Answering the Unanswerable Question: How Secure Are We?

Jason Bohreer

Answering the Unanswerable Question: How Secure Are We?

GIAC (GCCC) Gold Certification

Author: Jason Bohreer, jason@bohreer.com
Advisor: Jonathan Risto

Accepted: 2020-May-08

Abstract

Business environments consist of invisible or ill-defined risk factors which create challenges with prioritization for business owners, systems owners, and IT/Security teams in their goal to improve their security position. The security of the environment relies upon the appropriate people understanding and addressing the risks. However, they typically do not have the relevant understanding, and therefore, the capability to act, due to the complexities of the defense-in-depth strategies.

Security professionals have a good understanding of the relationships between the various controls and have numerous tools to consolidate logs and network traffic. However, while many of these tools are "best-of-breed" and operate within their information silos, they lack native methods to populate external systems to aggregate the findings in a risk-based approach which business stakeholders require to make decisions.

By designing a framework to collect and measure different aspects of security, this research explores how to remove the operational fog that obscures our vision of our environments. With layers of fog removed, the improved clarity allows us to make quantitative assessments of our security by examining how security controls relate to one another.

1. Introduction

The principle which states that eventually things fail is the foundation of information security. Eventually, the industry reports vulnerabilities that, even within hardened systems, render the configuration moot. Configuration changes are missed, forgotten, or overlooked (Ravenel, 2006). Frequent, and legitimate, security exceptions are granted to further a business need, and eventually, the security of the system is not as robust as it once was. As an industry, security professionals have worked around this issue by adhering to the defense-in-depth principle (Limoncelli, Hogan, & Chalup, 2007) to utilize multiple layers of security to protect our critical systems. While this principle provides protections, it unfortunately, makes assessing the effective state of security for a system more complicated.

To understand the state of security, a data security professional must be proficiently knowledgeable in the functionality and operation of each layer and then be able to quantify the performance of each one. Add to this complexity of systems the additional complexity that individuals bring into the environment (e.g., requirement for separation of duties, existing skill gaps), and an understanding of the number of different layers of security that may exist in an environment becomes a daunting, if not impossible, task.

Security professionals face several challenges when trying to quantify the security of an environment. The first is not having a comprehensive list of the security components within the environment, and more importantly, how those components relate to each other. We accept the principle of defense-in-depth and utilize multiple layers to protect our systems but fail to assign scores to those controls. Second, we often lack the required access to assess those systems by enforcing least privilege and separation of duties. The individuals responsible for collecting the information are often not the same as the people managing the system (National Institute of Standards and Technology, 2013). Finally, we have the false expectation that to measure or quantify security means that we have an exact value, a perfect picture, rather than using those measures to illuminate or remove uncertainty about our environment (Hubbard, Seiersen, Geer, & McClure, 2016).

1.1. Current/Historical Approach

The standard approach used to understand and score security is through a risk assessment process (National Institute of Standards and Technology, 2020). The process typically involves identifying the various threats and threat actors that may take advantage of a vulnerability, deficiency, or risk, and then providing a qualitative assessment of the impact and likelihood based on some pre-defined scoring matrix. While these assessment processes are essential to understanding the overall risks to an organization, they are typically focused on business processes, or on the organization as a whole, and occur based on regulatory requirements (e.g., HITRUST; PCI). They do not provide clear tactical direction to the Information Technology and Security teams on actions to improve their understanding of the state of security in the environment (Chew, et al., 2008).

Another method the industry has used to score security is through the Critical Security Controls. This framework (Center for Internet Security, 2020) recommends the best steps for organizations to implement security controls by addressing the timing, people, and resources needed to secure the environment. Unlike the NIST or ISO frameworks, the Critical Security Controls use recommendations from industry experts and the changing security landscape to define the implementation order that organizations should follow. However, while the Critical Security Controls framework also provides a metrics plan for each of the layers, the design focuses on measuring adherence to each layer as a whole through a six-sigma process maturity model. While beneficial from a security leadership position to understanding how the security program is functioning within the organization, it does not provide tactical guidance or improve clarity on the current state of security.

1.2. Future State

What the industry lacks is a method to assess and score individually implemented security aspects and then present the results to the Information Technology and Security teams in a way that allows them to prioritize work and conduct additional investigations. A method to relate the individual security aspects to the whole is needed, to better understand how they impact the overall state of the environment. While not attempting

to develop a "perfect" solution, this solution must be one that works to remove uncertainty about the security of the environment. Therefore, this research will establish a framework to collect security aspects, establish a standard for measuring those aspects, and then presenting the information such that the business understands the state of security.

2. Methodology

2.1. Develop Collection Framework

Before determining what type of security aspect information would be collected, a collection and storage framework was developed. The expectation was that there would be a large number of data sets from different sources, each with their unique data definitions. It was important to develop a standard method for defining, collecting, importing, and then measuring the data, to simplify the process and limit the amount of duplicate work (e.g., creating output for each of the different output file types, importing data, creating database tables). The principle for the framework was that each stage should operate independently of the core processes within but should have a relationship with the adjacent stages within a single pipeline (as shown in Figure 1). By utilizing multiple scripts to perform collection, the pipeline can be extended to handle the needs of different organizations.

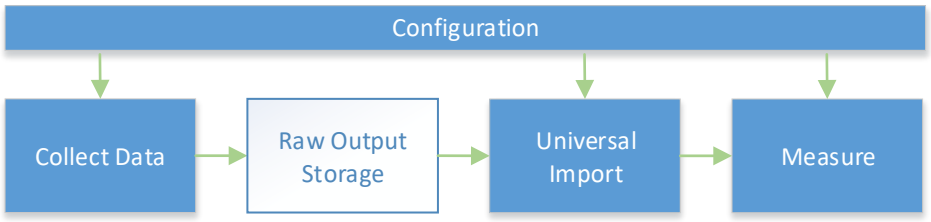

Figure 1: Collection Framework Pipeline

The framework handles 1) the routine tasks of defining the common populations that would be required, 2) the details required for the collection of security aspects, and 3) the methods for measuring each security aspect. At the heart of the definition framework are three components: configuration information, a universal import process, and measurement definitions.

2.1.1. Configuration Information

A core configuration JavaScript Object Notation (JSON) contains the information that each of the collection scripts needs to complete the task, including the core raw output storage, databases, and queries to execute. The collection scripts, along with support processes, returns a standard object that defines all of the destination information for the raw output. The configuration JSON also provides details for the universal import process on where to collect the raw data and where to store the results within a Microsoft SQL database system.

2.1.2. Universal Import

In place of an import process for each population or measurement, a universal import process was built that works against the contents of the data files. The import process determines which configuration to use based on the parent folder structure of the file. The import process then reads the contents of the XML file and dynamically builds a series of SQL insert statements based on each of the column data types, and then inserts the records into the appropriate table. The original data files were moved to the *Processed* folder once the import process completed.

2.1.3. Measurement

The variability in each organization's security program dictated that the measurement process would require the most flexibility to address various use cases, including measuring security aspects using different calculations. As the data resides in a database environment, the calculations take advantage of the improved processing of the SQL engine. Support for multiple versions of each measurement was incorporated to permit the addition of new formulas without impact to any existing reports, permitting for risk-free experimentation.

2.2. Collect Common Populations

Several core populations were required to evaluate the security of an environment. These populations provide the starting point for determining which systems to evaluate, individuals that are in scope, and ultimately how the relationships between individuals and systems are to be defined.

2.2.1. Systems

As defined in Critical Security Control #1 (Center for Internet Security, 2020), to protect an environment it is critical to have a complete list of systems. The system-specific security aspect collection processes utilize this list from which to collect data. While *hostname* is the only required property, additional properties could be collected to allow for specific reports to roll up security measurements based on where a system is located, or system ownership.

2.2.2. Active Directory Account

Accounts provide the link between the resources available on systems and the individual responsible for the account. The expectation was that a person might have multiple accounts due to common security standards (National Institute of Standards and Technology, 2013). A typical example of this includes Information Technology professionals using dedicated accounts for privileged functions. Two properties are critical to collect to establish relationships between other data sets. The first property, the *samAccountName* property (which is always present with Active Directory user accounts), provides the unique primary key in the table and provides the link to data stored in the Group Memberships and Local Assignments on Local Systems. The second property, *emailAddress*, provides the foreign key used to establish a relationship between the Active Directory account and the person tied to the account. In the cases where the Active Directory account does not have an associated *emailAddress* property (e.g., service accounts), the impact of people measurements was not evaluated.

2.2.3. Group Memberships

All Active Directory groups and associated memberships were collected. These memberships, coupled with the local assignments, provide the foundation of understanding which users have access to individual systems. Additionally, the local groups of each system were also evaluated and collected. Defining the individual group memberships involved unraveling any nested local and global groups to provide a list of unique accounts with access to the individual system.

2.2.4. People

Unlike accounts, which are artifacts of an authentication system, people represent the individuals who utilize those accounts. Because the goal is to provide clarity on the overall security stance, having visibility into the systems *and* the people who use those systems was necessary. This understanding requires information on the 1-to-many relationship between a person and their assigned accounts.

While not every Human Resource department allows direct access to query their employment data, they are typically required to provide this information as part of any security audit (e.g., SOC2, Type 2). With this information, we can map each person, through their accounts and group memberships, directly to each system within the environment as illustrated in Figure 2.

Figure 2: System to Person Mapping

2.3. Collect Security Aspects

Security Aspects are properties associated with either the systems or people in the environment. By themselves, they do not represent measurements, but rather data points. Examples include a list of running processes, installed services, open ports available from the network, or a list of certifications or licenses a person has achieved. The project focused on a collection of security aspects loosely based on the Critical Security Controls (CSC). The Center for Internet Security (CIS) developed the Critical Security Controls (Center for Internet Security, 2020) principally as a "set of actions for cyber defense that provides specific and actionable ways to stop the most pervasive and dangerous attacks" (SANS Institute, 2020). The collection process focused on four CSC families to measure the security aspects of systems and one CSC family to measure the people that utilize those systems.

2.3.1. Inventory and Control of Software Assets (CSC2)

The core of the *Inventory and Control of Software Assets* control centers around two processes for software management. The first, a software inventory collection process (CSC 2.1), describes what applications are used throughout the environment,

while the second, an enforcement process (CSC 2.7), defines which software is allowed to run through the use of application allow lists. After connecting to each system remotely, PowerShell examined the AppLocker configuration utilizing built-in cmdlets before storing the results.

2.3.2. Continuous Vulnerability Management (CSC3)

Continuous Vulnerability Management (CSC3) states that enterprises should run vulnerability management software to assess the current exposure level for systems within their environment. Information Technology and Security teams use those scan reports to understand the scope and depth of their exposures, develop remediation plans, and prioritize patching and reconfiguration work.

Vulnerabilities are an inevitable part of the world of security, and one of the recent and most widely recognized data breaches occurred due to an organization failing to patch their systems promptly (Equifax, 2017). Measuring this information provides insight into the exposure, but there is no clear standard on how to address these vulnerabilities. Should one address vulnerabilities with a high Common Vulnerability Scoring System (CVSS) score, or by the vendor's severity level? How does an organization deal with lower vulnerability scores that have been unaddressed for long periods?

The process collected two sets of data from the Qualys vulnerability scanner. The first set of data captured specific information about the vulnerability, including the title, the Qualys' unique ID, the severity, date released, and the CVSS values. The second set of data captured every instance of the open vulnerabilities, which included the hostname on which the vulnerability was detected, the date of detection, and the number of times detected.

2.3.3. Controlled Use of Administrative Privileges (CSC4)

Controlled Use of Administrative Privileges (CSC4) focuses on how to manage high-level accounts within the environment. Privileged accounts provide unrestricted access to system configurations, often allowing the owners to modify core security settings. As a result, they are often a key step an attacker uses when trying to achieve their ultimate objective. Without a regular process for review, or tight change

management processes, accounts may retain access longer than is required. Alternatively, groups may be incorrectly assigned privileges, resulting in an unexpected influx of accounts with elevated rights.

For each system in the environment, the process collected each local Windows group, along with their respective memberships. Afterward, a SQL query processed the data set to enumerate each of the next groups to create a list of accounts that are members of the specific local Windows group.

2.3.4. Security Configuration for Hardware/Software on Servers (CSC5)

The final system-specific security aspect that will be collected is *Secure Configuration for Hardware and Software on Mobile Devices, Laptops, Workstations and Servers* (CSC5). Organizations such as the Center for Internet Security with their Secure Benchmarks (Center for Internet Security, 2020) and the National Institute of Standards and Technology's Security Technology Implementation Guide (STIG) (National Institute of Standards and Technology, 2019) provide system hardening standards. The enforcement of these standards works to configure the least amount of services and functionality for an attacker to exploit, while concurrently ensuring that proper audit controls are enabled (Center for Internet Security, 2020). The process collected the policy configuration results from the Qualys policy compliance scanner based on the *CIS Benchmark for Windows Server 2012/2016* (Center for Internet Security, 2020). This data included the hostname assessed, the policy statement, and policy statement test results.

2.3.5. Implement a Security Awareness and Training Program (CSC17)

Ultimately, because people must interact with the systems in the environment, and some people exhibit more risky behaviors than others, the *Implement a Security Awareness and Training Program (CSC17)* control was evaluated. To address this, companies roll out security awareness training programs to educate people on their responsibility and the company's expectations when it comes to security. The process collected the two sets of data out of the security awareness management system: security awareness training and phishing campaign results.

The security awareness training results captured a person's adherence to completing the required training material. The phishing campaign results captured a person's ability to detect and appropriately respond to suspicious emails. While only two data sets were examined, combined, these results provide an insight into an individual's maturity with a security-focused mindset.

2.4. Develop Aspect Measurement Process

Several questions arose when designing the measurements of security aspects. The most pressing challenge involved how to deal with inconsistent results or empty values as part of the collection process. As the purpose of the process is to provide clarity by removing uncertainty, any security aspects that could not be collected or tested were assigned a score of -1 during the measurement process. This value represents the failure to remove uncertainty from the security aspect and provides easy identification of failed collections. For each security aspect that could be measured, they were assigned points based on a 100-point scale. Partially compliant items were either pro-rated based on a fixed percentage or had specific points assigned based on business rules.

Conducting the measurement process as a point-in-time assessment, as opposed to building calculations that dynamically calculate the results during the reporting phase, allowed the individual responsible for designing measurements the ability to design and run different measurement profiles without losing access to historical versions of the measurements. Measuring the security of the collected security aspects involved two different measurement levels: System and Person. These levels allowed for a detailed review of individual systems or people. Each measurement result stored is as provided in Table 1: Table Definition for Measurement Results.

Field Name	Type	Purpose
dateAdded	Date/Time	The date/time of the record storage
dateAssessed	Date/Time	The date/time of the assessment of the measurement
Include	Int	Used to exclude specific records from the reporting system, if needed
measurementID	String	A unique value that identifies the type of measurement

Field Name	Type	Purpose
measurementVersion	String	The version identifier for the measurementID. Allows for the running of different scoring methods of the same measurementID
measurementSource	String	The name of the system, or person, associated with the measurement
measurementValue	String	The result, or score, of the measurement.

Table 1: Table Definition for Measurement Results

The choice to use String values for *measurementValue* was made to allow the greatest flexibility in scoring the results. Allowing non-number values (e.g., A-F Grade score) was not considered a hindrance when coupled with PowerBI's ability to easily convert data types.

2.5. Conduct Aspect Measurement Process

2.5.1. AppLocker

The collection of AppLocker information included five different allow lists, along with their enforcement mode. However, since the environment only enforced limitations on executables (i.e., EXE), it alone was used for scoring this measurement (as defined in Table 2: AppLocker Scoring).

Enforcement Mode	Points	Rationale
Enabled	100	The application "allows rules" are enforced, and the system denies any executable that does not match the defined lists.
Audit Only	75	This level of enforcement logs all executable launches, including whether the executable would have launched if enforcement was enabled.
Not Configured	25	While not configured for auditing or enforcement, the system is accessible to the measurement system to collected information.
Unknown	-1	The measurement process was unable to find information on the system. Unable to view the status of the system represents the highest risk to the environment, and thus the lowest score. Commonly this was caused by a lack of permissions to query the AppLocker information remotely.

Table 2: AppLocker Scoring

2.5.2. System Vulnerability

Assigning an overall system vulnerability score to an individual system utilized two core methods based on the associated CVSS values. The first method took the maximum CVSS value for an individual system (0 – 10 scale) and multiplied it by 10 to normalize the results to a 100-point scale. Then, this value was subtracted from 100 points as defined in Equation 1: Maximum CVSS Vulnerability Scoring. The purpose of this calculation is to measure systems based on the highest-rated CVSS that is impacting the system.

$$100\ points - (10 * max(System_{vulnerabilty_{cvss}}))$$

Equation 1: Maximum CVSS Vulnerability Scoring

Measuring the system against the highest vulnerability allows the team to see systems with the riskiest vulnerabilities. Unfortunately, by focusing on just one vulnerability the team would be blinded to other "slightly less" risky vulnerabilities. For example, a system with one 9.0 vulnerability and two 8.0 vulnerabilities will be scored at the 9.0 value and will receive 10 points.

The second method took the average CVSS value for an individual system (0 – 10 scale) and multiplied it by 10 to normalize the results to a 100-point scale. Then, this value was subtracted from 100 points as illustrated in Equation 2: Average CVSS Vulnerability Scoring. The purpose of this calculation is to provide another system vulnerability perspective by highlighting systems that have multiple lower-rated CVSS that might be overshadowed by simply looking at the maxCVSS value.

$$100\ points - (10 * average(System_{vulnerabilty_{cvss}}))$$

Equation 2: Average CVSS Vulnerability Scoring

Measuring the system against the average vulnerability allows the team to see an overall picture without focusing on the worst vulnerability. Unfortunately, this approach can skew the results of systems with a large number of lower-ranked vulnerabilities. For example, a system with one 10.0, one 9.0, and one 3.0 vulnerability will score at the 7.3

value and receive 27 points. A system with three 7.0 vulnerabilities will score the 7.0 value and receive 30 points.

Each of the above scoring methods have benefits and costs. Neither approach represents a perfect understanding of the risk associated with a system. However, by including both the forest (average vulnerability score) and the tree (maximum vulnerability score) equally, an improved understanding of the environment can be achieved.

2.5.3. Local Administrator

Each system in an Active Directory environment has several accounts listed within the local Administrators group for Information Technology administration, configuration management, and other automation tasks. The measurement process used a pro-rated range between an expected normal value and a maximum acceptable value. Without a maximum acceptable value, percentage average calculations would permit systems with extremely high numbers of users with local administrator rights to receive some points. Systems that had less than the normal number of accounts were assigned 100 points, while systems over the maximum acceptable range scored 1 point. Any system that failed an evaluation was assigned -1 points. Systems with a score between the normal and maximum amount were assigned pro-rated points, based on the Equation 3: Local Administrator Scoring formula.

$$100\ points + \left((@normalCount - count(samAccountName)\right) * \left(\left(\frac{100\ points}{@maxAcceptable - @normalCount}\right)\right)$$

Equation 3: Local Administrator Scoring

2.5.4. Policy Compliance

The policy compliance process assessed various configuration items and scored the results as passed, failed, or unknown. The overall calculation included any unknown status items, as this better represented the overall understanding of the system's compliance with the policy as defined in Equation 4: Policy Compliance Scoring. Unknown status items represent a failure in the process to assess the configuration item correctly and could artificially increase the measurement assigned to the system.

$$100\ points * \left(\frac{count(config_{passed})}{count(config_{passed}) + count(config_{failed}) + count(config_{unknown})}\right)$$

Equation 4: Policy Compliance Scoring

2.5.5. Phishing and Security Awareness

The measurement process for phishing needed to work within the constraints of the data that exists as a result of the assessment process. This, coupled with the fact that each phishing campaign may test different techniques, meant that an additive measurement process was required. The less the system was able to detect a person interacting with the phishing message, the more points the person earned, as described in Table 3: Phishing Results Scoring. The exception is that a person who reported the phishing email via the appropriate processes earned additional points.

Enforcement Mode	Points	Rationale
Clicked	0 or 10	0 = The system detected the user clicked on the message. Scored lower due to detection issues related to mobile devices. 10 = No user interaction was detected
Opened	0 or 10	0 = The system detected the user opened the email message. Scored lower due to detection issues related to mobile devices. 10 = No user interaction was detected
Replied	0 or 15	0 = The user replied to the phisher 15 = No user interaction was detected
Date Entered	0 or 15	0 = The user entered data into the website listed in the phishing email 15 = No user interaction was detected
Attachment Opened	0 or 15	0 = The user opened the attachment included in the phishing email 15 = No user interaction was detected
Macros Enabled	0 or 15	0 = The user enabled macros on the attachment included in the phishing email 15 = No user interaction was detected
Reported	0 or 15	0 = No user interaction was detected 15 = The user reported the phishing email via the Outlook plug-in.

Table 3: Phishing Results Scoring

Unlike the phishing measurement process, measuring security awareness focuses on the person's completion status of the assigned training modules. Individuals that complete their assigned training module by the due date receive the full 100 points. If

they complete the training after the due date, they receive 25 points. Assigning a severe reduction in points was the logical choice due to the numerous automated reminder emails that were sent out to complete the training, and a generous 30-day training window. Individual training modules that have not been started or are in progress at the time of the assessment are assigned 0 points. While assigning 0 points lowers the score during the initial measurement, subsequent measurements correct the calculations (assuming the person completes the training). This allows the measurement process to handle trainings that are assigned without required completion dates.

Training Status	Points	Rationale
Completed (before or equal to due date)	100	The person completed their assigned training within the defined time.
Completed (after due date)	25	Training not completed within the required time.
In Progress	0	If completed by the next assessment period, the person is assigned the appropriate points.
Not Started	0	

Table 4: Security Awareness Scoring

3. Reporting

While other reporting solutions, such as Tableau, could have been used, Microsoft PowerBI is freely available. It also provided an easy-to-use interface for defining the relationships, assigning missing security aspect measurements, and normalizing collected data. Reports were easy to craft and publish for the Information Technology and Security team to review.

3.1. Relationship Definitions

As the populations and aspect collections polled from different sources, the expectation was that different data sets would describe attributes differently. As an example, email information that was collected from Active Directory came from the *emailAddress* property, but the Human Resources people data feed returned the information in the *email* property. Also, different letter casing caused inconsistencies between data sets. As an example, email in one system used all lowercase, while another system used an initial letter casing (e.g., John.Smith@test.com). A hostname in one system was all uppercase, while lowercase in another. These different types of casings led to inconsistent results when working on drill-through reports. The transactional

nature of PowerBI allowed the data to be normalized based on attribute type (e.g., email addresses were set to lowercase).

The relationships between the various data sets were established once the attribute names and values were standardized. While it was possible to enforce referential integrity within the database environment, this assumes that all data collected at all times would perfectly align. In practice, this assumption of needing "perfect" data hampers the goal of understanding the entire environment as it is, not what we wish it would be. Instead, the relationships were defined within PowerBI based how they should exist (see Appendices: Relationship Diagrams).

3.2. Summary Reports

High-level summary reports aggregated the details for each system and person within the environment. The summary reports combined each of the measurements into a standard grade-point scale. A distribution graph showed how systems or people landed within the ranges and allowed the user to filter result sets down through the use of built-in context filtering. These records then allowed the reviewer to drill down into the specific system detail records.

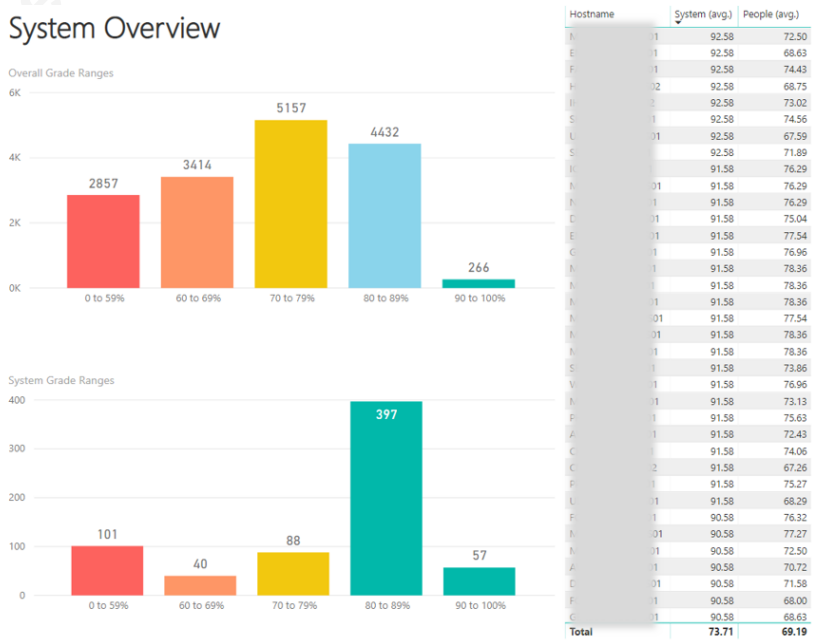

Figure 3: PowerBI System Overview

3.3. Detail Reports

Upon reviewing the summary-level details for a system or person, a common question arose: "What measurement is driving this score?" The detail reports utilize the inherent drill-through feature of PowerBI to allow the reviewers to examine the contributing factors to each of the scores. Additional information collected allowed the Information Technology and Security team members to review system details without needing to navigate to external systems.

Figure 4: PowerBI Detailed Report

4. Results of Experiment

In collecting and analyzing each of the data sets, three main lessons arose. The first was a better understanding of how a person's security mindset impacts the operational security of a system. The second was an understanding of the impact that missing values and/or out-of-sync collections have on metric collections. And finally, the complexities of developing an overall vulnerability score for a system became evident.

4.1. Comparing System Security to User Security

The security mindset of the people who have access to systems can have a direct, measurable impact on the operational security position of a system. A majority of the

systems had solid security configurations based on industry standards. However, when people's security awareness and phishing measurements were applied, there was a consistent drop (~11%) in the overall system score. While each environment has different training and education for their staff, the impact of people who fail to keep a security mindset was easily identifiable.

4.2. Population Collection Issues

While each population is collected and scored independently, when combined into the reports the results were inconsistent. These broke down into two categories. The first revolved around timing issues. Collection of system or people populations that occurred on different days occasionally resulted in orphaned population values. For example, a system in the Asset management system was listed as "In Use" but had no results in the vulnerability management system because the system was new and had not been part of the scheduled vulnerability run yet. These types of issues were resolved by running the collection process on subsequent days, through the use of scheduled jobs and naturally worked themselves out.

The second issue revolved around process breakdown issues. When processes required human interaction, breakdowns occurred that were invisible to the Information Technology and Security teams. These mostly involved the handling of assets. The adding or removing of systems from the asset management system, and were not modified appropriately from the scanners used for vulnerability and policy compliance management, resulted in orphaned records. In other cases, conflicting Group Policy Objects prevented remote access from Information Technology management systems even though they were properly deployed. Each of these conditions was easily fixed upon detection, made possible by the -1 score assigned to each. Until that point, the Information Technology and Security teams had not been aware of the situation. While fully automating these items would ultimately solve these issues, the ability to score orphaned records helped expose the scope of inconsistent processes within the environment.

4.3. Vulnerability Scoring Complexities

While certain measurements were straightforward to determine, a standard method to determine the overall system vulnerability score was more difficult. The NIST standard defines how to assess a single vulnerability through the CVSS process. However, choosing how to combine vulnerabilities CVSS scores from a system, to a single value presented several different possibilities.

4.3.1. Straight Scoring

The straight scoring approach is what was initially performed (as described in section 2.5.2: System Vulnerability), and while simple, it weighed heavily on systems that showed *any* vulnerability. Systems that contained a single CVSS 3 would score 70% (C Grade) with a straight scoring approach, while systems with an average CVSS value of 5 would score 50% (F Grade). For a system to score reasonably well, every single vulnerability would need to be addressed. While a noble goal, this type of scoring did not provide clarity to Information Technology or Security teams as to what actions to take, nor did it provide a better understanding of the security stance of the environment, as most systems scored very low.

4.3.2. Scoring on a Curve

The second approach scored system vulnerability on a curve. The system that had the lowest CVSS average value would be assigned 100% of the points, and all other systems would be evaluated based on the results of the best system. Unfortunately, this approach presented the likelihood of *decreasing* the clarity of understanding of the environment. In certain conditions, such as when the best system has a mediocre score, the scores of every system were overly raised. In turn, this resulted in the data showing that the environment was doing better than it was. While this approach would be useful in environments where a robust vulnerability management program exists, it does require more understanding of what the scores mean and careful observation.

4.3.3. Fixed Grades

The approach chosen in the end was based the scoring framework on business risk, and a qualitative severity ranking structure defined by NIST (National Institute of Standards and Technologies, 2020). The scoring matrix, as shown in Table 5, allowed the

flexibility to weigh the final averaged CVSS in a vulnerability management program, and for future adjustments once the program matures.

CVSS	Percentage	NIST CVSS Risk
0.0 – 0.9	100%	None
1.0 – 1.9	95%	Low
2.0 – 2.9	90%	
3.0 – 3.9	85%	
4.0 – 4.9	80%	Medium
5.0 – 5.9	75%	
6.0 – 6.9	70%	
7.0 – 7.9	65%	High
8.0 – 8.9	60%	
9.0 – 9.9	55%	Critical
10	50%	

Table 5: Fixed Grade Scoring Matrix

5. Additional Benefits

An additional benefit of the process of collecting populations and security aspects was documenting the historical state of the Active Directory environment. The daily collection of population information resulted in a set of snapshots that were useful when identifying the state of accounts and group membership on specific days. While any appropriately configured SIEM could retrieve information on individual changes to the environment, the snapshot provided a complete picture in a single view. Collecting additional items would be useful as a pre-forensic process to ensure that Information Technology and Security teams have a known state of the environment should an attacker compromise it (e.g., installed services; open ports; running processes).

6. Future Research Opportunities

Measuring an organization's security is an ever-growing, ever-changing process. As more information becomes available, the understanding of the environment becomes clearer if properly applied. During this research, several additional items arose that would be worth future research.

6.1.1. Data Classification

The research focused specifically on measuring the application of technical security controls on a system. However, not all data hosted on each of those systems were equal in value. By nature, systems hosting high-risk data, such as PII, PHI, or credit card data, represent a higher risk within the environment. Developing a scoring method to address data classification would require additional analysis but would ultimately further improve the understanding of the security position of an environment.

In addition to the data classification, the system's location also presents a valuable aspect to measure. Systems that have external interfaces, such as public web sites, inherently present a higher risk to attack than those segmented behind strict network access control lists (ACLs). Based on an organization's network architecture, additional measurements could be added as a multiplier on vulnerability and policy compliance scoring. For example, systems that are publicly exposed would have their vulnerability scores decreased by 20% thus raising the bar for a passing grade.

6.1.2. Trend Analysis

Initially built for viewing the current state of the environment, the framework does contain history tables of the populations, security aspects, and aspect measurements. A Security team could use the information to perform trend analysis to track how the environment, projects, and investments impact the security over time. As new technologies and training initiatives are onboarded, specific measurements could be developed and captured as a "before and after" state of the environment. This information could then, in turn, be reported to leadership to illustrate how the investments are making tangible improvements to security within the environment. The evaluation of the impact of security changes to an environment would need to be made on a per organization level due to the differences in each organization.

6.1.3. Security Program Measurements

With a framework in place to collect measurements, the inclusion of information security programs would be straightforward. As an example, the measurement of a vulnerability management program could examine scoring systems based on end-of-life software. This scoring could be as simple as, "Does the system contain End-of-Life

software, or not?" or utilize lists such as the End-of-Support (Center for Internet Security, 2020) software report to slowly increase the weight on software as the end date draws near.

Alternatively, the program could examine measuring vulnerability ages within the environment. Measurements could be as simple as scoring each vulnerability by how long it has been detected on the system, or by weighting the age score based on the individual vulnerability CVSS values.

7. Conclusion

Each organization must come to its own conclusion as to how to prioritize securing systems. The goal was to present an interface that would allow Information Technology and Security teams a way to remove the fog and uncertainty about how the applied security controls impacted the overall stance.

The process revealed several possible ways in which the collection of measurements could fail (e.g., system offline, disconnected during measurement, improper configuration, and lack of access). These deviations were not unexpected when dealing with separate and disconnected systems. However, the development of the framework to address collection gaps, orphaned records, and other anomalies was as important as the collection and measurement processes. By identifying systems that are trusted (i.e., connected to the environment), but not verified (i.e., unable to access), the environment became less foggy. Problems in deployment and configuration processes that previously would have remained invisible were easily identified due to poor system scores. People's failures in completing security awareness training, or phishing exercises, were identified and evaluated against how many systems to which their accounts had access. These "invisible" conditions represented a security risk that the organization believes were addressed through process and training.

While not every company possesses the same security tools used in this research, the approach and standard method of collecting, processing, reporting, and ultimately expanding are beneficial in clearing away the fog that obscures our understanding of the effectiveness of the security program.

References

Center for Internet Security. (2020, April 4). *CIS Benchmarks.* Retrieved from https://www.cisecurity.org/cis-benchmarks/

Center for Internet Security. (2020, March 30). *CIS Controls v7 Measures & Metrics.* Retrieved from Center for Internet Security: https://www.cisecurity.org/white-papers/cis-controls-v7-measures-metrics/

Center for Internet Security. (2020, April 5). *CIS Controls v7.1.* Retrieved April 2020, from Center for Internet Security: https://www.cisecurity.org/controls/

Center for Internet Security. (2020, April 5). *End-of-Support Software Report List*. Retrieved from Center for Internet Security: https://www.cisecurity.org/blog/end-of-support-software-report-list-2/

Chew, E., Swanson, M., Stine, K., Bartol, N., Brown, A., & Robinson, W. (2008, July). *Performance Measurement Guide for Information Security.* Retrieved from Computer Security Resource Center: https://csrc.nist.gov/publications/detail/sp/800-55/rev-1/final

Equifax. (2017, September 15). *Equifax Releases Details on Cybersecurity Incident*. Retrieved from investor.equifax.com: https://investor.equifax.com/news-and-events/news/2017/09-15-2017-224018832

Hubbard, D. W., Seiersen, R., Geer, D. E., & McClure, S. (2016). *How to Measure Anything in Cybersecurity Risk.* New Jersey: Wiley.

Limoncelli, A. T., Hogan, C. J., & Chalup, S. R. (2007). *The Practice of System and Network Administration* (2nd ed.). Boston: Pearson Education, Inc.

National Institute of Standards and Technologies. (2020, April 4). *NVD Vulnerability Severity Rating*. Retrieved from National Vulnerability Database: https://nvd.nist.gov/vuln-metrics/cvss

National Institute of Standards and Technology. (2013, April). *Security and Privacty Controls for Federal Information Systems and Organizations.* Retrieved from Computer Security Resource Center: https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final

National Institute of Standards and Technology. (2019, April 25). *Microsoft Windows Server 2019 Ver 1, Rel 3 Checklist Details.* Retrieved from National Vulnerability Database: https://nvd.nist.gov/ncp/checklist/914

National Institute of Standards and Technology. (2020, April 8). *Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy.* Retrieved from Computer Security Resource Center: https://csrc.nist.gov/publications/detail/sp/800-37/rev-2/final

Ravenel, J. P. (2006, December 21). Effective Operational Security Metrics. *Information System Security*, pp. 10-17. doi:https://doi.org/10.1201/1086.1065898X/46183.15.3.20060701/94183.3

SANS Institute. (2020, April). *CIS Critical Security Controls for Effective Cyber Defense*. Retrieved from SANS.org: https://www.sans.org/critical-security-controls/

Appendices

Relationship Diagrams

The relationship structure for the framework consists of three types of data: tables dedicated to system information (in blue), tables dedicated to account information (in green), and tables dedicated to people (in yellow) as illustrated in Figure 1. While other properties are collected and stored, only the core fields are displayed to illustrate the relationship structure.

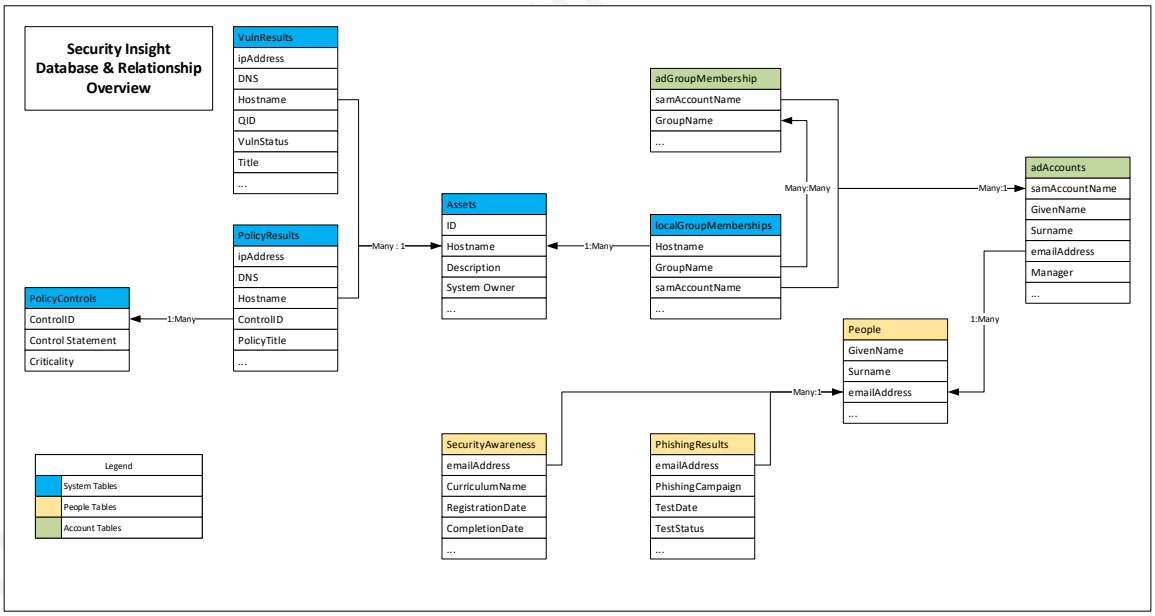

Figure 5: Relationship Diagram

Code Source Location

All code used to create the framework is available on GitHub:
https://github.com/snowstormSecurity/snowIntelligence

Testing Environment

- Traditional Active Directory Environment
- 600+ servers; 300+ user accounts
- Requirements
  - Ability to execute code on remote systems using PowerShell invoke-command functionality.
  - SQL Server
    - Setup: Ability to create a database, tables, and views
    - Operation: Ability to execute SELECT, INSERT, DELETE statements within the database
  - Asset Inventory

- Vulnerability Scan Results
- Policy Scan Results
- Inventory of People

Note: While Qualys was used for Vulnerability and Policy scans, any system could be used as long as a CSV export is available. This includes any other population or aspect collections.

Grade Scale

Grades were assigned based on the following table:

Grade	Percentage
A	90%+
B	80%-89%
C	70%-79%
D	60%-69%
F	0%-59%

Last Updated: July 4th, 2020

Upcoming SANS Training

Click here to view a list of all SANS Courses

SANS Amsterdam August 2020 Part 1	Amsterdam, NL	Aug 03, 2020 - Aug 08, 2020	Live Event
SANS Reboot - NOVA 2020	Arlington, VAUS	Aug 10, 2020 - Aug 15, 2020	Live Event
SANS FOR508 Canberra August 2020	Canberra, AU	Aug 17, 2020 - Aug 22, 2020	Live Event
SANS Amsterdam August 2020 Part 2	Amsterdam, NL	Aug 17, 2020 - Aug 22, 2020	Live Event
SANS Virginia Beach 2020	Virginia Beach, VAUS	Aug 30, 2020 - Sep 04, 2020	Live Event
SANS Philippines 2020	Manila, PH	Sep 07, 2020 - Sep 19, 2020	Live Event
SANS London September 2020	London, GB	Sep 07, 2020 - Sep 12, 2020	Live Event
SANS Baltimore Fall 2020	Baltimore, MDUS	Sep 08, 2020 - Sep 13, 2020	Live Event
SANS Munich September 2020	Munich, DE	Sep 14, 2020 - Sep 19, 2020	Live Event
SANS Network Security 2020	Las Vegas, NVUS	Sep 20, 2020 - Sep 25, 2020	Live Event
SANS Northern VA - Reston Fall 2020	Reston, VAUS	Sep 28, 2020 - Oct 03, 2020	Live Event
SANS San Antonio Fall 2020	San Antonio, TXUS	Sep 28, 2020 - Oct 03, 2020	Live Event
Oil & Gas Cybersecurity Summit & Training 2020	Houston, TXUS	Oct 02, 2020 - Oct 10, 2020	Live Event
SANS OnDemand	OnlineUS	Anytime	Self Paced
SANS SelfStudy	Books & MP3s OnlyUS	Anytime	Self Paced