Recent Advancements in Cloud Security

Matthew O. Mitchell
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
Mitch723@morris.umn.edu

ABSTRACT

The issue of cloud security is an ever-present problem as more businesses are moving to the cloud. Startups' cloud services that are potentially unsecured are appearing more often. Information that is text based is moved between browsers and different windows, and information can easily be disclosed accidentally. In this paper, we look at different possibilities for security on the client side to enforce policies that are set in place with little to minimal impact on performance in day to day activities.

Keywords

Cloud security, Security, Service Level Agreement, browser-based middleware, data tracking

1. INTRODUCTION

As the Internet is getting faster and devices are getting smaller and easier to use, the cloud is becoming a convenient place to store and access data. Having an off-site location is convenient and doubles as a form of security. It is convenient because the cloud is easy to access with a stable Internet connection. It also provides a data backup since the information is stored off location,which is helpful in the event of environmental or physical damage since there is a backup in the cloud.

As hardware becomes more efficient, cloud service becomes much more affordable as data storage. For example, Amazon prime offers unlimited photo storage. As the cloud becomes easier to access we have to worry about what is being saved. Since the cloud so far has been convenient to share documents and upload data, the cloud is taken for granted as being safe and secure. This is an assumption that will continue as the cloud becomes more and more prevalent in daily use. Data security in the cloud has been taken for granted, but as security breaches happen, the data can be used for malicious intent immediately.

Some types of data, such as data governed by HIPPA and FERPA have privacy rules governing how it can be accessed and by whom. Employee information, bank records, social security numbers, phone numbers, and home addresses stored in the cloud should be handled especially carefully because they are an appealing target. Sensitive information such as these examples may need special precautions such as encrypting it before it is uploaded. With all this sensitive data containing hundreds if not thousands of individuals information, cloud service providers (CSPs) need security solutions that work without creating a big impact on performance.

Some key points of all the services is the CSPs ability to provide a competitive service as opposed to building a local IT department that wouldn't be able to scale on-demand as a CSP would. Most if not all of the services will always be up to date with the current patches.

UMM CSci Senior Seminar Conference, April 2018 Morris, MN.

2. HOW CLOUD WORKS

The cloud is where much of our data resides. For example you're using Google docs to edit your documents on a browser,later on a desktop computer added a picture to the document using your cell phone, this convenience is the magic we have come to know as the cloud. With so many different options available to access our data on different platforms, security is a real concern. As shown in figure 1, any device that is connected to a network has the ability to connect to the cloud, using applications that have access to data anywhere. There are three categories the cloud is organized into according to the National Institute of Standards and Technology: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS)[8].

CSPs have a Service Level Agreement (SLA) with clients that outlines what the CSP will provide and what the client should expect. The CSP and the client reach an agreement on what type of service the CSP will provide, including the type of security that the CSP will provide. The SLA will also outline how much hardware the CSP will provide to the client. Until recently cloud network's speeds haven't been written into SLAs. With the high demand of modern clouds, internet connection has been included. As infrastructure gets cheaper for data centers, these CSPs can provide more for the same cost of operating.[6, 7]

Each and every device that connects to the cloud is using services that a CSP provides, the most common being SaaS where applications are available for the end user to access. This SaaS has no local installation. Instead it uses the cloud as the platform, which is normally accessed through the internet browser. This gets rid of the middlemen and changing how software is now being distributed.[9]

Most CSPs and IT places track computers by their Internet Protocol-Address (IP), which is typically based on location. Another way to track what connects to the cloud is by tracking the Media Access Control Address (MAC) which

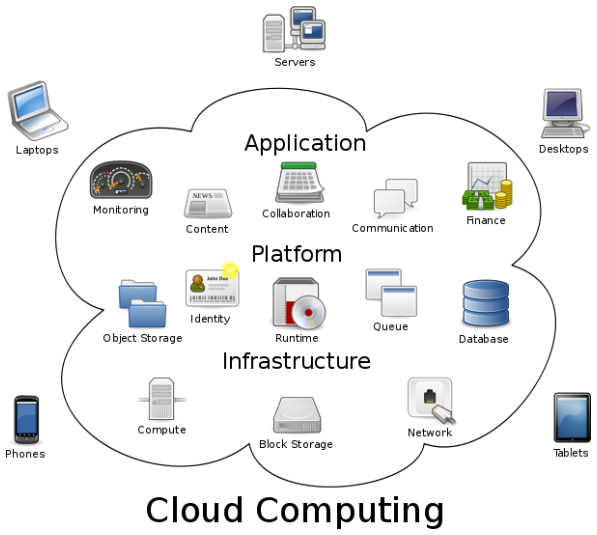

Figure 1: Overview of how the cloud connects our devices.

is the physical address that is unique to network adapters. LAN or Wifi both have unique MAC numbers. Using both to track who is accessing and where they are accessing cloud data make it simple to block IP of anywhere, or add just MAC numbers into the *white list* to allows access to the cloud from anywhere.

3. BIG BUSINESS CLOUD

As a business grows in size access to data becomes vital to the company. Access to important data is a reason why businesses are moving to cloud services. Business and corporations are moving from storing data in a more traditional manner, such as printed ledgers and spreadsheets, to a form of digital storage. Once this data is on the cloud, the data is easier to access both locally and in the cloud. Information technology departments have seen more of a demand for access to CSPs through personal laptops.[10]

The fast-moving pace of growing business has an ever growing set of data. This contributes to growth of the cloud where space is allocated to store the data, with the possibility of on-demand allocation of space making it simple to scale up and down. This makes the cloud very appealing to businesses not needing to support an infrastructure [6]. The cloud is a resource that is abstract: there is no physical location of the hardware that the user is aware of. Helping the client easily build a platform to store data by easily expanding the need for data storage.

Control over access is crucial to observe where the data is being accessed. Who has access to this information is vital in keeping it within an organization [6]. Given this critical point, there are admins who control the broad overview of who has access to the cloud, using a vast array of tools at their disposal. There are ways that an admin can limit access, which includes banning or approving applications, limiting the range of an IP address, and giving users access to different portions of the cloud. As shown figure 1, each client can see public interface of the cloud. If a client is not allowed to connect to an application, such as monitoring, this option would not be available to that particular client's IP.

3.1 Security Methods

Having so much control and "open" access to the cloud presents a problem for IT departments because of the increasing amount of clients that can connect to different CSPs and moving the same data over and over [10]. Here are a few existing approaches to mitigate unauthorized data leak:

- *The Data leak prevention system* inspects outgoing network traffic and keeps sensitive data from leaving a network.
- *Data flow tracking system* which takes data and is then tracked by a program when this data is moving from one point to another. This is typically used for tracking passwords, since this type of tracking has a costly overhead and grows along side the amount of data that is being tracked.
- *Static data flow analysis* tracks the data by looking at the program source code using program analysis, which is not ideal for legacy systems.
- *Browser-side enforcement* is doing work before data is sent to the cloud, for example, encrypting before uploading the data. This is not ideal since the data is encrypted and the CSP can't index the data, or use it for collaborative editing like Google docs. [12]
- *Client-side middleware* protects data by encrypting that data between user applications and the cloud. This data is encrypted since it's separated from the source code of the application. Meaning the application that created the data is the only one that can read the data by using a key to decipher the text, and not in a easy to read format. [9, 4]

As with any organization, when individuals release sensitive information, it can be destructive. This breach is hard to find because of how easy it is to transfer data from one medium to another. This doesn't mean that employers should restrict how employees access data, but they can find a way to endorse a practice and create a simplified robust environment [5]. This has caused problems for IT departments across the world since it's simple to move information. As users interact with different applications throughout the course of a business day, a file that could contain sensitive data might be shared with others accidentally. By using different methods to monitor the network the circle of security becomes more inclusive. [6]

3.2 Multi-Tab Cloud Issues

As cloud usage grows, so do different forms of data: mp3, docx, movie and many more. Copying and pasting text from Google Docs into Evernote is a few clicks, and the data will be moved over. This can be prevented by having a middleware browser-side plugin that is always looking at the data within internet browser tabs a user has open. This plugin, BrowserFlow has a Policy Enforcement that looks up a security label. Based on this label it decides to either allow the data to be uploaded to the cloud or prevent it.

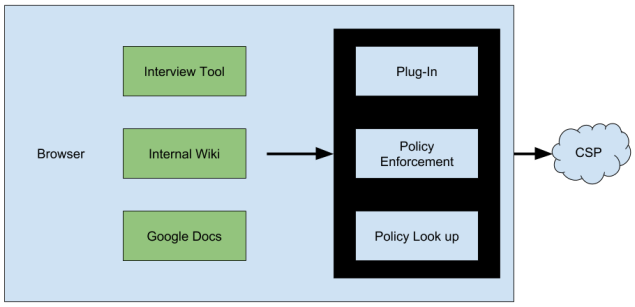

Figure 2: A Browser Middeware option.

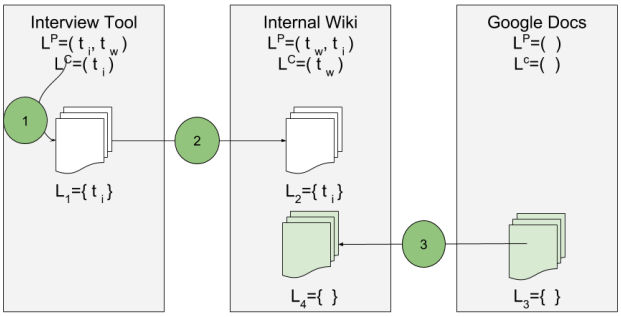

Figure 3: Using Labels to look up policies for a given web page.

When a user wants to upload a paragraph to an untrusted CSP there is an override that can be preformed. This override depends on what privilege level the user has and what type of confidentiality the original document contained. The override is then noted in the system along with who authorized and where the data is going in case anything happens. If an unauthorized upload happens, BrowerFlow will either block the data from the client to the CSP or encrypt the data before uploading to the CSP allowing it to be used by the client but making it unlikely to be shared to others.[9, 1]

In figure 2 there are three web pages open in a browser. BrowserFlow is monitoring the Page text of the internal Wiki and Google Docs. Each window is assigned two labels L^P and L^C, standing for the privileges of the user and confidentiality of the document respectively. For example a front page website everyone would have access to versus student data only select users would be able to see. Each tab is given a set of labels; some are known and trusted, others are not and are not assigned a label [1].

3.3 Digital Fingerprint

In order to track data that moves from one tab to another there are tags that are assigned to the chunk of text that is moved, L_i. This L_i becomes the n-th label as BrowserFlow tracks the data moving, as it moves from Interview Tool, that contains personal information, to the Internal Wiki this label becomes L_2, creating a trackable system where data is moved. Each label has tags that keep track of where the text comes from. In Figure 3 each tag is assigned when moving from one page to another. Every CSP has a label marked L^P that is assigned two tags (t_i,t_w). The first t_i is assigned by the user, the other t_w is assigned by a security admin, these tags controls who has privileges to these CSPs.[1]

As shown in figure 3, Google Docs doesn't have any labels since this was created by the user. If the user is to copy the data from one tab to another, it will only do so correctly if $L_i \subseteq L^P$. If this condition is met then BrowserFlow will allow the data to be uploaded to a cloud service.[9]

A fingerprint is created by using an algorithm for plagiarism, this allows BrowserFlow to build a database and can find text that is similar to each other. Since this is a well studied subject there is plenty of information and resources that exist. This fingerprint is computed by removing punctuation, whitespace and character case. For a simple case take 6 characters from "Hello World!" and is converted into a hash value of 5 unique numbers. The string "helloworld", includes these 6-character windows: "hellow", "ellowo", "llowor", "loworl", "oworld" and might convert to sample values (52,40,50,12,22). Creating a window of 3 values from this we get (52,40,50), (40,50,12), and (50,12,22). Choosing the minimum values that are presents in the windows gives (40,12). 40, and 12 are the smallest overlapping numbers in different windows.[9]

Every document has a fingerprints and there are two ways the fingerprint are computed. One is to look at the whole document and the other is paragraphs. Information smaller than paragraphs is not confidential in nature, such as fragments of a sentence. When moving data from Doc A to B, where F is the fingerprint of the document or paragraph, using the formula below gives us a value from zero to one, which is the percent that a document contains, similar text to other known documents.[9]

$$D_{document}(A,B) = \frac{|F(A) \cap F(B)|}{|F(A)|}$$

$$D_{paragraph}(A,B) = \frac{|F(A_p) \cap F(B_p)|}{|F(A_p)|}$$

This offers a bit of overlap depending on the confidentiality label that is assigned to the initial paragraphs. $D_{document}$ is how much the total document is a copy, and $D_{paragraph}$ for how much each paragraph is a copy. There is a threshold that can be set by using the above formula, giving the admin control over how much percent a Doc can have similarity to the original text.

$$D_{doc}(A,B) \geq T_{doc}(A) or \exists A_p \in A : D_{par}(A_p, B_p) \geq T_{par}(A_p)$$

This formula refers to both documents and paragraphs. Here the threshold is set by an admin of the original document. T refers to the threshold, for example for $T_{par}(A_{p1})$=0 and $T_{par}(A_{p2})$=0.5 then in this case 50% of paragraph 2 contains information from another document in the database[9]. The plagiarism algorithm is used in order to figure out when to allow a user to send off data or blocking the data by throwing an error, or encrypting the data so only the user who created the document is able to access it. This allows the end users to be freer without the direct control and yet not risk accidental data disclosure.[9]

3.4 Results of Middleware

Client-side Middleware works well when tracking the data that moves between tabs, and each tab could be a different

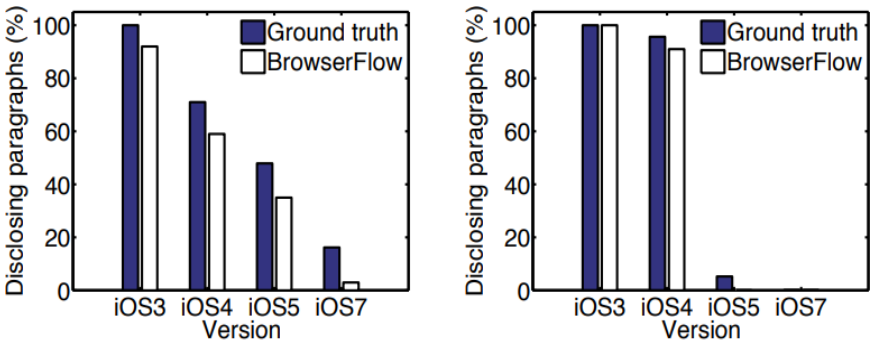

Figure 4: Using Labels to look up policies for an given web page.

CSP [6]. This flow of data is hard for an individual CSP to track, implementing a reliable plug-in on the client side can be the best way for organizations to track data [7]. In order to test how well BrowserFlow find similarities, researchers compares results to a "ground truth". The ground truth in this case is a human who is looking for similar content between the two different versions of manuals. In Figure 4 we can see how the two relate.[9]

The threshold of T_{par}, as shown in figure 4, is set to 0.5 or 50% of the paragraph is considered to be a copy and paste of the former document. Over 90% of the time BrowserFlow matches what the human expert considered a disclosing paragraph. This an ideal result since this process is automated and BrowserFlow was only set to 50%, meaning there is room for improvement [7]. However there were some false positives for short paragraphs, thus showing the limitation of the fingerprinting algorithm.[9]

3.5 Larger sets of data

Since BrowserFlow runs asynchronously, it is important to look at the speed of the algorithms on larger tables of data. If for instance there was noticeable latency or "lag" from the time BrowserFlow runs as middleware and sends a signal to the CSP, it would create an error on the CSPs end.[9, 10] When BrowserFlow runs a check on every keystroke it could be assumed some type of lag would be present. Since the fingerprint data is stored in RAM memory, it allows fast lookup times using a pairwise hash table giving $\mathcal{O}(n)$ look up in the worst case [2]. The hash table allows BroswerFlow to have response times $\leq 30ms$ over 85% of the the time BrowserFlow queries the database to check if it's okay to send the Doc or not. This delay is not noticeable to either the client or the CSP since most CSPs have tolerance built-in due to normal network latency [3]. Running all the time one could assume that it would affect other applications such as spell checking. But since it only exchanges information between the browsers and the CSP it does not affect other applications [6].

4. MULTIPLE CSPS

In order to keep up with modern day applications, instant responses to a client-side request is a key point in any modern day programming objective [11]. Most CSPs offer an on-demand type service, meaning space is allocated automatically [3]. One possible issue is that CSPs may have different SLAs with clients. Each CSP and client will have unique needs when completing an SLA. Within the SLA there are measurable Service Level Objectives (SLOs). Each SLO is a feature the client wants and a metric value is assigned to help measure an how effective an SLA is compared to the needs of the client. [11]

The SLO helps a client choose a CSP that has the best options. Organizations commonly have multiple clouds for security and keep data separate from each other [10]. When cloud service depends on multiple CSPs infrastructures becomes important. If different departments are running individual SaaS and they share data, then there is some dependency between different CSPs. In order to maintain a quality of service the dependencies between the clouds have to be quantified from weak (1), medium (2), strong (3) depending on the needs of the client. These dependencies can be from sharing the same database to depending on another CSP to provide security. [11, 7]

4.1 Service Level agreement

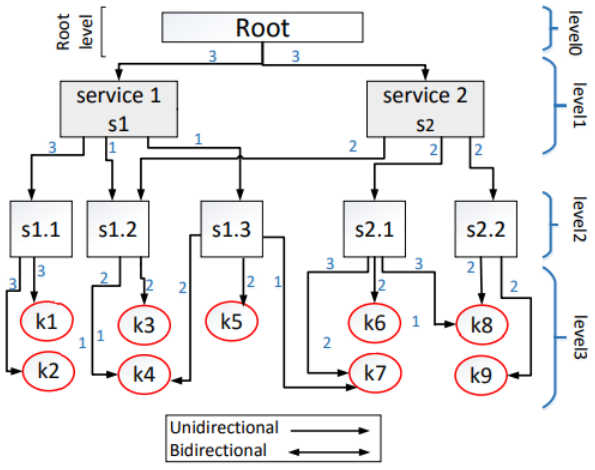

Figure 5: SLA flow that has dependencies.

When a client goes shopping for a new CSP, requests are sent out to different CSP containing SLO's request in order to find the best price and performance for what is needed. The SLA is setup in a hierarchy shown in (Figure 5). The level 0 is a SLA request for the client, level 1 is a CSP,

level 2 is the services that the CSP will provide, level 3 is the SLO that the client outlines. As Figure 5 shows there are dependences, direct and indirect, between the different levels. There are a few conditions have to met to be a valid SLA the formula shows one conditions. [11]

$$\forall s_1, s_2 \in S.s_1 \rightarrow s_1, s_2 \Rightarrow l(s_2) \; or \; l(s_1) + 1 = l(s_2)$$

The above formula is one of four that is outline to make a valid SLA and states the service is only dependent on the next lower level. In order for an SLA to be in considered a valid state a set of services (S) where a service s can't depend on same or lower level service that is province by CSPs. The s_1 or s_2 is from the set of services then it has to depend on another SLO from a different service or the level above it as shown in figure 5. For example, s1.1 can depend on s1 or s2. Each SLO is assigned a k(n) value where n denotes the SLO number. All services depend directly or indirectly on an SLO. If an SLA is considered in a valid stat then each CSP can begin to be assessed for the client needs.

$$CSP_{1,k}/CSP_{2,k} = \frac{v_{1,k}}{v_{2,k}}$$

The values are assigned from zero to one using the above equation. Where v_1, k, k denotes a SLO that the client is requesting for a CSP. If the CSP can provide the SLO then they will have a numerical value that is added to the over all score of v(n), where n denotes the SLO number, otherwise when compared to other CSP it will be assigned a zero. Each CSP is compared to each other using a matrix to build a graph to compare different CSPs based on the SLOs request.[4, 11]

In order to have a accurate representation of what an CSP is providing in comparisons to what the client needs, the SLO k is normalized by prioritizing what the client is looking for by removing services that the CSP provides that are not needed by the client. By using this approach business are able to quickly search for the best option accessible. Since CSPs have a lot of services but not what the client is looking for, by normalizing this process it can help clients quickly find the correct service. [7]

5. CONCLUSION

Everyday users flock to the cloud with the promise of ease of usage. This has caused the rapid growth of Cloud Service Providers and end-users who are increasingly looking for the best. There are many different options that are offered for security. Using the BrowserFlow method allows freedom for the end-user while giving the CSP and organizations the ability to keep track of sensitive information.

As more CSPs are coming on-line the term "Cloud" becomes synonymous with secure. This leads to a false pretense that all cloud's security are created equal. When looking for a CSP automating and quantifying security the process will help CSPs and clients. The clients are able to easily compare what a CSP has to offer from a choice of many, and CSP is able to decide what kinds of upgrades it should invest in to make itself more appealing, or a startup CSP can decide what to focus on when comparing to other CSPs.

Client-side software is not the end all be all [6]. It only offers one solution of the many that were talked about in 3.1. Given that the cloud will have latency problems, uneven data load causing slowdowns, and end users who will try to work around any security that is implemented, it's a combination between all the different features that is needed in order to keep sensitive data from leaving the controlled and secured area.

6. REFERENCES

[1] F. Armknecht, J.-M. Bohli, D. Froelicher, and G. Karame. Sharing proofs of retrievability across tenants. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pages 275–287, New York, NY, USA, 2017. ACM.

[2] T. H. Cormen and C. E. Leiserson. *Introduction to algorithms, 3rd edition*. 2009.

[3] D. Felsch, M. Heiderich, F. Schulz, and J. Schwenk. How private is your private cloud?: Security analysis of cloud control interfaces. In *Proceedings of the 2015 ACM Workshop on Cloud Computing Security Workshop*, CCSW '15, pages 5–16, New York, NY, USA, 2015. ACM.

[4] D. Koutsourelis and S. K. Katsikas. Designing and developing a free data loss prevention system. In *Proceedings of the 18th Panhellenic Conference on Informatics*, PCI '14, pages 14:1–14:5, New York, NY, USA, 2014. ACM.

[5] H. Liu, C. Li, X. Jin, J. Li, Y. Zhang, and D. Gu. Smart solution, poor protection: An empirical study of security and privacy issues in developing and deploying smart home devices. In *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, IoTS;P '17, pages 13–18, New York, NY, USA, 2017. ACM.

[6] R. Mahmud, F. L. Koch, and R. Buyya. Cloud-fog interoperability in iot-enabled healthcare solutions. In *Proceedings of the 19th International Conference on Distributed Computing and Networking*, ICDCN '18, pages 32:1–32:10, New York, NY, USA, 2018. ACM.

[7] V. Martin, Q. Cao, and T. Benson. Fending off iot-hunting attacks at home networks. In *Proceedings of the 2Nd Workshop on Cloud-Assisted Networking*, CAN '17, pages 67–72, New York, NY, USA, 2017. ACM.

[8] P. Mell, T. Grance, and NIST. Sp 800-145, the NIST definition of cloud computing.

[9] I. Papagiannis, P. Watcharapichat, D. Muthukumaran, and P. Pietzuch. Browserflow: Imprecise data flow tracking to prevent accidental data disclosure. In *Proceedings of the 17th International Middleware Conference*, Middleware '16, pages 9:1–9:13, New York, NY, USA, 2016. ACM.

[10] E. Rios, E. Iturbe, and M. C. Palacios. Self-healing multi-cloud application modelling. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ARES '17, pages 93:1–93:9, New York, NY, USA, 2017. ACM.

[11] A. Taha, P. Metzler, R. Trapero, J. Luna, and N. Suri. Identifying and utilizing dependencies across cloud security services. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '16, pages 329–340, New York, NY, USA, 2016. ACM.

[12] L. Wang, R. Nojima, and S. Moriai. A secure

automobile information sharing system. In *Proceedings of the 1st ACM Workshop on IoT Privacy, Trust, and Security*, IoTPTS '15, pages 19–26, New York, NY, USA, 2015. ACM.