## MySecurityKarma Assessment Framework

## Introduction

MySecurityKarma is a initiative to enable enterprises and small medium businesses to secure their networks. Currently these are opensource and difficult to use or the commercial versions are costly. In addition the frameworks only talk of the issues but do not provide the solutions and then they have to rely on others to secure the same. The other issue is that people do not know what to test and why. So this site is supposed to give you peace of mind. So the team has already developed assessments and remediations for various operating systems.

Project Scope

The current scope of the project is to develop a pluggable framework in python to be able to plugin various benchmarks for various devices and then provide means to audit also? remediate the same. We start with certain cloud based benchmarks and the we can move that over to other tools.

The feeds are shared in excel format in the subfolder Assesment Feeds. We will start with the two starting with CIS\_Microsoft\*. These are Office365 and Azure benchmarks.

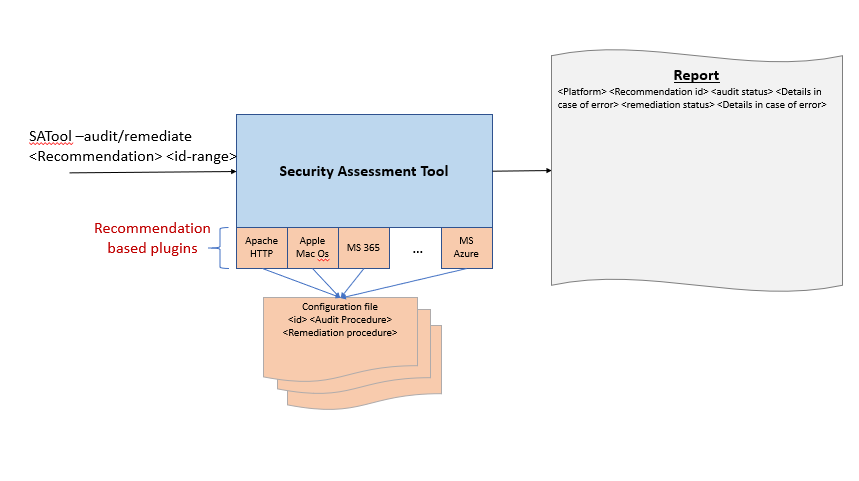
Inside these files the two important columns are audit procedure and remediation procedure. The fundamental requirement is that the moment audit fails the user may choose to auto remediate or manually remediate. So the python code should allow me to pass a Benchmark name (like Microsoft Azure?) and Recommendation id and then remediate the same.

Yes. This name will be registered in our DB

So in effect

1. A framework to pass Recommendation name, level and get an audit of how we are against the benchmark
2. A framework to pass Recommendation Name, id and ask it to remediate
3. All credentials for devices are stored in a vault(Hashicorp). Assume that a caller will retrieve the same and provide it.

## Implementation

****

* SATool will have 2 modes of invocation
  + CLI: For independent testing
  + Module: For loading into another program like the one we have for doing OS remediations to be used in live scenario.
* SATool can be run with options -audit, -remediate and Level (1,2) with Platforms(Apache HTTP, Apple Mac OS, MS 365, MS Azure etc) and a list of recommendation ids. SATool will invoke the plugin corresponding to the recommendation, get the result and add to the report.
* To add support for a new platform, following plugins will have to developed and deployed
  + The configuration file <Platform>.conf will contain following information

<id> <interface> <audit command> <condition for success> <remediation command><condition for success>

* Class implementing saplatform
  + The procedures will require the tool to run various interfaces like CLI or Azure Power Shell or Graph API. The implementation of these interfaces can either be at the central tool or with the plugins dependending on how generic or specific this interface is.
  + Format of the Procedure definition
    - Interface to be used: CLI, Azure PowerShell, Graph API
    - Command to be run
    - Input parameters that are required: {subscripton:subscription,accessToken:accessToken}
    - Condition for success
* Interface Implementation

| **Interface type** | **Implementation** |
| --- | --- |
| CLI | **Subprocess** module  Running the command with input parameters can be common.  Verifying the output will be specific to each command/id. |
| Graph API | TODO |
| Azure Power Shell command | TODO |

**Suggestions for Modules** (TODO):

* **pandas** : to read from/write to xls
* **pywinrm**: Python client for the Windows Remote Management (WinRM) service
* **python requests**: HTTP library for Python
* If the requested Platform, Recommendation id, procedures are not available appropriate errors will be displayed
* **Example1**:
* **Input**: SATool -audit Topic=Microsoft 365, Level= 1/2
* **Output**:
* Total recommendations: XX, Audit Success = XX, Audit failed=XX, No audit procedure available = XX, Manual audit procedure = XX

| **Recommendation**  # | **Result**  Compliant/ Not Compliant/ Cannot Check | **Details** |
| --- | --- | --- |
|  | Cannot Check | No audit procedure available |
|  | Cannot Check | Manual procedure |

**Example2**

Input: tool remediate Recommendation Name=Microsoft 365?, id= 1.1. 1.2

Output: excel file with the following information

Total recommendations: XX, Successfully remediated = XX, Remediation failed=XX, Remediation

| **Recommendation**  # | **Level** | **Result**  Success, Failed, No action | **Details** |
| --- | --- | --- | --- |
| **1.1** | **1** | Success |  |
| **1.2** | **2** | Failed | Error: “xxxxxxxx” |
| 1.3 | 1 | No action | No audit procedure available |
| 1.4 | 2 | No action | Manual procedure |

https://gitlab.com/SreedeviGattu/research/tree/master/SAT

Testing

Test Environment

Need help in getting around in the system - Show

how to check one Microsoft Azure audit procedure (CLI)

How to check one Microsoft 365 audit procedure (Power shell)

| **Office 365 URL - non-delegated**  <https://www.office.com/>  Is this a VM? Loging didn’t work | username: madanica\_a@eagleconstructionco.onmicrosoft.co  pwd: Dyno@2016 |
| --- | --- |
| **Office 365 URL - Delegated:** <https://partner.microsoft.com/en-us/cloud-solution-provider/csp-partner>  Is this a test system? | username: admin@csptip72s.onmicrosoft.com  pwd: T8%F4=eD |

Test cases

* Invoke it as a CLI
* Load it as a module
* Pluggability/Extensibility:
  + Add a new platform, interface
  + For a existing platform, add a new recommendation id, audit procedure, remediation procedure
* Inputs:
  + 1 platform, ids with only audit, remediation and both
  + 3 platforms, ids with only audit, remediation and both

- Platform, Recommendation id, Interface exists, Audit/Remediation - Cmd Successful, Condition for sucess exists - success, fails

- Platform, Recommendation id, Interface exists, Audit/Remediation - Cmd Fails, hangs

- Platform does not exist (not implemented)

- Platform exists, Recommendation id does not exist (not implemented)

- Platform, Recommendation id exists, Interface does not exist (not implemented)

- Platform, Recommendation id exists, Audit/&Remediate Cmd does not exist (not implemented)

- Platform, Recommendation id, Interface exists, Audit/Remediation - Successful, Condition for sucess does not exist (not implemented)

Clarifications

In CIS\_Microsoft\_365\_Foundations\_Benchmark\_v1.0.0\_Certification (1).xls

| **Level1 Recommendations** |  | **Level 2 Recommendations** |  |
| --- | --- | --- | --- |
| Type of audit procedure | Total | Type of audit procedure | Total |
| ManualProc | 51% | ManualProc | 24% |
| No audit proc | 14% | No audit proc | 28% |
| PowerShell | 33% | PowerShell | 40% |
| REST API | 2% | REST API | 8% |
| Graph API |  |  |  |

1. What should the handling be if there is no audit procedure or if there is only a manual procedure i.e. there is no command based interface? Just leave a comment against the recommendation id?

If there is no audit procedure we will take a manual input against it to ask the user to tell us if this is done or not.

In case of the Manual Procedure we are talking of checking the Web based UI for the settings right. We can automate that using the Web scraping tool like go-colly. If we cannot then we can ask user to Verify

1. Only a small part (35% or 48%) seems to be automatable. Is that a good value-add?

That is a significant . If we add the web scraping we may be able to push this to 65-70%