SECURE IT QUEST

In Conjunction with Ugli Scripts

Get Compromised Users from REST API
and
Remediate Office 365 Accounts

Revision Number	1.0
Date	August 19th, 2018
Prepared By	Stephen "Sully" Sullivan
	ssullivan@ugliscripts.com

Table of Contents

Disclaimer

These instructions and associated PowerShell script are provided "as is". Every effort was made to adhere to PowerShell best practices. Please keep in mind, this is a "Script" and not an "Application". As such, logging is limited to script actions and assumes underlying PowerShell, Windows Authentication and associated modules/methods are configured and functioning correctly.

Lastly, this script and document only covers Email related remediation and therefore should not be treated as an exhaustive solution. Endpoint, Proxy, Firewall and a number of other tools can and should be used.

Problem Statement

The growing number of Cloud Accounts being compromised and the resulting required remediation steps are creating a resource drain on Security Operations.

Problem Overview

There is a separation between security tools that identify compromised accounts and the tools available to remediate compromised accounts. Complete remediation requires a number of steps and can take a considerable amount of time and effort.

The disconnect often exists because security tools identify multiple accounts, remediation tools are designed around remediating a single account.

Project Objective

To leverage Targeted Attack Protection (TAP), Threat Response Auto-pull (TRAP) and Windows PowerShell to automate a comprehensive remediation plan.

Solution Summary

Targeted Attack Protection (TAP) provides alerting and tracking of User Mailbox threats. TAP support authenticated API access to alert, campaign and forensic detail.

Proofpoint Threat Response (PTR) Auto-Pull (TRAP) leverages the TAP APIs to manage alerts and provides many options for remediation. In addition to extensive built-in functionality (e.g. TRAP), PTR also provides a REST API.

For this use case we will only be discussing Threat Response Lists and the PTR API.

PTR can be configured to automatically add users to a "List". Members of a list can be retrieved via a secure REST API web GET request to the PTR server. Members can also be deleted with a secure web DELETE request. Additional detail regarding the PTR API can be access via the PTR Portal from a licensed PTR Console.

Windows Task Scheduler is used to execute a PowerShell script every *n* minutes. This script uses a web GET request to retrieve all members of the configured list, validates the account against AD users and verifies the AD User meets requirements. Validated AD Accounts will be remediated. A web DELETE request removes the user from the configured list.

Script logs are stored in a configured path on the PowerShell hosting the Scheduled task. Updates are also shown in the PTR Console.

Account Remediation Options:
- Force User to change password at next logon
- Change the Account password
- Enable Strong Password
- Enable Multi-Factor Authentication
- Disable external email forwarding rules
- Delete external forwarding rules
- Remove Delegates
- Disable Delegates
- Enable mailbox auditing

Basic Requirements

- I don't have access to Windows 2016 so this has only been tested on Windows 10.
- This version does not work on Windows 2012r2. I am working on a Windows 2012 version but assume most Azure AD organization will be on 2016, so I am posting the script now.
- Account with 'Administrator' access to Proofpoint Targeted Attack Protection (TAP)
- Licensed version of Proofpoint Threat Response Auto-Pull (TRAP) installed
  - https://ptr-docs.proofpoint.com/trap-guides/trap-installation/
- An Active-Directory Account to Modify user accounts
- Windows 10 Domain computer with PowerShell and PC Active-Directory Module
- The provided PowerShell Script

Configuration

TRAP installation and configuration is well documented, so I will fore go any repetition and assume TRAP is up and running. I recommend going through the TRAP set up first. It will get you acclimated to the PTR console and introduce some concepts we will use (e.g. Match Conditions).

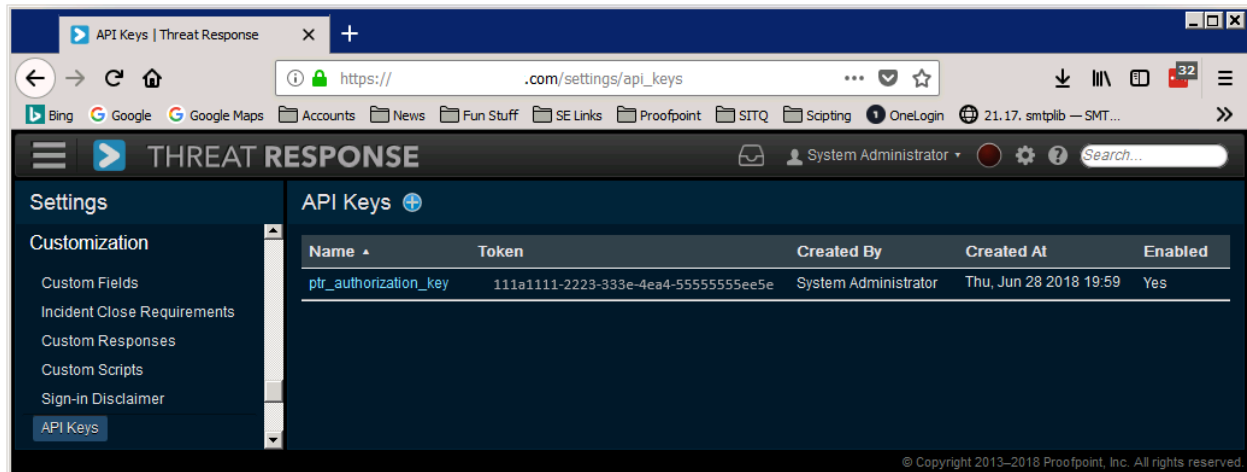Step 1: Configure a 'User List' in Threat Response

> When creating the 'User List', make a note of the List number (aka. Id) in the URL. We will use this number for our web GET and DELETE requests. In the example the List ID is '4' (figure 1.1).
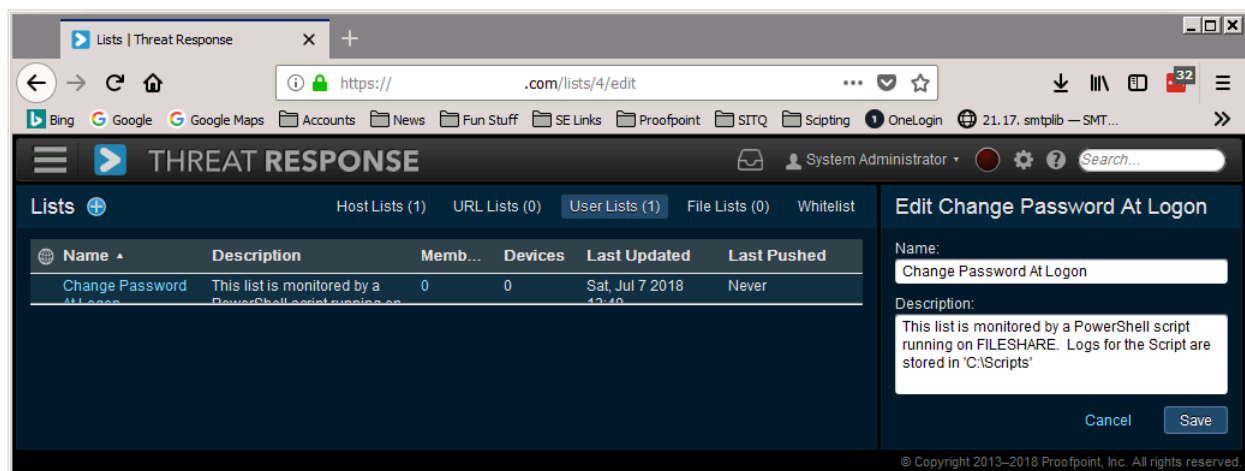
(figure 1.1)

Step 2: Generate a Threat-Response API key

> Logon to the Proofpoint Threat Response console. Access PTR settings. Under Customization and API Keys generate a new key and make a note of the Token (figure 2.1)

Step 3: Setting run-time parameters and testing PowerShell ISE

Login with the account the has access to Office 365 and will be used to schedule the script. Open PowerShell ISE as Administrator and open the provided scripts. The first time you run the script you will be prompted for configuration details (figure 3.1). The script creates 4 configuration files:

account.cred – encrypted account Name
password.cred – encrypted account Password
threatresponse.cred – encrypted Threat Response key
configuration.xml – stores options

The **.cred** items above are encrypted and only accessible to the user that created the files. This is why the scheduling user needs to be the same.

If you need to change any setting simply browse to the script folder and delete the appropriate file. If you are making basic change to the options, you can edit the **configuration.xml** directly.

```
PS C:\Windows\system32> C:\Scripts\PS for o365 Remediation\ptr-remediate-o365-account.ps1


    Directory: C:\Scripts\PS for o365 Remediation\Log


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        8/10/2018   9:45 AM              0 script-2018-08-10.log
Please enter the FQDN or IP of you Threat Response Server: app.example.com
Enter the Threat Response user list ID: 5
Enter the UPN suffix (e.g. @example.com): @example.com
Would you like to require users to change password at next logon? (Yes/No): Yes
Would you like to require a strong account password? (Yes/No): Yes
Would you like to ENABLE Multi-factor Authentication? (Yes/No): Yes
Would you like to change the account password? (Yes/No): Yes
Would you like to ENABLE Mailbox Auditing? (Yes/No): Yes
Would you like to REMOVE Mailbox Delegates? (Yes/No): Yes
Would you like to DISABLE external forwarding Inbox rules? (Yes/No): Yes
Would you like to REMOVE external forwarding Inbox rules? (Yes/No): Yes
Would you like to disable Mail forwarding? (Yes/No): Yes
Would you like to GET the Mailbox Audit log? (Yes/No): Yes
```
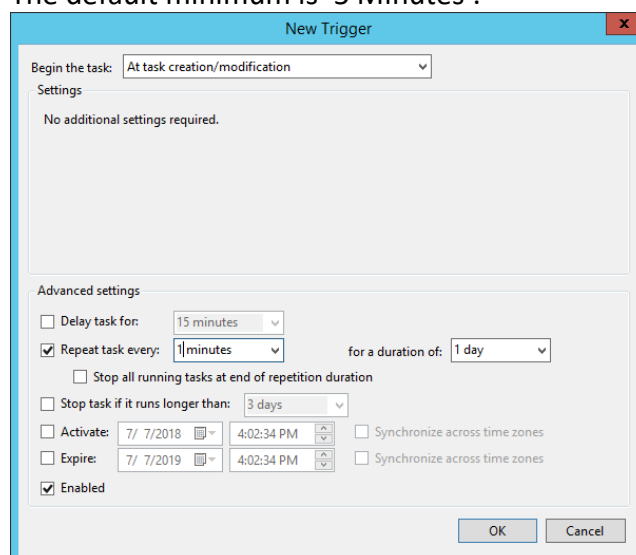
(figure 3.1)

Step 4: Create a 'Scheduled Task' to run the PS Script

Open 'Windows Task Scheduler' and select 'Create Task' (not 'Create Basic Task') under actions.

- General –
    - o Configure task to run as the 'PTR User'
    - o Select 'Run with highest privileges'
- Triggers –
    - o Configure as shown (figure 4.1)
    - o The default minimum is '5 Minutes'.

(figure 4.1)

- Action –
  Update the parameters below to reflect Script location. Use 'Start in' to determine where log file will be written.

  - Program/Script:
    'PowerShell.exe
  - Add Args:
    '-ExecutionPolicy Bypass C:\Scripts\PTR-remediate-o365-account.ps1 -RunType $true'
  - Start in:
    'C:\Scripts\
- Settings –
  Update 'Stop the task if it runs longer than:' to '1 hour'

Step 5: Create a 'Match Rule' to add Users to the List

Go to the TAP Source created during the TRAP set up. Create a new 'Match Rule' to match the Primary and Secondary objectives (figure 5.1).

- **Primary -** Force user who "click" on Phish, Malware and Imposter links to change password at next login.
  - Alert Type: 'Permitted Clicks'
- **Secondary –** Force users who have received attachments threats and unprotected URLs to change password at next logon.
  - Alert Type: 'Delivered Attachment Threats'
  - Alert Type: 'Unprotected URL Threats'
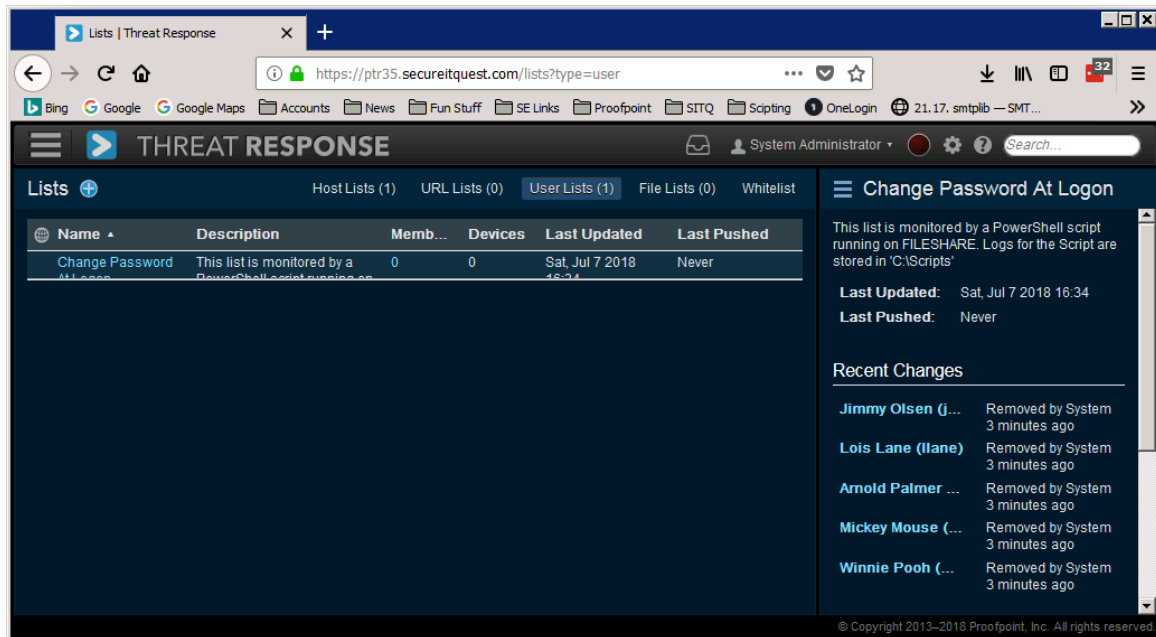

(figure 5.1)

Step 6: Validate functionality

You should now be able to manually add Users to the PTR List to validate functionality. Inside the PTR Console, within approximately ~5 minute you will see the User has been removed from the List. This means the User's Office 365 account has been remediated (figure 6.1).

(figure 6.1)