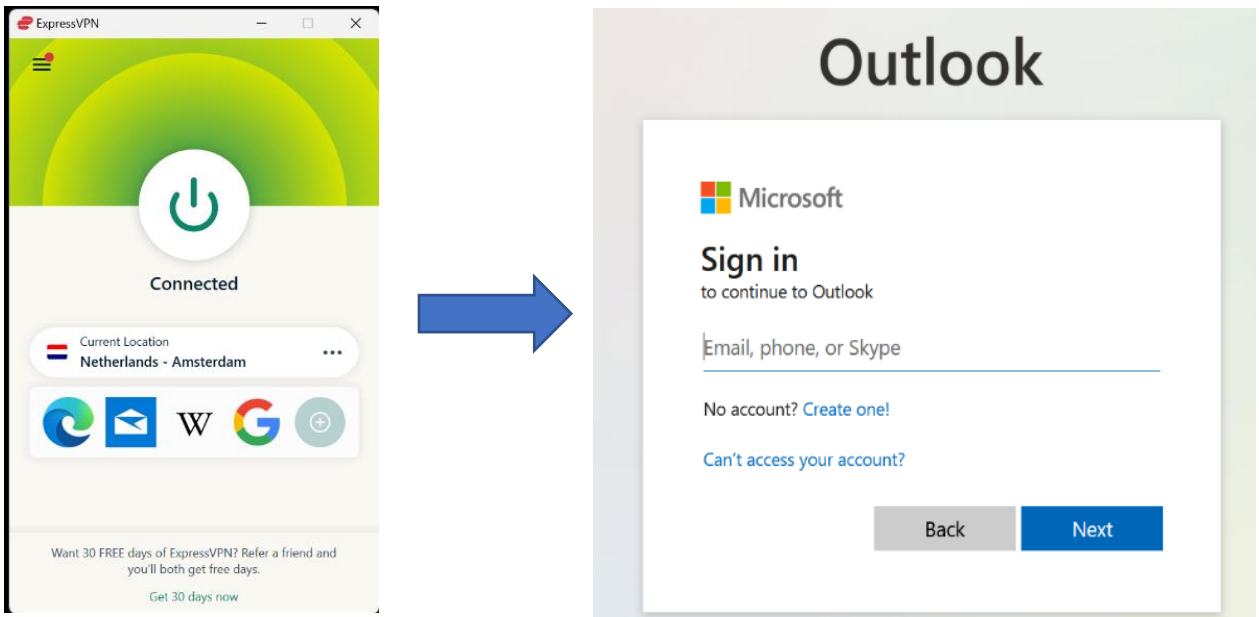
Test Workflow:



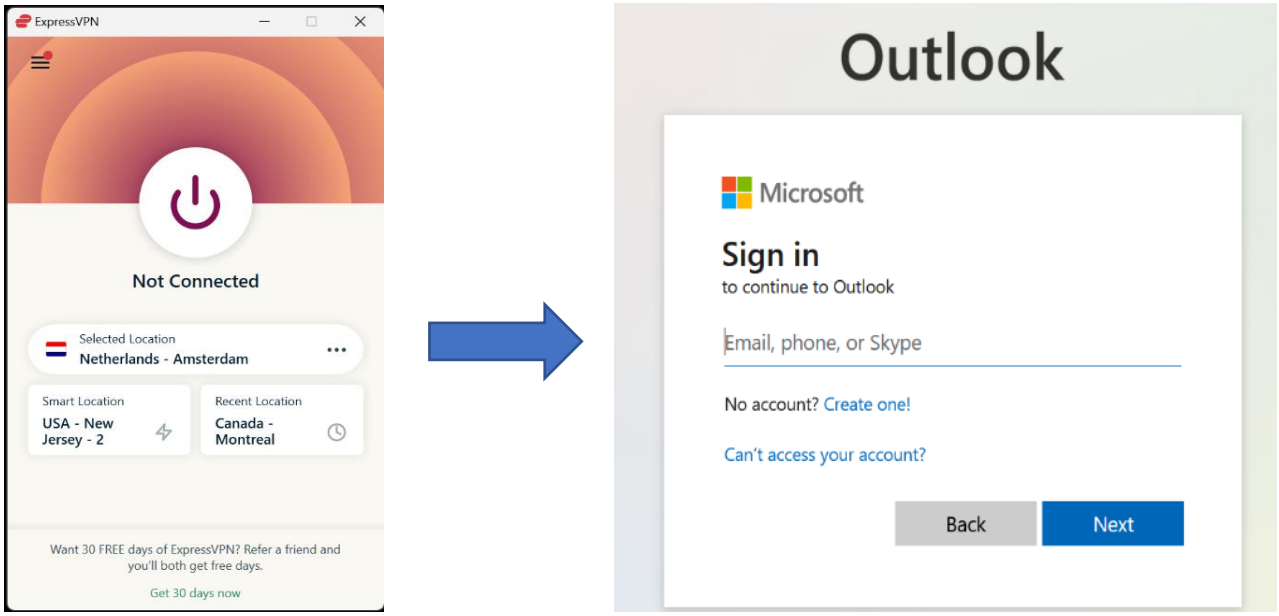
How to test:

1. Choose a VPN Tool such as ExpressVPN, NordVPN, etc. Download and install the VPN client on your device.
2. Once the application is downloaded, open it and you will be able to connect to a VPN server from a foreign location. Ex: Amsterdam, Netherlands

3. Once connected, login to your Microsoft365 account. After successfully authenticating, logout of your Microsoft account.



4. Then disconnect VPN, and immediately login to your Microsoft365 account again from your current location.



5. A Barracuda XDR alert will trigger from the SOC describing this impossible travel scenario. The alert can be viewed via the Barracuda XDR Security Dashboard.
6. We request that you reply to the security alert stating that the reported activity was associated with authorized security testing.
7. The SOC team will close the incident, marking the conclusion of this threat simulation test.

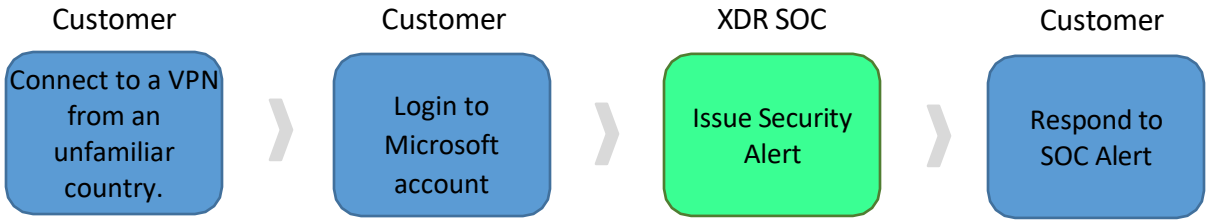
XDR Cloud Security: Conditional Access Policy Block from New Location

Rule:
Office 365 Conditional Access Policy Block from New Location

Purpose:
This alert generates when a Conditional Access Policy blocks user authentication originating in country the user has not previously authenticated from in the last 30 days.

Objective:
Verify detection when a Conditional Access Policy blocks a login from a new location.

Test Workflow:



How to test:

1. Ensure the test user has a Conditional Access Policy that restricts logins to familiar locations (e.g., specific countries or regions).
2. Use a VPN service to simulate a login attempt from a new, unfamiliar country (a location where the user has not logged in for the last 30 days).
3. Attempt to log in to the Office 365 account from the VPN endpoint.
4. A Barracuda XDR alert will trigger from the SOC. The alert can be viewed via the Barracuda XDR Security Dashboard.
5. We request that you reply to the security alert stating that the reported activity was associated with authorized security testing.
6. The SOC team will close the incident, marking the conclusion of this threat simulation test.

XDR Cloud Security: OneDrive Malware File Upload

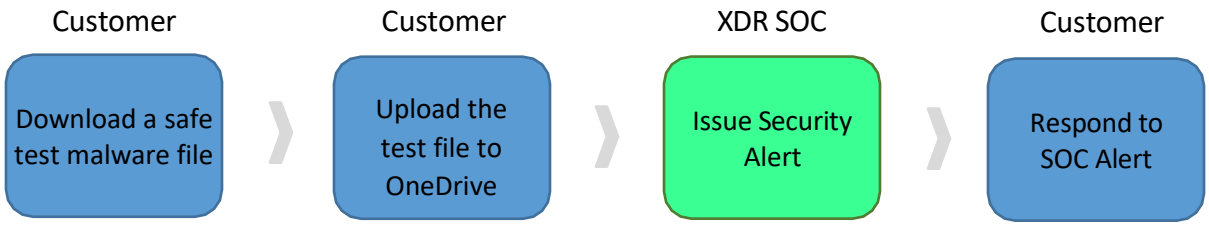
Rule:
Office 365 OneDrive Malware File Upload

Purpose:

Identifies the occurrence of files uploaded to OneDrive being detected as Malware by the file scanning engine. Attackers can use File Sharing and Organization Repositories to spread laterally within the company and amplify their access. Users can inadvertently share these files without knowing their maliciousness, giving adversaries opportunity to gain initial access to other endpoints in the environment.

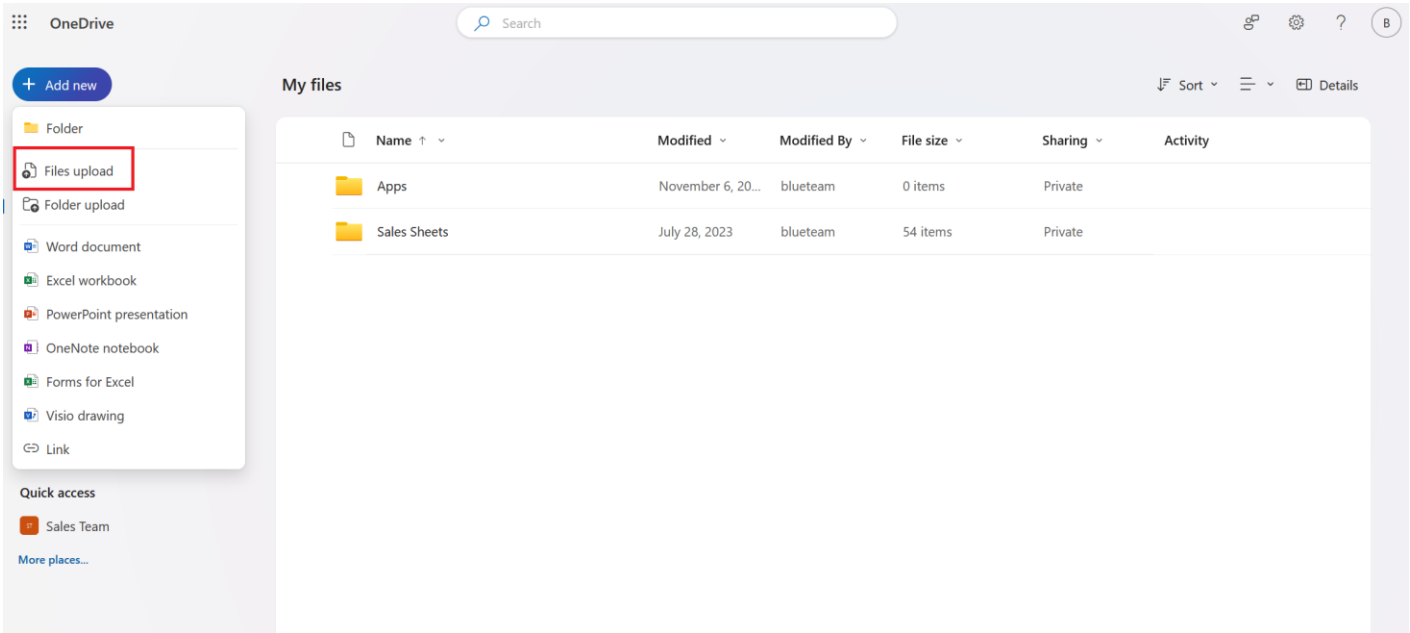
Objective:
Ensure that malware file uploads to OneDrive are detected.

Test Workflow:



How to test:

1. Download a safe test malware file (e.g., EICAR test file used for AV testing).
2. Log in to OneDrive with the test user account.
3. Upload the EICAR test file to the OneDrive account.



4. Allow the OneDrive file-scanning engine time to scan the file.

XDR Cloud Security: Two Factor Authentication Disabled

Rule:

Office 365 Two Factor Authentication Disabled

Purpose:
Detects when Two-factor authentication is disabled to a user account.

Objective:
Detect when Two-Factor Authentication (2FA) is disabled for an account.

How to test:

1. Log in to the Azure Active Directory or Office 365 admin portal using an administrator account.
2. Disable 2FA for the test user account:
3. Navigate to Users > Multi-Factor Authentication settings.
4. Find the test user and disable 2FA for their account.
5. Verify 2FA is disabled by attempting to log in to the test user's account without 2FA.

XDR Cloud Security: Unusual Volume of Emails Sent

Rule:
Office 365 Unusual Volume of Emails Sent

Purpose:
This detection triggers when 200+ emails have been sent out by this mailbox within 1 hour.

Objective:
Detect unusually high volume of emails.

How to test:

1. Use a test Office 365 account.
2. Create a script or manually send over 200 emails within one hour. This can be achieved using PowerShell or a bulk email-sending tool.
3. Example:

```
import smtplib
import time
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

# Office 365 SMTP server configuration
smtp_server = "smtp.office365.com"
smtp_port = 587
smtp_user = "your_email@domain.com"
smtp_password = "your_password"

# Email details
```

```
subject = "Test Email"
from_email = smtp_user
to_email = "recipient email@domain.com"
body_template = "This is test email number {}."

# Number of emails to send (e.g., 201 emails to trigger the alert)
num_emails = 201
interval_seconds = 17  # Time between emails (adjust to send 200+ emails within 1 hour)

def send_email(smtp_server, smtp_port, smtp_user, smtp_password, from_email, to_email,
subject, body):
    # Create a MIME message
    msg = MIMEMultipart()
    msg['From'] = from_email
    msg['To'] = to_email
    msg['Subject'] = subject

    # Attach the body to the email
    msg.attach(MIMEText(body, 'plain'))

    # Connect to the SMTP server and send the email
    try:
        with smtplib.SMTP(smtp_server, smtp_port) as server:
            server.starttls()  # Secure the connection
            server.login(smtp_user, smtp_password)
            server.sendmail(from_email, to_email, msg.as_string())
            print(f"Email sent to {to_email}")
    except Exception as e:
        print(f"Failed to send email: {e}")

if __name__ == "__main__":
    print(f"Starting to send {num_emails} emails...")

    for i in range(1, num_emails + 1):
        body = body_template.format(i)
        send_email(smtp_server, smtp_port, smtp_user, smtp_password, from_email,
to_email, subject, body)

        # Wait for the specified interval before sending the next email
        time.sleep(interval_seconds)

    print(f"Completed sending {num_emails} emails.")
```

XDR Cloud Security: Anomalous Login

Rule:
Microsoft 365 Anomalous Login

Purpose:
This detection identifies potentially compromised Office 365 accounts with sign-in scenarios that are anomalous in nature. In this case, we are looking at every unique sign-in and comparing them with the last 90 days usual sign-in characteristics of login for a user such as source geo city rarity, geo country rarity, source IP rarity, user agent rarity, distance travelled from the user's usual location of login, high confidence countries check and suspicious countries check to identify the anomaly using an ML model.

Objective:
Detect anomalous logins based on unusual activity patterns.

How to test:

1. Use the test user account to simulate an anomalous login by:
    a. Login from a rare geographic location (using a VPN).
    b. Using an unusual IP address or rare user agent string (e.g., different browser or device).

XDR Cloud Security: Brute Force Login Attempt

Rule:
Office 365 Brute Force Login Attempt

Purpose:
Detects an unusual condition where one source has 50 authentication failures for the same user within 15 minutes timeframe.

Objective:
Detect multiple failed login attempts (brute force).

How to test:

1. Use a test system to simulate 50 failed login attempts within a 15-minute window for the same user from the same source.
2. This can be scripted using tools like Hydra, Medusa, or a custom Python script
3. Example:

```
import requests
import time

# Office 365 login URL (adjust to your specific login endpoint)
login_url = "https://login.microsoftonline.com/common/login"

# Define test credentials
```

```
username = "testuser@yourdomain.com"
incorrect_password = "incorrect_password"

# Number of failed login attempts
num_attempts = 50

# Headers (you may need to modify this based on your login form)
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/85.0.4183.121 Safari/537.36",
    "Content-Type": "application/x-www-form-urlencoded",
}

# Payload template (adjust based on the actual form parameters of your Office 365 login
page)
payload_template = {
    "username": username,
    "password": incorrect_password,
    "login": "Sign in"
}

# Function to simulate a brute force attack
def brute_force_login():
    for i in range(num_attempts):
        response = requests.post(login_url, data=payload_template, headers=headers)

        # Log the response status
        if response.status_code == 200:
            print(f"Attempt {i+1}: Login attempt failed with status 200 (OK) - Incorrect
credentials.")
        else:
            print(f"Attempt {i+1}: Status Code {response.status_code}")

        # Adding delay between attempts (to avoid hitting rate limits)
        time.sleep(1)

if __name__ == "__main__":
    print("Starting brute force login simulation...")
    brute_force_login()
    print("Brute force login simulation completed.")
```

XDR Cloud Security: SharePoint Malware File Upload

Rule:

Office 365 SharePoint Malware File Upload

Purpose:
Identifies the occurrence of files uploaded to SharePoint being detected as Malware by the file scanning engine. Attackers can use File Sharing and Organization Repositories to spread laterally within the company and amplify their access. Users can inadvertently share these files without knowing their maliciousness, giving adversaries opportunity to gain initial access to other endpoints in the environment.

Objective:
Ensure malware file uploads to SharePoint are detected.

How to test:

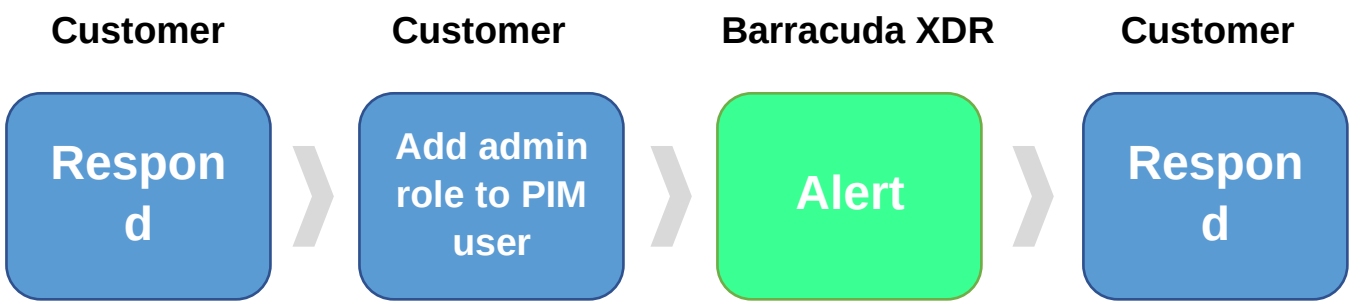
1. Download a safe test malware file (e.g., the EICAR test file).
2. Login to the test SharePoint environment with a test user account.
3. Upload the test malware file to SharePoint.
4. Allow time for the file scanning engine to scan the file.

XDR Cloud Security: PIM user granted administrator role in Azure

Purpose:
Allows for the ability to detect an administrator role being added to a PIM user. Threat actors might give an account they've compromised increased permissions as a means of persistence, just in case they get kicked out of the account they are currently using.

Objective:
Prevent increased permissions being added to an account that doesn't require them.

Test Workflow:



How to test:

1. Log into Azure using an administrator account.

2. In the search bar, search for the Users.
3. Click on the Users service.
4. Click the name of the user to modify.
5. On the left side of the screen, under the Manage tab, click Assigned roles.
6. Click Add Assignments.
7. Search for the administrator role you want to add, for example, Global Administrator.
8. Set the conditions for the role assignment.
9. Click Assign.