# Azure + ThreatWatch + Teams

*Scan and agent-less, always-on, prioritized full stack vulnerability management*

## Discovery and Inventory

Choice of Cloud Native Inventory ( Azure ) *OR* ThreatWatch twigs

## Deployment

Azure VM within your VPC *OR* On-Premise

## No-Agents and No-Scanning

No need to install agents or open network ports

## Corp to Prod

Protect Azure + Corporate Assets and Network devices

## De-centralized Ownership

Delegate visibility based on Azure resource groups or your organizational structure with out-of-the-box integration with MS Teams.

## Coverage

Virtual machines in cloud ( Azure ) or on-premise, physical hosts running any *supported OS, network devices, container images, source code repositories for Python, NodeJS, .NET, Go, Ruby and others.

# Getting Started

With three steps, you will get vulnerability detection , prioritization and reporting setup for your Azure or Multi-Cloud environment. In this section we will leverage Azure's cloud native VM discovery and inventory but you can achieve the same using twigs , an open-source CLI maintained by ThreatWatch for discovering assets in Azure or other hybrid environment.

## Step 1 Azure LogAnalytics and Automation Account Configuration

ThreatWatch leverages Azure's cloud native support for gathering asset inventory. This guide will take you through the steps required to get the automation account setup for inventory collection. **[URL]**

Your organizational assets can be spread across different Azure resource groups. The automation account is shared across different resource groups and is linked to a single log analytics workspace. If you have multiple Azure subscriptions, then the same steps will need to repeated for each subscription.

## Step 2 Provision a ThreatWatch instance for your organization

The ThreatWatch accounts team will provision an instance for your organization and would have a URL that would look similar to "yourorganization.threatwatch.io". The only external access required by this instance is for https (443), no other inbound or outbound access is required other than locked down ssh connectivity for troubleshooting as needed.

The ThreatWatch instance can be deployed and managed by ThreatWatch in a deployment environment of your choice ( Azure, on-prep of ThreatWatch cloud ). Once provisioned , accounts can be self provisioned. Every user can generate an API key for his / her account for operational tasks such as pulling inventory from Automation account ( described above )

## Step 3 Run *twigs*

ThreatWatch uses *twigs* ( an open source audit-able CLI maintained by ThreatWatch ) to gather asset inventory meta data from multitude of sources including Azure. *twigs* allows ThreatWatch to know enough about assets so that no direct access to your assets is ever needed. Running *twigs* at a periodic cadence helps keep the asset meta refreshed ( eg. post patching ).
Refer to the twigs guide for more information.

**How do I get twigs and where do I run it ?**
twigs support is available both on Linux ,Windows or Mac OSX. On Linux it's as easy as running the command, 'sudo pip install twigs'. On Windows there is a PowerShell executable that can be downloaded from ThreatWatch website.

*twigs* can be run literally anywhere ( on your laptop, on a windows / linux VM within your cloud, a jump-host etc ). It's recommended that you run twigs on a host that has access to your Azure cloud. So it can be a jump host or any VM within your cloud subscription.

*That's it, you are all set !*

## ThreatWatch + MS Teams = Actionable Intelligence

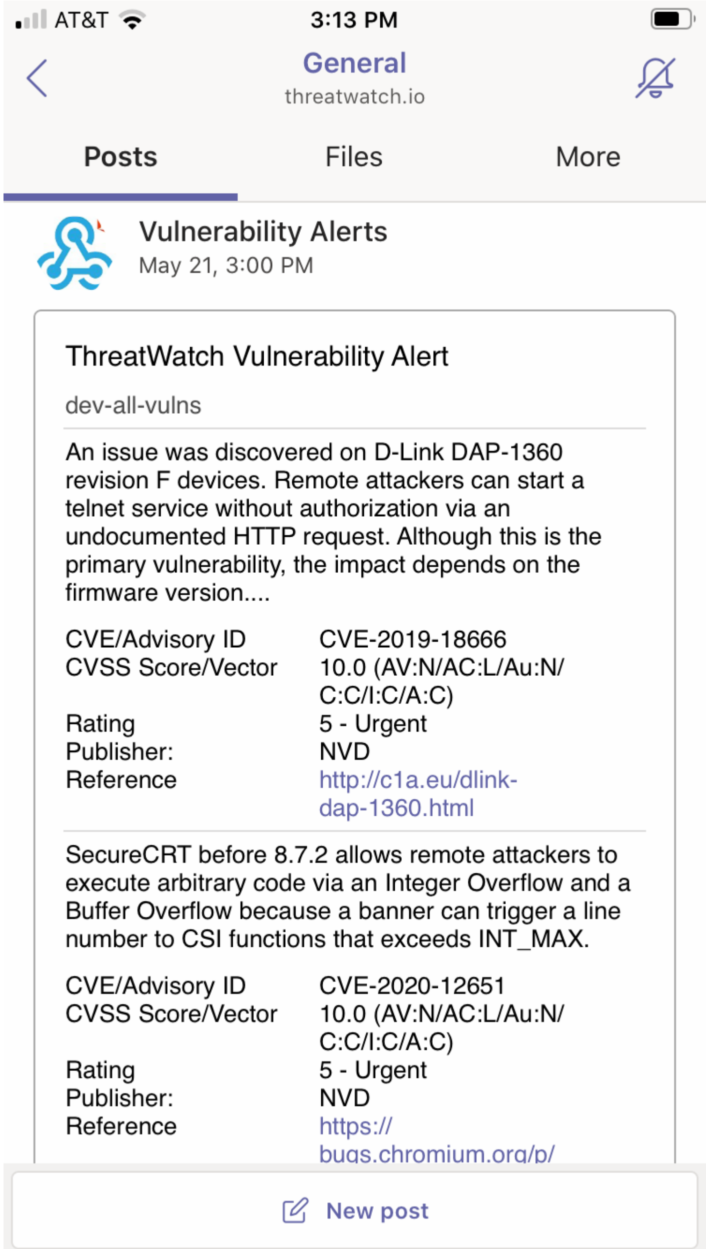
ThreatWatch will continuously monitor for vulnerabilities and the ThreatWatch portal shows the details and trends and can notify users via email alerts as well for real time alerting. However, there is a much better way to channel information within your organization if you are already using MS Team.

Using ThreatWatch portal you can set up very granular alerts either for vulnerability intelligence ( vulnerability alerts ) or for impacts that those vulnerabilities have on your assets. This is a great way to channel information to not just relevant teams and individuals but use existing tooling and mode of communication to share actionable information across the organization for timely action.

ThreatWatch alerts can be very granular and some examples of the supported filters are,

1. Publisher / vendor or product
2. Severity or availability of a patch or exploit for a given technology stack.
3. Impact detection of a vulnerability on a set of assets.
4. Priority impacts ( high severity or more likely to be exploited ) etc

A combination of a such filters can be used to create alerts and assign a MS Teams web hooks that delivers this information to precisely the right set of individuals.



Vulnerability Alerts
May 21, 3:00 PM

ThreatWatch Vulnerability Alert

dev-all-vulns

An issue was discovered on D-Link DAP-1360 revision F devices. Remote attackers can start a telnet service without authorization via an undocumented HTTP request. Although this is the primary vulnerability, the impact depends on the firmware version....

| | |
|---|---|
| CVE/Advisory ID | CVE-2019-18666 |
| CVSS Score/Vector | 10.0 (AV:N/AC:L/Au:N/C:C/I:C/A:C) |
| Rating | 5 - Urgent |
| Publisher: | NVD |
| Reference | http://c1a.eu/dlink-dap-1360.html |

SecureCRT before 8.7.2 allows remote attackers to execute arbitrary code via an Integer Overflow and a Buffer Overflow because a banner can trigger a line number to CSI functions that exceeds INT_MAX.

| | |
|---|---|
| CVE/Advisory ID | CVE-2020-12651 |
| CVSS Score/Vector | 10.0 (AV:N/AC:L/Au:N/C:C/I:C/A:C) |
| Rating | 5 - Urgent |
| Publisher: | NVD |
| Reference | https://bugs.chromium.org/p/ |