March 2020

RESTRICTED DEVICE DATA SANITIZATION AND RANDOMIZATION

Mikhail Mooney

RESTRICTED DEVICE DATA SANITIZATION AND RANDOMIZATION

AUTHORS:

Mikhail Mooney

ABSTRACT

Presented herein is a Trusted Access Platform (TAP) that assists customers with highly restricted data access to solve their business critical challenges. The TAP sanitizes, anonymizes, and randomizes data from network devices, provides deeper analytics visibility into customer networks, mitigates severe outages and improve operational efficiencies through differentiated dashboards.

DETAILED DESCRIPTION

With compliance regulations or the risk of unintended consequences, customers are wary of sharing data with vendors. This wariness creates a big challenge, prolongs unnecessary downtime, and constrains vendors from providing the best customer experience. As an example, troubleshooting an outage within a classified facility can be extremely difficult due to the inability to share device configurations and command output without a massive, time consuming sanitization and approval process. This restriction results in an extended resolution time (e.g., days), even in a Severity 1 situation where the network operation is completely impacted.

In addition, personally identifiable information (PII) can be present in device configuration outputs, sometimes in predictable location and sometimes in random locations. Finding and manually cleaning all PII information, as well as getting the approval that the sanitized output meets customer restrictions, can take a huge amount of time. There can also be additional constraints, such as rules governing the movement of data from one classification to another.

The problem addressed herein is the ability to quickly sanitize customer device data and provide it to the customer approval process for publishing to the vendor for use in troubleshooting / analytics. This is a critical gap in the servicing of devices located in highly restrictive or sensitive environments. Financial, healthcare and research and

development customers that are charged with protecting identifiable information are affected by these challenges.

Here is an example of the challenge in a real world scenario:

- A network outage occurs in a classified facility.
- The customer leaves the facility to call Tech Support, asks for help and tells the Tech Support engineer that they have a problem, but cannot share any details of the issue or the environment the issue is occurring in.
- The Tech Support engineer asks for any information from the device(s) having issues. The customer tells the Tech Support engineer that they cannot send any device information until it is sanitized and approved, which will take hours to days.
- The customer requests generic troubleshooting instructions for a generic device in a generic situation.
- Once that information does not work, the customer repeats the process until either they solve the issue or they get approved sanitized data to send, which may not include the information needed to troubleshoot the problem.

In summary, device data from highly restrictive/sensitive environments cannot be quickly sanitized to an acceptable level and approved in a fashion that still provides the necessary information for the performance of analytics. Presented herein is a method for creating data files that are anonymized using one-time tokens to create salted, hashed identifiable data randomly during every collection, allowing safe transmission outside the customer environment every time.

FIG. 1, below, illustrates a Trusted Access Platform (TAP) randomization process in accordance with techniques presented herein.

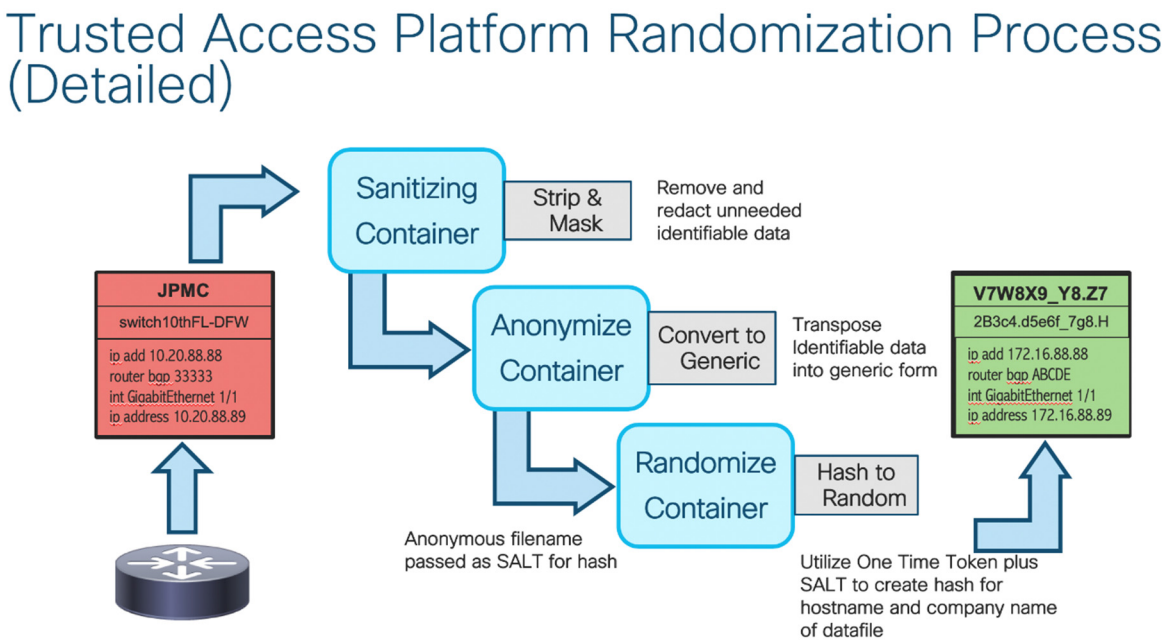

FIG. 1

The architecture of the techniques presented herein is created through the combination of Data Sanitization, Anonymization, and Randomization of components and processes, as detailed below. including:

- **Data Acquisition Module**: This module performs the collection of device data inside a restricted environment by creating and launching a highly hardened Sanitization container-based server with custom scripts for sanitizing the initial data file. The collection module uses dissolvable agents so that device specific data is not stored or maintained once collected and passed on to the Data Sanitization process.
- **Data Sanitization**: Using custom scripts and data identification processes, the collected data sets are sanitized by masking known PII data that is not required for the performance of analytics, and then transforming data required for analytics into a random, associated form that maintains consistency throughout the data set.
- **Data Anonymization**: Once the clean formatted data files have been created, they are abstracted using custom scripting to create randomly named data files. This process eliminates references at the file level to the specific customer or device. This random data file name is used in the Randomization process as a SALT for the encrypted hashing

function. Once the data has been abstracted, a highly hardened Randomization container-based server is created and the anonymized data files are transferred to this new container. The Anonymization container is then destroyed.

- **Data Randomization**: The Data Randomization process takes the random, abstracted data file name and passes it to the encrypted hashing function as a SALT. The output of this process is a salted, 256-bit Blowfish encrypted hash that is 64 bytes in length. This hash is then used to replace the data file names and the device specific data in the data files to prevent reverse engineering the data to discover the data owner, device or location that this data came from. Once the data files have been Randomized, the data files are transferred back to the Analytics Platform for use in analytics-based reporting. The Randomization container is then destroyed to remove the randomizing hash.
- **Data Analysis**: Once the Randomized data files have been transferred to the Analytics Platform, the data will maintain data relationships used for performing device specific analytics. The analytics results, if performed in a batch of devices, will be randomized and will not be able to specifically identify a particular device. This process is most useful when performing analytics on a specific device for consulting or troubleshooting where the data owner performing the analysis knows what device the data is from but cannot risk exposure if the data files were compromised.
- **Data Review and Approval for Publishing**: After Randomization, the data files will be available for review by the data owner through the use of the Trust Portal. The Trust Portal provides visibility to the actual format of the Data at Rest in the Analytics Platform, allowing the data owner to review and approve the cleanliness of the data prior to approving it to be published outside of the data owner's environment. This process creates Trust between the data owner and Cisco, allowing Cisco to perform consulting, analytics and troubleshooting on devices quickly that cannot be currently analyzed due to device data access restrictions.

There are two primary method that may be employed to identify PII data, post-acquisition of the base files. The first method includes the identification of a fairly exhaustive list of data that is considered PII, and the utilization of an extensive library of 263 custom Regular Expressions that can be used to search out those types of PII data, rather than merely matching on a specific command and masking the data if a line is

matched. The second method uses scripting that will analyze a configuration looking for specific keywords and using those to determine lines/parameters that contain info that needs to be masked or transformed in order to anonymize. The method may be made more effective/automated through the creation of a Machine Learning (ML) model that learns what data could be PII and recognize it in the context of configs/telemetry/streaming data, creating the ability to sanitize in transit. This new process would also allow the system to find/sanitize PII in unusual locations such as interface descriptions and banner messages more accurately.

Algorithm / Pseudo-code:

1. Create on-demand hardened container instance from pre-built library of server containers to perform Data Sanitization.
2. Launch dissolvable agent inside restricted data environment and run script to collect device data files.
3. **sanitization.py**: Run Sanitization scripts against collected data files.
4. Create on-demand hardened container instance from pre-built library of server containers to perform Data Anonymization.
5. Transfer Sanitized data files to Anonymization container.
6. Destroy Sanitization container.
7. **anonymize.py**: Run Anonymization scripts against collected data files.
8. Create on-demand hardened container instance from pre-built library of server containers to perform Data Randomization.
9. Transfer Anonymized data files to Randomization container.
10. Destroy Anonymization container.
11. Generate random One-Time Token to be used in the Randomizing hash process.
12. **randomize.py**: Run Randomization scripts against collected data files. Use the Abstracted file name from Anonymization process to SALT and the One-Time Token the encrypted hashing during the Randomization process.
13. Transfer Randomized data files to Analytics Platform.
14. Destroy Randomization container.

15. Review the Randomized data files for approval and publishing outside of data owner restricted environment.

16. Use the Randomized data files in analytics or perform troubleshooting.

Advantages of the Data Anonymization and Randomization Process:

The process of data anonymization and randomization presented herein automates many functions that currently take a large amount of time, causing delays in performing useful activity on device data. This automation includes a number of processes that when combined creates a highly secure output that can be used by a customer to use in our analytics or to provide outside of their environment quickly without fear of exposing identifiable data causing an opportunity for being compromised.

The uniqueness of this method is the multi-step process that translates the data, in multiple ways, into a form that both meets highly restricted regulations yet is still usable in performing analytics. In particular, the inline combination of the Sanitize, Anonymize, and Randomize concepts creates a multi-layered data cleaning process that will quickly create an approved version of customer device data that can be shared outside of the customer environment without causing exposure to the customer. Allowing the customer to not only approve prior to publishing the randomized data file, but also customize the scale of sanitization are unique components of this process as well. The proof point is simple as the problem exists on a major scale, and is of such importance that a solution is widely desired, and would have been implemented if a suitable one existed. The techniques presented herein provide a solution that is composed of existing components to create a new function that solves this wide-scale issue that must be solved.

The TAP architecture presented herein provides a trusted platform that enables customers to approve and share device data securely. The TAP architecture provides a sanitization method to collect sensitive device data without compromising security posture. The techniques can include the use of on-demand containers to create a secure one-time processing facility for sensitive data, which is destroyed immediately after use. The TAP architecture also provides an anonymization method to create an abstracted data set, maintaining data relationships for use in analytics. Moreover, the TAP architecture provides a custom randomization method performed using an anonymization string for

combining with the One-Time Token to SALT the encrypted hashing function. This hashing function creates a 64-byte string which is used to replace identifiable data components to eliminate reverse engineering and exposure of data owner's information. Finally, the TAP architecture enables automation of the entire process to quickly create a secure data set that can still be used for performing analytics on the output.