Test Application of Cloud Computing Risk Model

William Woodson Anastasia Walter Margaret Kareithi
The University of Texas at San Antonio

Abstract

The purpose of this project was to evaluate the capabilities and functionality of the “Cloud Computing Risk Analysis Model” [1] previously developed within the INSuRE program by developing and executing a testing framework during risk assessments of a sample IaaS cloud service provider (CSP). The research used the results of testing a commercial public cloud, Amazon Web Services (AWS), to validate the risk model and demonstrate the viability of testing identified controls from an end-user perspective. A summary of control tests, procedures, and results by control family for a “reasonable” configuration within AWS are included in this poster. Detailed results are published though the Purdue University Research Repository (PURR)[1]. Future assessments performed via the cloud computing risk analysis model and “cloud control assessment framework”[2] tool may provide for more rigorous validation and confidence scoring/ weighting as the model moves closer to a production-ready state.

[1] https://purr.purdue.edu/projects/insureclass/files/

[2] https://github.com/wjwoodson/cloud-control-assessment-framework

Introduction

The Federal Risk and Authorization Management Program (FedRAMP) was developed to provide a central assessment and certification system for cloud service provider (CSP) assurance and appropriate use by federal agencies [2]. Understanding and being able to adequately evaluate and quantitatively measure risks associated with public clouds has become an imperative for the FedRAMP program.

Specifically, there is a need for organizations to have the capability to evaluate and assess risks of using the public cloud without external assessors or privileged access to the CSP. In order to achieve this goal, the previously developed cloud computing risk assessment model must be tested to ensure its validity from an end user perspective in order to serve as a reusable tool for assessing risk of public clouds in the future.

This research is based on evaluation of and scoping of controls identified by the **FedRAMP moderate security baseline: 325 security controls (2329 control objectives)**. Previous work identified **175 unique control aspects of the FedRAMP Security Controls to be testable by end users.**

This task is associated with gathering, analyzing, and retaining a great amount of data and understanding of different procedures to be taken during the testing phase. To successfully accomplish this task, the research team reviewed all associated resources and proposed testing procedures, developed technical control tests specific to the AWS platform, and executed control tests with verbose documentation of all steps and procedures taken.

In addition to the assessment, we have began work on a testing framework and application to accomplish this task and provide for repeatable, flexible assessments with standardized scoring and reporting.

Goals

Utilize “Cloud Computing Risk Model” identified end-user controls derived from NIST SP 800-53A [3] to assess a CSP.

Develop reusable methods and technical control tests for assessing the candidate CSP.

Automate testing and reporting where possible.

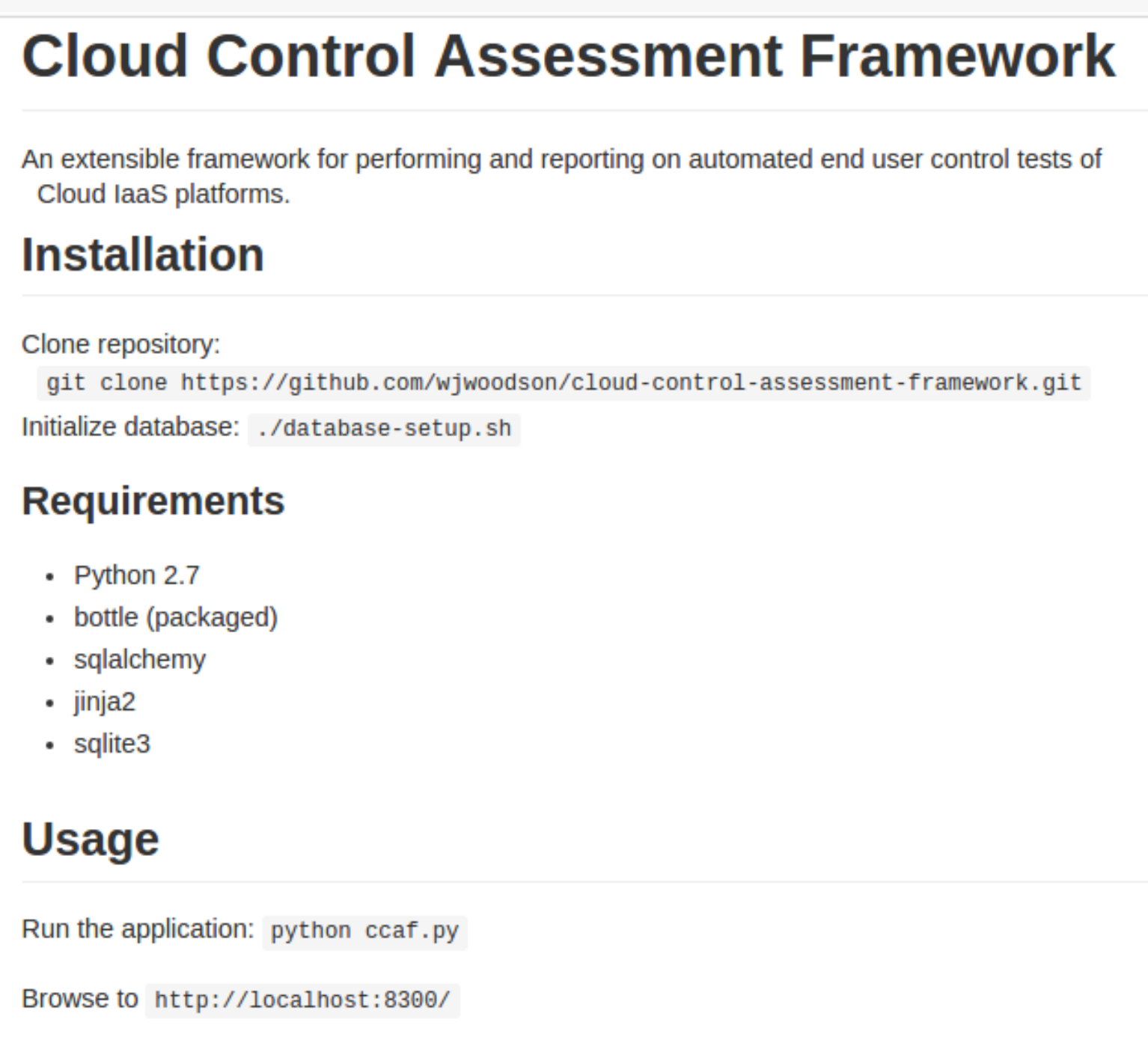
Methods

Utilized empirical research, which tests the feasibility of a solution using empirical evidence. Our research plan has been to systematically assess the cloud security controls determined by previous research and develop/implement technical controls tests for the specific Amazon Web Services platform.

175 control tests assessed if 'technical, end-user control'

86 control tests reviewed for applicability to AWS

75 control tests developed & executed for AWS assessment



Cloud Control Assessment Framework

An extensible framework for performing and reporting on automated end user control tests of Cloud IaaS platforms.

Installation

Clone repository:

```
git clone https://github.com/wjwoodson/cloud-control-assessment-framework.git
```

Initialize database: `./database-setup.sh`

Requirements

- Python 2.7
- bottle (packaged)
- sqlalchemy
- jinja2
- sqlite3

Usage

Run the application: `python ccaf.py`

Browse to `http://localhost:8300/`

Figure 1.

Cloud Control Assessment Framework (ccaf.py) – Python web application (API & designed GUI) for assessment and storage of control test results.

Utilizes standard or customized control test modules loaded at run time to manage storage of manual testing steps, documentation, results and/or control test automation (Interaction with CSP API, etc.)

Results

As assessed by our team to a ‘reasonable level’ of configuration by the AWS Administrator (account root) and authorized security administrators, the AWS platform was 83% successful in application of the Cloud Computing Risk Model - User Validated controls (75).

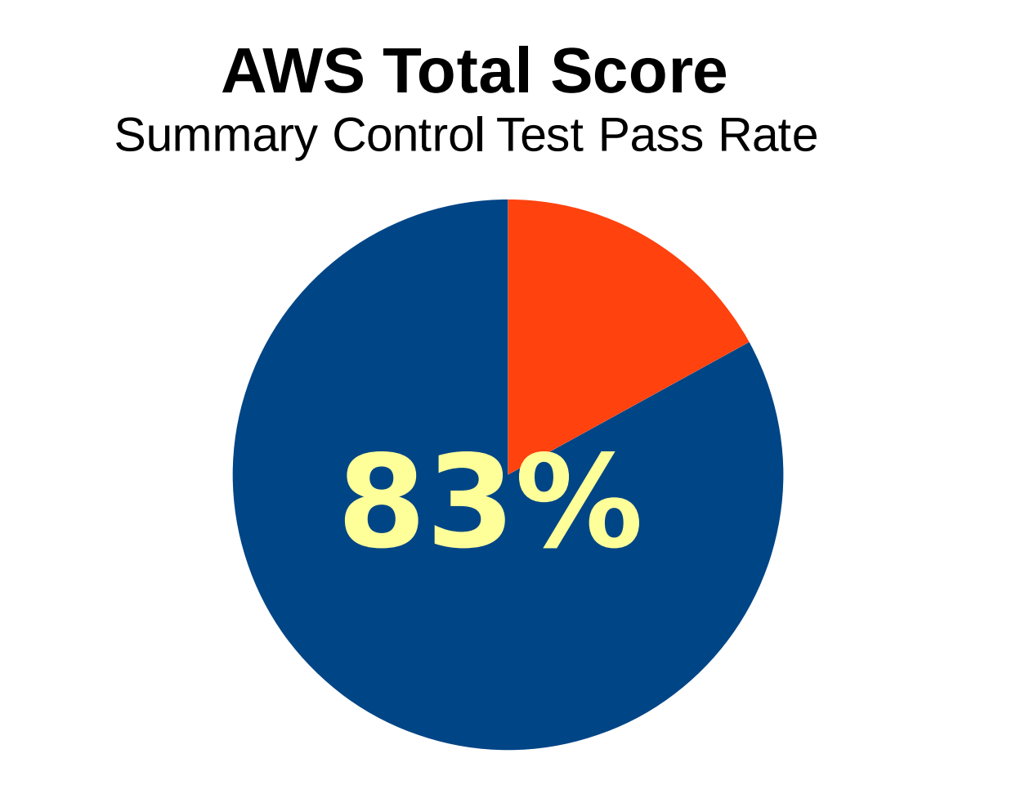


Figure 2.

A summary point score of 62.25 was calculated for the 75 controls tested when AWS is configured to a 'reasonable level'. This represents a 83% overall pass rate.

Control scores for this assessment were equally weighted on the 'control aspect' level (NIST SP 800-53A rev4 Control ID).

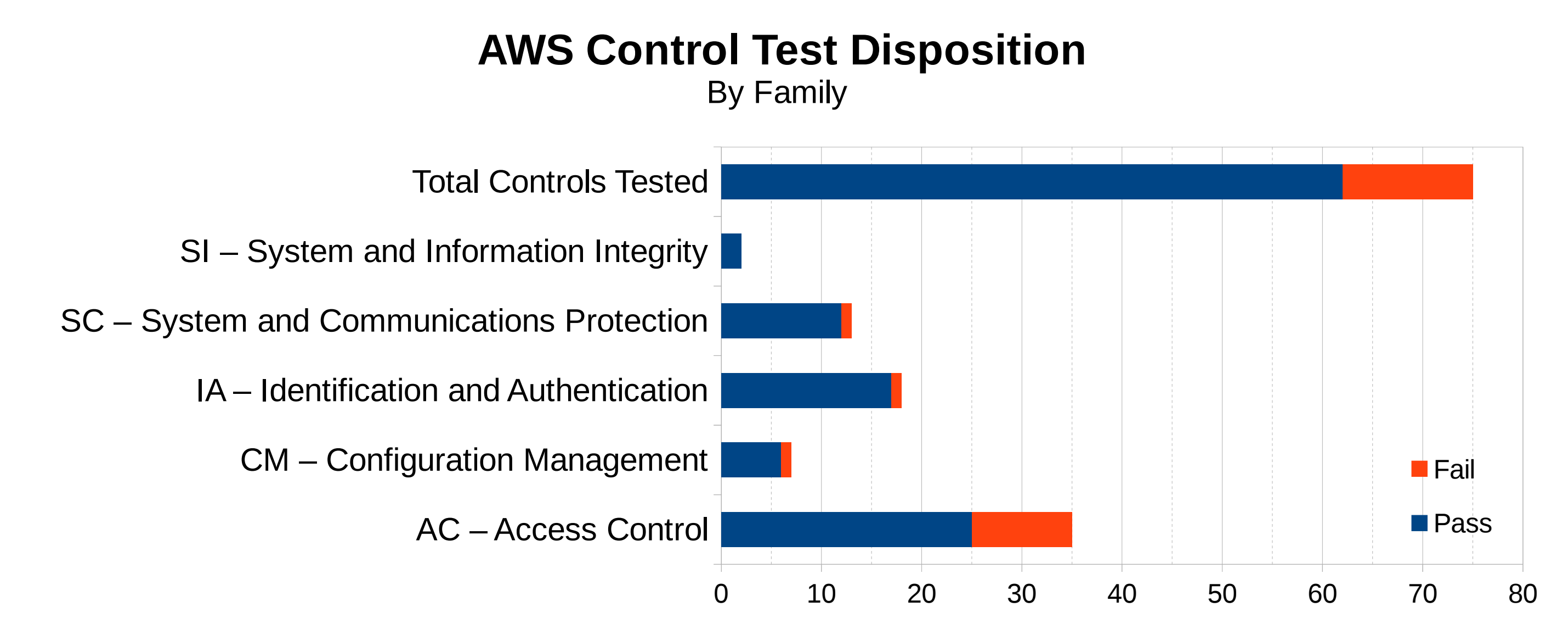


Figure 3.

Results of AWS control testing – test disposition (pass/fail) by control family (62/13)

Results - con’t

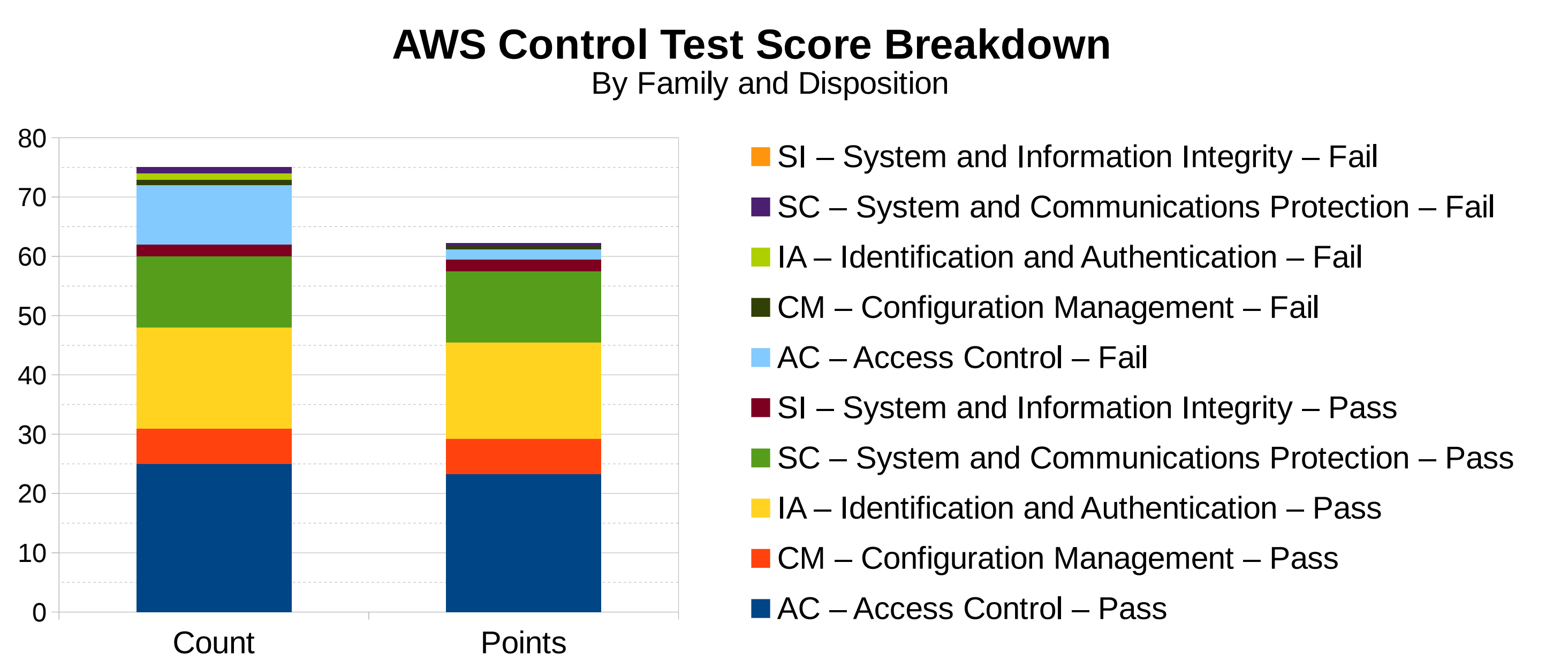


Figure 4.

Final count and point score per control test family and control test disposition. Control tests were considered 'passing' if a weighted sum of sub-test tasks >=75%.

Selected Results: Role-Based Schemas – **Detection/Prevention of Password Guessing Attack** – AC-2(7)(b)

Summary: Captcha implemented on AWS Admin (root) attack, however no alerting or lockout for failed password attempts for IAM users (even when provisioned Admin role)

Recommendations: AWS should permit account lockout policy on failed authentication attempts via the IAM global password policy

Selected Results: Password-Based Authentication – **Minimum Password Age -** IA-5(1)(d)[3]

Summary: No way to enforce minimum password age through AWS IAM. Mitigating` factor – additional logging configurable in (CloudTrail) to monitor users for a non-compliant password change given additional correlation and alerting tools.

Recommendations: AWS IAM should allow user password policy to be configured by the Administrator to include requirements for minimum password age.

Conclusions

- If configured to a 'reasonable level' AWS satisfies many of the security concerns addressed by cloud computing risk model (end-user assessed)
- AWS platform passed end-user tests with 83% score
- Manual testing validated publicly available AWS documentation is accurate.

References

[1] Auger, G and Hilgers, R. (2015). “Public Cloud Providers Security Implementation; End-User Validated.” Purdue University Research Repository.
[2] GSA (2015). About FedRAMP. Retrieved from Gsa.gov.
[3] Ross, R. et al. (2014). "NIST SP 800-53A, Revision 4." Assessing Security and Privacy Controls in Federal Information Systems and Organizations.

Acknowledgements

We would like to thank **Dr. Glenn Dietrich, Ph.D** and **Mr. Andy Sampson** for their guidance in this research.

This material is based upon work supported by the National Science Foundation under Grant Nos. 1344369 and 1027493, and the National Security Agency under Grant No. NSA-H98230-15-1-0300. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the National Security Agency.