Engineering Threat Modelling Tools for Cloud Computing

Muhammed Mustafa Aydin

MPhil

University of York
Computer Science

May 2016

The rise in the use of cloud computing has also been accompanied by increasing numbers of online security incidents and concern about the overall security of cloud services. Current threat modelling methods, one of the most common ways of identifying threats to systems, are not able to provide a useful remedy to this. The manual threat modelling techniques require experts and can take too long, and the automatic tools are aimed primarily at software developers and do not apply to systems.

This thesis presents the underlying theory for Cloud-COVER (Controls and Orderings for Vulnerabilities and ExposuRes), a threat modelling tool developed to identify threats to cloud computing systems. Cloud-COVER models the system under observation, and determines the priority of threats by using a system of relative preferences provided by the tool user. Cloud-COVER also looks at how threats from individual parts of the system present a danger to other parts of the deployment, identifying ways in which beachhead based attacks can be prevented. Cloud-COVER's model is abstracted in such a way that it is extensible, allowing users to shift the model's perspective to suit their own circumstances.

This work presents a number of major and minor contributions to security and threat modelling. The main contributions of this thesis are an alternative way of ranking threats by using relative preferences, and an extensible model which analyses the way threats can propagate through systems by looking at the permissions given to connections between instances.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisor Jeremy Jacob, for all his hard work and support throughout the period I have been working on this thesis. His encouragement has helped a great deal and motivated me to continue working through some difficult conditions. Most of the useful research directions my work has followed has come from his astute suggestions, and resulted in the work presented here.

I would also like to thank my parents, who have spent so much time worrying about me. They have given me a great deal of drive. Now that this period is at an end, I will try to use this to push me towards bigger and better things.

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, university. All sources are acknowledged as References.

Unless stated otherwise, all work in this thesis is the original work of the author. Part of the work in this thesis is taken from the following papers:

- M. Aydin and J. Jacob, "Cloud-COVER: Using User Supplied Valuations and Propagation Analysis to Prioritise Threats to Systems," European Intelligence and Security Informatics Conference (EISIC), 2015 [1].
- M. Aydin and J. Jacob, "Providing Extensibility of Attributes, Permissions and Threats for Cloud-COVER's Threat Modelling Tool". European Intelligence and Security Informatics Conference (EISIC), 2016 [2]

1 Introduction

1.1 Security

Computer security relies on anticipating threats to systems, and taking measures to respond to them. In security, defenders are always reacting to attackers, with every new attack requiring a quick response to prevent knowledge of the method spreading before defences against it have been developed. In addition, systems are getting ever more complex over time, requiring more detailed knowledge of how to protect against all threats.

The last few years have seen attacks reach the biggest scale yet, with the data of millions of people being hacked from organisations such as OPM [3] and Adobe [4]. Some security experts believe that the balance in computer security has shifted in favour of the attacker, with defenders now playing catch up [5].[1]

The increasing popularity of cloud computing has also raised concerns about the security of the hosted machines (also known as instances) and data. The virtual machine instances in the cloud are hosted outside of a user's material premises, possibly on the same physical machine as a competitor, and could even be in a country considered to be hostile. There are many possible problems with virtual machine security, such as issues with maintaining the security of increasingly large numbers of instances [6]. Also, some attacks previously only thought to be theoretical, such as attackers identifying specific target virtual machines within data centres and attacking them, have actually been demonstrated to be

[1] Bruce Schneier is of the opinion that at the current moment in time, the offensive capabilities available to attackers give them much more advantage than the defensive capabilities available to anyone guarding a system, but that over time this balance can be changed.

effective [7]. This gives rise to widespread concern about cloud security, which according to some sources is the primary worry preventing more businesses and users migrating to the cloud [8]. Chow *et al* describe one of the fundamental issues arising from cloud's unique delivery model as coming from the problem of outsourcing computation without outsourcing control of the hosted data [9].

There are many concerns about cloud services, although many of the concerns are actually older threats manifesting in new clothes due to the unique delivery model of cloud services. Many of the problems encountered in the cloud have actually been known about for some time. Virtualisation, the mechanism by which physical machines can share their resources among several virtual machines, was originally developed by IBM to enable multiple users to effectively access their (few) mainframe machines in the 1960s [10]. Most of the associated security problems with hypervisors therefore have also been known about for the same period. The notion of computing as a utility (something cloud computing is frequently described as) also originates in the 1960s, when it was proposed as the future of computing by John McCarthy [11].

Despite many of cloud computing's security problems having been encountered since then, there are in fact some aspects, such as issues arising from the mix of locations of user, host, and provider, that do present new problems [12].[2] Threats can also be considered from the perspective of the many hardware and network layers, not all of which are under the control of the cloud customer [13]. This includes governance issues such as compliance, which covers problems such as data location or the quality of service being provided by the cloud service [14]. Using digital forensic techniques to investigate successful attacks can also be problematic due to ownership and location issues of data on cloud services, such as network logs which are not routinely available to cloud

[2] A user may be based in one country, the physical machine and data may be in a second country, and the company providing the service may be based in a third country. There are many problems which can come from this model, with legal issues being a prominently used example.

customers [15].

In a distributed system, such as those in the cloud, the way that the machines in a deployment connect to each other may have implications for security requirements. The many options available to configure the internal workings and external connections of multiple machines makes understanding their security needs quite challenging, and getting increasingly complex as the size of the deployment gets larger. In addition, making even small changes to any individual system could have a significant impact on previously developed security defences. This could be for an individual instance, or for whole deployments, with the effects of any changes causing a domino effect through the network connections to the entire system.

Computer security is not just a matter of solving technical problems but a combination of people, process, and technology (also referred to as 'the triangle') [16].[3] Good security planning needs to incorporate from each of those in order to create good defences for any system.

Risk management is one of the important processes used to assess and improve security. As part of the risk management process, risk assessments are commonly used to try to manage threats to systems, and commonly utilise threat modelling, a specific type of modelling developed to help to understand threats to computer systems [17]. Many of the terms presented here are based on those presented in the ISO27000 series, and can also be seen in the Glossary presented at the back of the thesis [18].

[3]People refers to human strengths and weaknesses. Although people may know to keep their passwords secret, they may write them down in easily accessible locations, endangering security. Security education is therefore a way to improve the 'people' aspect of security. Process refers to procedural expectations which maintain overall security. For example, regular password updates are part of a good security process. Finally technology refers to any kind of engineering, whether software or hardware, which is used to protect systems. For example, this can be a firewall (software), or a fingerprint lock on a laptop (hardware).

1.2 Modelling

Models are a useful way of abstracting away and simulating the components and concepts within systems. By representing only those things which are needed in order to fulfil a given purpose, models can be used to analyse complex problems. The modelling of systems can be used to better understand the ways in which systems work and how they can be improved. By continual redevelopment and refinement, adding useful things and taking away less useful ones, models can be adapted and simplified to aid in understanding problems in relation to improving real life systems.

The mathematician George Box famously observed that, "essentially, all models are wrong, but some are useful" [19]. This observation illustrates the problem with modelling anything, which is that much of the detail needs to be abstracted away, leaving only enough detail in order to answer the question being posed. To this end, when developing models one of the most important things to think about is the amount of information included. Too much information risks creating a muddied pool, with useful information difficult to extract amongst all the data. Too little information can mean not enough useful analysis, leaving model developers to carefully consider the level of balance they require. The other thing to consider is the recipient audience, which will also influence this balance. For experts more information is needed to answer their questions, whilst novices obviously need simpler answers.

Security is not just about the need to protect an asset from threats. It is also about balancing the need to prevent access to unauthorised parties with the needs of the owner and permitted parties to access the asset [20]. It is this balance which is important to get right in any consideration of security. Finding the right balance involves trying to understand how different choices can end up affecting this balance. In computers, installed software may contain vulnerabilities allowing attackers to successfully breach the system, but they might be absolutely necessary to perform certain functions. Creating multiple user accounts and extending them

permissions may allow an unprecedented level of access to a protected resource, but it may be necessary in order to get a job finished. There are many considerations for the owner of a resource to think about.

When applying the same thought process to distributed systems, these questions can become more complex to answer. Which instances should be allowed to connect to each other? What kind of precautions need to be taken to protect against specific threats? With any more than a handful of instances, such questions could easily overwhelm anybody.

It is clear that modelling can be used to aid in the process of understanding how to make such decisions. In fact, modelling has been used to look at computer systems, or components within systems, in order to understand and improve their design and performances. It can also be used to identify threats, and to take measures to protect against those threats.

1.3 Threat Modelling Background

Risk management has been commonly used as one of the main ways in which to manage the security of systems, and threat modelling has become one way to help to achieve this. Threat modelling is used to identify threats, users can then try to fix the vulnerabilities and then, if that is not possible, to look for protections which do not allow for those vulnerabilities to be exploited. Threat modelling is a specific type of modelling, which involves making a model of a system or components of a system, and using this model to understand how attackers could take advantage of the architecture of that system in order to attack it.

Threat modelling originally developed as a manual process, looking at identifying threats to systems. The fact that no single way of modelling could identify all threats to a system meant that different models emerged. Microsoft developed the STRIDE [21] and DREAD [22] threat modelling approaches, both of which are manual methods. However, manual analysis needs to be performed by security experts and can prove costly, especially due to the constant need for re-evaluation in most development

processes. The development of more than one approach was partly due to the fact that some of the categories in STRIDE occasionaly overlap.[4] Later on the STRIDE process was incorporated in Microsoft's SDL threat modelling tool, helping to make parts of the process automatic [24]. This tool concentrates primarily on application development and is aimed at developers, and is of little use to system administrators. On Information Security Exchange, Adam Shostack (one of the primary developers on the SDL Threat modelling tool) is one of the most frequent commenters on the topic of threat modelling, and confirms this to be the case [25].[5] Although systems administrators need to think about security just as much as developers do, they need to take care of systems composed of many computers, running many different programs. Developers need only think about the single piece of software they are working on, and need not concern themselves with the code and security issues of other software. Therefore, systems administrators need alternative tools to support their own needs with regards to systems security.

Threat modelling is not just limited to using these tools or processes. Data flow diagrams are frequently used, as they are a useful way of thinking about how data moves through systems [26]. Threat modelling can also involve using penetration testing to find places to attack, involving applications such as Firesheep or Wireshark [27] [28]. In addition to this, attack trees can be used by considering attacks as tree structures with goals as the root and the different ways of reaching the root as branches [29]. By attaching different values to the branches reaching the goals, each path can be evaluated to consider how to prioritise defence within systems. They can help to consider any situation when given values for each branch for the situation under consideration. Sea Monster is an automated example of a tool used for attack trees [30]. Also, SeaSponge

[4] Larry Osterman notes on his Microsoft threat modelling blog that many threats between the classes Elevation of Privilege and Tampering of Data overlap, as well as some from the other classes. [23]

[5] Studying security encourages paranoia: the user identifying as Adam Shostak may be someone else masquerading as him. There is no proof either way, other than his display of knowledge of the subject in his many postings is consistent with him being genuine.

is a recently developed threat modelling tool targeting systems administrators [31]. However, despite the development of these processes and tools their adoption has been mixed, with their popularity not becoming widespread and many people choosing not to make use of them [32].

It is a mistake to limit the scope of threat modelling frameworks and tools to application developers alone. Systems themselves are also in need of protection from threats, and could make use of threat modelling. Regular systems do not currently benefit from threat modelling tools, a glaring omission. In the more complex scenario of distributed systems, there are currently no candidates. Yurcik *et al* observe the fact that the security of distributed systems are frequently managed with an attitude in which they are treated no differently than a single standalone system [33]. In addition, even some of the original developers of the STRIDE and DREAD threat modelling processes concede that the lack of academic rigour applied to them has been an issue [34].[6] In fact, most threat modelling tools and frameworks have been developed outside of any academic process. This provides us with the required motivation to look at research on the development of methodologies which can aid the threat modelling process for cloud computing. There is a clear need to develop such processes and theories, in order to increase the popularity of threat modelling to a wider range of users.

1.4 The Threat Modelling Process

A threat modelling process is the act of threat modelling. The goal of a threat modelling process is to produce a threat model. The threat model contains a list of threats to users of that system, so that they may take action against them.

There is no single definition of what threat modelling is [35]. It is generally considered to be a process of identifying threats to a system

[6] Although David LeBlanc does say that the lack of academic rigour is an issue, he also says that sometimes very useful things don't need to pass through the 'ivory towers of academia' to be useful

(or components of a system) by modelling the system in question, and in taking precautions to guard against the threats. Threat modelling is usually asset-centric, attacker-centric or software-centric, which allows for different ways to understand how the system being modelled can be attacked [36].

In the same way, there are no agreed list of steps which constitute a threat modelling process. However the general stages in which the process is carried out tend to retain common features wherever they are discussed. These involve the following:

1. Modelling the system and including components, connections, trust boundaries, and assets which need to be protected.
2. Analysing the model in a structured way to identify the way in which threats to the system exist and ways in which attacks could take place.
3. Finding protections against the attacks and implementing them.
4. Although optional, a good threat modelling process is never finished and should be repeated at regular intervals. This is because additional threats could have developed in the meantime. Additionally, considering the system from a different perspective, or even using a different threat modelling process, can also be used to identify additional threats to the ones identified in the original process.

Threat modelling can be approached differently. For example, although the probabilities of threats are considered in many methodologies, sometimes a threat modelling framework may state that it is not something which they consider [37].

1.5 Threat Propagation

Threat propagation is a big security problem for networked systems. It is very common to find that infiltrations to networked systems are often

achieved through an initial point of compromise (commonly referred to as a "beachhead"), after which the attackers find additional targets on the same network [38]. These instances which could be used as useful beachheads need to be identified, in order to prevent attackers using them as stepping stones to attack other better protected parts of the deployment.

For example, in a deployment with 2 instances (A and B) where instance B has much more valuable data, instance B is likely to receive much higher levels of protection. However, the fact that instance A (receiving much lower levels of protection) connects to instance B (and is trusted by it) means that an attacker could also use instance A to connect to, and attack instance B. Therefore the threat to the data on B, also has to considered to be a threat to the data on B *from A* as well. The threat can be said to have propagated. This is referred to in this work as *threat propagation*.

1.6 This Thesis

This thesis presents the work of the development of the underlying theories for a well engineered threat modelling tool for cloud computing. As a demonstration of its sound engineering, additional work is also presented on its extensibility, allowing users to shift perspective to their own circumstances.

1.6.1 Contributions

The contributions of this thesis are based around the underlying theory for Cloud-COVER (Controls and Orderings for Vulnerabilities and ExposuRes), a threat modelling tool for cloud computing [39]. The contributions are listed below, with the major contributions mentioned.

- Cloud-COVER makes a major contribution to security in the way it allows users to rate impact from attacks to their system, by using relative preferences between data security attributes (and the threats

which they represent to the system). This is in contrast to other methods which use numerical valuations.

- Cloud-COVER's major contribution to threat modelling is the way in which the analysis of threat propagation takes place. By looking at the connection permissions and their ability to violate specific security attributes, Cloud-COVER is able to identify which threats need to be protected against from other instances.
- Cloud-COVER allows users to change the perspective of the model to match their preferences. By allowing users to input any of three of the inputs which are used to determine threat propagation, users are able to adapt the analysis to one which considers issues which the default version of Cloud-COVER does not look at. This allows users to think about threats in different ways, an important part of a good threat modelling process.
- Cloud-COVER is the first threat modelling tool taking the perspective of a system owner, other available threat modelling processes and tools are aimed primarily at application developers or involve using manual methods.

1.6.2 Structure

The details of each of the chapters are included below.

- Chapter 2 introduces the underlying model for Cloud-COVER, a threat modelling tool for cloud computing. Cloud-COVER presents users with a prioritised list of threats to cloud deployments that they need to defend against. This chapter covers the the model representing the deployment, along with the user valuation system, which allows users to specify impact to their system by using relative preferences.
- Chapter 3 extends the analysis of Cloud-COVER, by discussing the feature of threat propagation, and how this is analysed in order to

produce the results presented to the user. The way in which the final results are ordered is also discussed.

- Chapter 4 covers the extensibility of Cloud-COVER, and how users are able to change the perspective of the tool to cover the issues relevant to their own specific circumstances.
- Chapter 5 presents a case study of users with different levels of security knowledge, their use of Cloud-COVER, and how this demonstrates the fulfilment of the original aims behind the development of Cloud-COVER.

2 Cloud-COVER

2.1 Introduction

The security of a system depends on many factors, related to every part of a system. This may include the operating systems, the application software, user permissions, and the networks connections, to name only a few. The way a system, its instances and connections may be configured is likely to make it, and therefore its security considerations, unique. Given so many factors and the complexity of trying to understand their implications, it can be extremely difficult for even the most knowledgeable security expert to determine how to approach defending any system. Threat modelling tools are one way in to help in identifying threats to systems. However, existing threat modelling tools are aimed mainly at application developers, and there is a need for ones which consider threats from the perspective of system owners and administrators.

Another major issue with threat modelling tools is the concentration on direct threats instead of indirect ones. Indirect threats are those which do not involve direct attacks on deployments, examples can include if attackers find where cloud services back up their data and access confidential data that way. This would mean someone could find their data compromised despite taking all necessary precautions to have protected their deployment, and the deployment never getting hacked. Users need to be made aware of threats like these. Other examples of this kind of threat are location based issues, where companies need to know that their data is kept within particular legal jurisdictions. As these kinds of threats come from cloud computing's delivery model, they are newer and have had less attention paid to them. They may be less well known and

therefore more in need of being presented to users in any threat analysis of cloud systems.

The other issue with threat modelling approaches and tools is their reliance on using numeric values to rank identified threats. The problem with using such values is that of being able to represent information about security numerically. This can be a problem, as different kinds of people may value things differently. For example, Dana Epp observes how during a threat modelling process developers (who are rarely security minded) tend to give low values to threats (usually a 1 or 0 out of 10), whilst security minded individuals rate almost all threats as a 10 [40].[1] This is a problem for the usefulness of any rankings produced by such methods. Cloud-COVER uses an alternative approach involving relative preferences between security attributes provided by the user, allowing rankings to be inferred from the provided preferences.

This chapter presents the underlying model of Cloud-COVER, a cloud security threat modelling tool. Cloud-COVER works by taking an input from the user and analysing the input model to order the identified threats. The input model takes in information about the cloud deployment, and information from the user about the priority of security attributes for their data. Using the provided information, Cloud-COVER models the deployment, and presents the user with a ranking of the most pressing threats in their deployment alongside countermeasures to these threats.

Cloud-COVER has the following selling points which set it apart from other threat modelling tools:

- Cloud-COVER considers threats to entire systems, not just low level details relevant to software and software developers, and is

[1] Bruce Schneier frequently comments on this kind of problem in information security as well as real life security, calling it CYA (Cover Your Ass) Security. The principle is that for anybody whose job involves responsibility for any kind of security, there is an unwillingness to allow people to think that they do not take all threats seriously. The result is that almost all threats are treated as a priority, as this ensures that they cannot be blamed in the event that something goes wrong. However, there is an obvious cost of wasting resources with such an approach [41].

intended to be more accessible to a wider variety of computer users (not just security experts).

- Cloud-COVER takes in user valuations as relative preferences between security attributes, not just as numerical valuations. This helps users to more effectively prioritise threats to their systems.
- Cloud-COVER considers the way in which threats propagate through systems, helping to make users better able to understand how to prevent attackers using less well protected instances to attack better defended ones.
- Cloud-COVER considers indirect threats as well as direct attacker threats to security.
- Cloud-COVER categorises threats into instance threats, and organisational threats, an important distinction which allows users to better understand the way in which threats need to be defended against to secure entire deployments.

This chapter is structured in the following way: first a discussion of the model used by Cloud-COVER to simulate cloud deployments is in Section 2.2, followed by a categorisation of threats in Section 2.3. The security attribute values for determining the rankings of the results are discussed in Section 2.4, related work is in Section 2.5, followed by possible further work in Section 2.6 and a discussion of the chapter follows in Section 2.7.

2.2 Model Requirements

The identification of the threats to a given deployment requires a number of factors. To start with, it requires a model capable of matching the deployment under analysis to the threats and exposures capable of causing damage to it. Prioritising those threats that do exist requires some

knowledge of their likelihood, and an understanding of the potential impact that they might have on the deployment.

Cloud-COVER is not intended to look at low-level details, and does not need to ask about information such as how data interacts with and flows through processes and threads. Although these are useful to identify how attacks may be successful at lower levels, it is important for Cloud-COVER to distinguish itself by identifying those threats at a higher level, which are just as important.

The main output of the tool should be a list of threats to the system. Details of these threats should include the instance they represent a threat to, and their priority. This information can be determined based on user input. The major steps to identify these threats and then order them according to priority is explained below, along with justifications for the choices made.

2.2.1 Stages

Before detailing the input to the system, an overview of the way in which the tool works may be helpful. Here a quick discussion of the major stages of Cloud-COVER's work is provided.

1. Stage 1: The user enters details of the deployment, which includes information about instances, data on the instances, and the connections linking the instances together.

2. Stage 2: The user enters their preferences regarding the relative valuations between the security attributes of all of their data.

3. Stage 3: Cloud-COVER analyses the provided deployment, and looks at the interaction between the connection permissions and the security attribute/permission table to see whether threats exist on the various instances, and where those threats may originate from.

4. Stage 4: The threats are ordered based on the user supplied preferences between data security attributes.

5. Stage 5: The results are presented to the user.

2.2.2 Input

Cloud-COVER models cloud systems by considering instances, the connections between them, and the data present on those instances. A UML figure defining how instances connect to each other and host data is in Figure 2.1. Instances are considered as nodes in a directed graph, with the connections as directed edges between them, such as the simple example presented in Figure 2.2. The data are represented as properties of those nodes. In Cloud-COVER's model the data itself is considered as the asset of a system, and ultimately the object of an attacker's target, whether to read, modify, destroy, or deny/delay access to it. Cloud-COVER therefore uses an asset-centric threat modelling perspective.

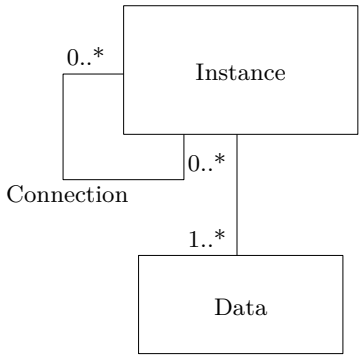


Figure 2.1: A UML diagram showing instances, data and connections. Instances can connect to 0 or more instances, but must contain at least 1 data item.

When entering information about the deployment, users are asked about the properties of the given item being entered. Instances, connections, and data all have properties which determine whether certain kinds of attacks or exposures are possible. Examples of attacks which depend on such properties are exploits which use SQL injection. These need data to be SQL data in order to work, without which the attacks are pointless.

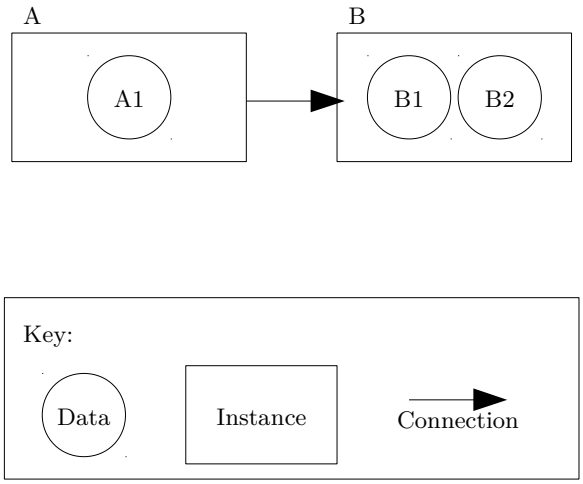


Figure 2.2: A simple cloud deployment and its required information, with instances, hosted data, and connections.

One of the main problems with user input in Cloud-COVER is in creating a balanced tool which is able to provide users with a list of system wide threats without expecting too much security expertise from them. Asking users questions with too much detail can risk making a threat modelling tool too difficult for non security experts, and also risk being too time consuming to use. Asking too little can risk only identifying obvious threats. The fact that Cloud-COVER is aimed at finding threats at a higher perspective is helpful, as it means that less information is required in order to provide an analysis.

Instances

In Cloud-COVER's model, instances are any cloud services hosting data (whether public or private cloud, or SaaS, PaaS or IaaS), and are linked by connections. Instances can host multiple data items, and can be connected to any number of other instances.

Users are asked about the properties of instances, in order to determine what possible threats exist against them. These properties can be whether they host specific types of software, whether they are updated regularly,

or any other actions which could mean that certain threats might exist.

Data

Data represents any data users host on their instances. Attackers can be a threat to different kinds of data depending on both the properties of that data, and on the security attributes of that data. Users enter this information to allow Cloud-COVER to determine the existence of those threats.

The security attributes of the data are discussed in more detail in Section 2.4, and determine *which classes of threat* are relevant to that particular data security attribute. This means that only threats which impact on a specific security attribute, such as integrity, will be retrieved at that point, and the threats related to other security attributes are retrieved depending on their order of valuation by the user. The properties of data determine whether *specific threats* from within those classes of threats exist against the data in question.

Connections

Connections represent any connection between processes from one instance to another instance. Any more than a single process is represented in a single connection, with all the permissions contained within.

Connections are obvious targets for attackers. One of the most important issues to determine is the kind of connection used. When connections are not encrypted, there are obvious security concerns to address. The connections link instances to each other, producing a directed graph of instances as nodes and connections as edges, which is used at the analysis stage.

Cloud-COVER looks at how threats to instances also need to be protected against from other instances, so the role connections play is very important within the model. To this end, users are asked to input which permissions are granted to connections to perform. Permissions are specific actions which the connecting processes are permitted to carry out at

the connected-to instance. The justifications of the analysed permissions are given in Chapter 3. Given these permissions, the analysis is able to understand what the implications are for how threats propagate across the deployment.

2.2.3 Impact

The same attack on two identical deployments could impact the two owners of those deployments in completely different ways, based on what the concerns of the deployment owners are with regards to their data. Whilst one owner may be concerned about possible data leakages violating confidentiality, another owner may only be concerned with the availability of their data and nothing else. The term 'impact' may itself be interpreted differently by people. The impacts of data breaches are often understood in economic terms, but even this can be understood differently by people. For example, this can include the lost opportunity cost of an attack, the cost to rebuild a deployment, and the possible compensation due to affected customers. It can also be difficult to assess non-economic interpretations — for example, there is no easy answer to how reputational loss could be measured.

Instead of trying to assess the impact of any attack on Cloud-COVER users in a uniform way, the users themselves are allowed to determine the potential impact to their deployment. This limits the kinds of problems which could be created by asking for information from users about things which they might interpret differently. By asking users to provide details about the importance of the security attributes of confidentiality, integrity, availability, and compliance to their data, Cloud-COVER is better able to understand how to prioritise the threats to their deployments, and make the analysis more relevant and adaptable to their individual circumstance. The method used for valuation of impact is described in more detail in Section 2.4.

2.2.4 Likelihood

Not all threats are equally likely to occur. When presented with two threats of equal ability to impact a system, the most important one to defend against is the one which is more likely to happen. Of course, it is impossible to predict for certain what the likelihood of a future event will be. However, there are ways in which we can use analysis of previously successful attacks to predict how likely certain kinds of attacks are to take place in the future. For example, methods for evaluating likelihood of threats are described in ISO 27005 [18], and in OWASP's Risk Rating Methodology [42].

Although victims of attacks are not keen to publicise their breaches, statistics on breaches are available. For example, there a number of internet security firms who investigate system breaches around the world, and compile annual reports detailing the statistics of these attacks. Trustwave [43], Cisco [44], Symantec [45] and Wensense [46] are a few of these. These statistics, which are used to help to prioritise the final threats presented to the user, are analysed in more depth in Chapter 3.

2.3 Threats and Exposures

A threat is a potential violation of security. Vulnerabilities are needed in order for attackers to exploit and thus realise those threats. The existence of many of these vulnerabilities is often due to the presence or absence of certain software or data on the system. In order to learn about the presence of these properties of the system, Cloud-COVER gives users the opportunity to input this information. Using this supplied information, Cloud-COVER is able to determine whether attackers would be able to take advantage of these vulnerabilities. When presented with a list of those threats, users can then be directed towards countermeasures to guard against them.

In Cloud-COVER, some threats may be realised through similar attacks, but are considered separately because the defences needed to

guard against them are different. Examples of this are threats from an unauthorised user executing code, or viewing data. Although both could be defended against using user permissions, the unauthorised user viewing data could also be prevented by encrypting the data. This means that it makes sense to keep these as separate threats, since the defences may vary.

2.3.1 Exposures

Not all of the threats faced by users on the cloud are from direct attacks against deployments. Other major issues can occur due to logistical reasons, such as issues concerning data location and compliance requirements. Data from industries where the confidentiality of information is considered top priority (examples include the defence and health industries), are frequently covered by laws or regulations governing where they are allowed to be stored. Cloud services may be owned by companies in one jurisdiction, the operations and staff located within another, and the services provided to users in locations anywhere in the world. Other issues come from cloud service policies, which users may not have given much thought to. For example, users may assume that data they have deleted no longer exists, but an attacker may be able to access a backed up copy based elsewhere. In this way, their data can be accessed indirectly from their deployment. The fact that these details are usually abstracted away means that they may not be clear to the user. This is the kind of threat to cloud computing deployments, different from those from more traditional (*ie* non cloud) systems, which are important to highlight in Cloud-COVER — and which are not covered by other tools. Cloud-COVER considers this class of issues, which are termed 'exposures' alongside those which are direct attacks on the deployment.

2.3.2 Propagating Threats

It is important to determine whether threats contained within the model are able to propagate to other instances. Not all threats are able to prop-

agate across connections. Those which are not able to do so, obviously do not need to be considered when analysing threat propagation. Only some types of attack remain workable when performed using trusted connections. For example, Cross Site Scripting (XSS) attacks work by attacking machines which are browsing the internet, but are not made any more effective from inside the network of a targeted instance. The nature of XSS attacks is to inject code to a website server, and wait for vulnerable machines to browse that website. They are not considered worthwhile as targeted attacks, so are not the kind of threat that would be considered to work as a propagating threat.

Each threat identified in the threat modelling stage is reviewed to see whether it makes sense for an attacker to be able to utilise it from other instances within the same deployment as well as from outside the network. If it is, then it can be added to the list of threats for which this is possible. When the results are being presented, any propagated threat values will only affect the placement of threats which are able to be propagated. Those that cannot propagate can remain in their original places.

2.3.3 Organisational Threats

The threats and exposures are split into two categories. Some of them are relevant only to particular instances, and the approach to their countermeasures needs to be implemented individually for those services. Other threats are relevant to the entire deployment or organisation as a whole, as they require a coordinated approach. For example, one of the major problems for a deployment is of not knowing where a cloud provider may base its service, which may be a problem for compliance purposes. Although the problem of data location can be handled on an instance by instance basis, the best preparation is a coordinated approach, involving an understanding of where all potential providers locate their services. Doing this instance by instance would be a much longer, and more ineffective process. This will allow the development of a policy

for how data location will be handled on all instances, and avoids the kinds of issues created by using an *ad hoc* approach for each instance. By highlighting threats in this way, Cloud-COVER can be seen to therefore place attention towards the idea of protecting deployments by looking at the trio of people, process and technology and not concentrating solely on technology issues.

Even more importantly, there are threats which are able to compromise entire deployments when responses are not considered from a deployment wide perspective, such as those from malware or poor password practice. Poor password practice (such as reusing passwords for multiple instances) can compromise an entire deployment if an attacker gains access to a cloud administrator's management account. Malware, by compromising a single instance, is able to perform a wide variety of attacks on hosted data, including retrieval of passwords. A single coordinated approach to managing these kinds of problems is absolutely necessary to maintain good security practice, and by being better advised against these threats users may be able to better protect their deployments. In contrast only using good security practice on a single instance, and not on others, can put data on that well protected instance at risk even if nothing connects to it, since the user's management portal may be compromised by the poor security practice on the other instances.

The other important difference with organisational threats is that the problem of threat propagation (covered in the next chapter) does not apply. Although some types of organisational threat do propagate, the need to consider each instantiation of such a threat as a threat to the whole deployment means that there is no need to think of them as threats to individual instances, but rather to the whole deployment. This means that they do not need to be covered by any analysis of propagation. The organisational threats therefore need to be presented to users separately from the instance threats. There is no direct way of comparing the two, and mixing the two categories when presenting results could end in some confusion for the users. By keeping them separate, users can understand that there are two distinct types of threats, with different

needs for approaching their defences.

2.3.4 Security attributes

The security attributes of confidentiality, integrity, and availability are part of a security model often referred to as the 'CIA' triad, with the acronym referring to the initial of each security attribute [47]. This model influences this work and other work in the area, by allowing threats to be categorised according to the data security attribute that they endanger. Not all kinds of attacks will impact the owners of data. For example, a data owner may be concerned with ensuring that their data remains available at all times (availability), that it does not change without the required permissions (integrity), but not with keeping their data private (confidentiality). By acknowledging that threats can be categorised in this way, data owners can better prioritise the threats to their systems.

In addition Cloud-COVER considers threats relevant to compliance, an important attribute of security for cloud systems which are not covered by the CIA trio. Compliance is an area which is relevant to cloud systems, particularly due to the remote nature of their services. Concerns about data have manifested in laws and requirements (national, international, and/or regulatory) safeguarding data. Examples of this include regulations about the location of data, or who is allowed access to data — such as the UK Data Protection Act [48], which many cloud users may be ignorant of. Other kinds of concerns are related to specific processes for data which are required to be followed. In the USA the Sarbanes-Oxley Act of 2002 includes laws about retention of evidence (and therefore data) about financial transactions [49]. The kinds of threats which apply to compliance therefore are not applicable to any of the three from the CIA triad, and are categorised using this separate attribute. In Cloud-COVER's default mode, compliance is considered to include threats to data location (keeping data within the boundaries of a jurisdiction) and threats to auditability (of knowing about important actions which are performed on data).

Each threat and exposure is only relevant to some of these security attributes. Denial of service attacks for example, are able to violate the availability of data, and also their compliance (since their auditability can be violated) and not their confidentiality, integrity. Others may be relevant to more than one, and some may even be relevant to all. Depending on the security attributes users choose to value as the highest priority, the threats relevant to those security attributes will be ordered according to the priorities given. So when a user rates availability of a particular piece of data as being the most important security attribute, the threats to availability are prioritised when they are presented with the results. This is discussed further in Section 2.4.

2.4 Impact Valuation

The need to allow users to determine the possible impact that different kinds of attacks could have on their deployment was discussed in Section 2.2.3. However even allowing users to determine this for themselves allows problems of ambiguity and misunderstandings to occur. Will Harwood's work on using alternatives to numeric valuations (discussed in Section 2.5) is one way to try to value impact. An overview of the advantages and disadvantages of absolute and relative valuations for security threats can provide some useful insight.

2.4.1 Absolute values

Advantages

- Absolute values make sense to people as they are used to rate things in everyday life.

- Absolute values are used in other threat modelling processes, and people are used to using them in this way.

Disadvantages

- Absolute values can have some ambiguity related to scale, such as whether scores double that of other ratings represent a doubling of the importance, or not.
- People are reluctant to value any kind of threats as having low values as this suggests that those threats are not being taken seriously. Anyone with a job involving security will be unwilling to suggest that they are not taking all threats seriously.
- Absolute values only allow for a maximum number of levels of importance. For example, when ranking out of a score of 10, items can only be valued on those 10 levels. For larger sets this may prove problematic, as there may be clear need to prioritise between more than 10 items.

2.4.2 Relative values

Advantages

- Relative values can be more suited to security as users do not have to specifically give low (or high) values.
- Users can give zero, one or multiple rankings to one item. This means that in cases where any two items may appear to be equally ranked by preference, users can add in an additional value between them in order to distinguish that one is preferred to the other. With numeric valuations, if two items are both given the same rankings, the only choice would be to change the value, which may move the newly valued items into the same level as other non-equal items, repeating the problem.
- Systems with a need to prioritise large numbers of items are not limited to the number of levels of the rating system, as absolute values are. Theoretically, such a method would be able to provide

as many levels of preference as can be identified from the user's valuations.

Disadvantages

- Relative values can be difficult to apply to final rankings, as they carry no inherent meaning apart from the words they have been labelled with. Users might infer their own understandings of the meanings given to them.
- Relative values can take more time to input, if users choose to provide more than one valuation for multiple items.

2.4.3 Cloud-COVER's valuations

From this comparison, it makes sense to choose the relative valuations, specifically due to the issues of users being unwilling to provide low ratings to security threats, and the issue of ratings being limited to the value of the highest score. Relative valuations do not have those problems, which are important to consider when ranking security threats. For this reason, Cloud-COVER uses relative values for its ranking system.

As all threats are linked to at least one security security attribute, the use of user priority between those security attributes is a good way of determining the orderings of those threats. By comparing the relative importance between two security attributes, a more objective valuation can be determined. Users therefore have a choice of providing a preference for the importance between the confidentiality, integrity, availability, and compliance of each data item. This means that security attributes of the same data, or those of different data, can be compared to each other to determine their relative importances for the user. This provides Cloud-COVER with a reference point from which to determine the importance of threats and exposures with regard to the user's preferences.

Given two security attributes, A and B, users are provided with the option of defining two kinds of relationships between them: A's impor-

tance is *greater than* B; or A's importance is *much greater than* B. Figure 2.3 provides an example of impact valuation between the security attributes of a single piece of data and its four security attributes, with an example of resulting threats retrieved in the order produced in Table 2.1. Figure 2.4 has an example where a user chooses to include more than one valuation for some security attributes, in order to ensure that there is less ambiguity. A1's integrity is rated as more important than A1's compliance, whereas in Figure 2.3 both security attributes were considered to be rated equally as important by the user. A user can choose not to include a security attribute if the threats to that attribute are of no concern to them. In Figure 2.5, a user chooses not to provide a valuation for *A.cn*, as they feel that the confidentiality of the data is not worthy of their attention.

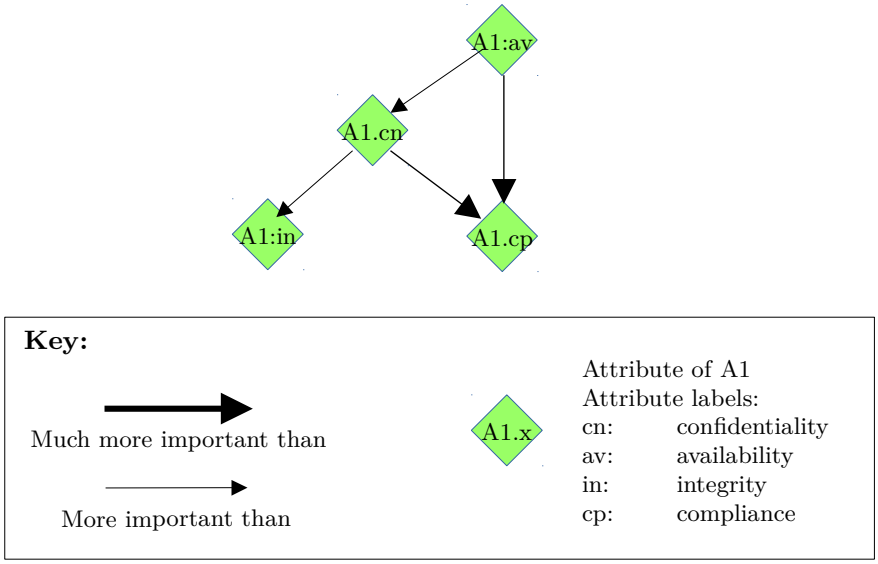


Figure 2.3: Each input datum has four security attributes which can be prioritised by the user: confidentiality, integrity, availability, and compliance. The user provides a valuation between those security attributes. Availability receives their highest rating.

Using these two relationships, and by defining at least one relationship for each data security attribute the user wishes to include, a graph is produced of the relationships and their importances relative to one

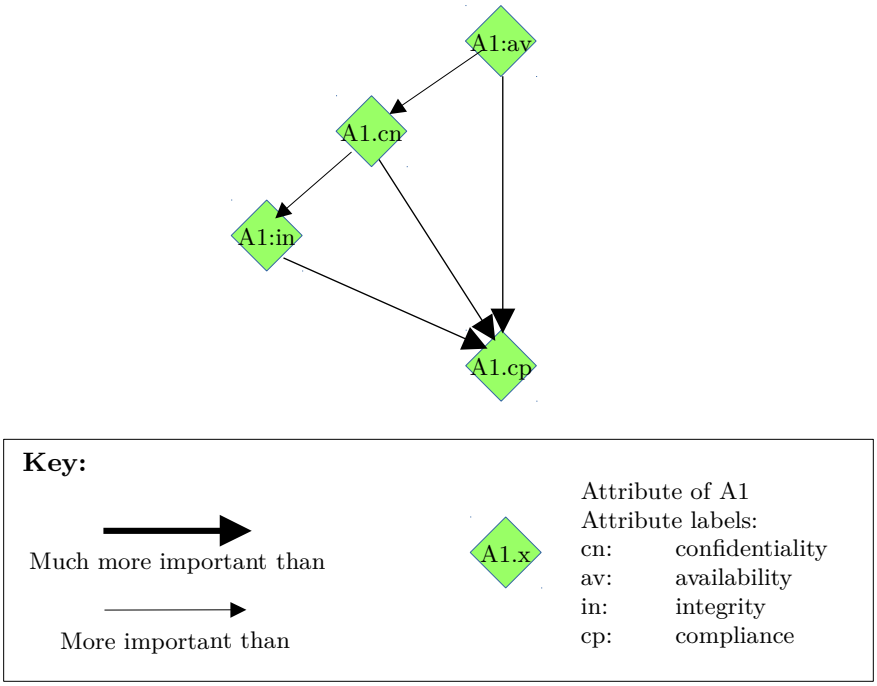


Figure 2.4: In Figure 2.3, integrity and compliance are considered equal. A user can add more valuations to reduce any such ambiguity, which they do here. $A1.in$ is now more important than $A1.cp$.

another. Subsequently the importances of all security attributes relative to each other can be inferred by their position in this graph. In Figure 2.6 a simplified example is presented, using only two security attributes (confidentiality and availability) from three data items, representing a total of six security attributes in the preference graph (integrity and compliance would also be used in a complete example). In this example, the availability attribute of datum D1 is specified as being the most important item ($D1.av$), with the confidentiality of D1 being the least important ($D1.cn$). These preferences can then be used in the analysis stage.

After the user's input, the valuations are looked at to see whether any relationships need to be re-evaluated to produce a second graph. This analysis is done in order to adjust paths which may need to be

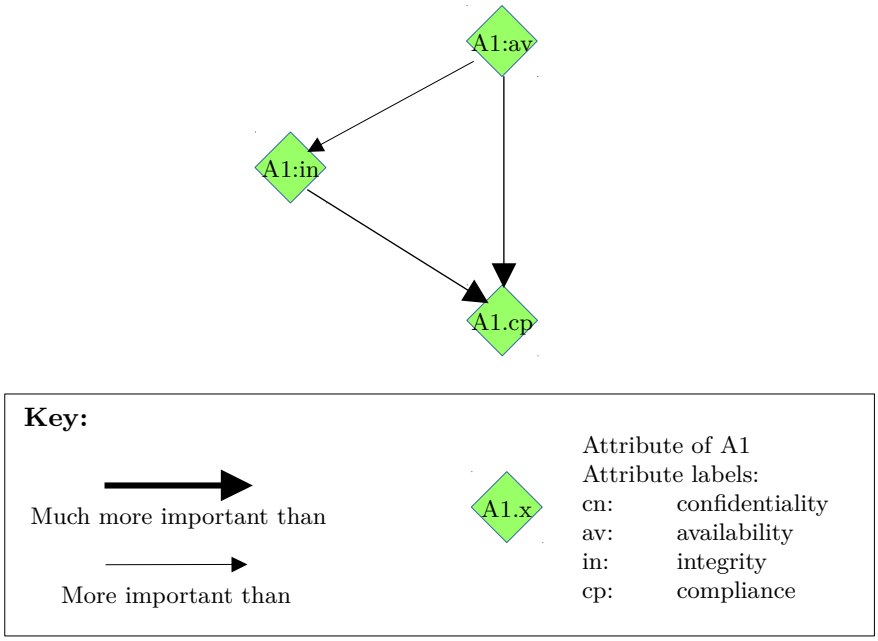


Figure 2.5: An example of a user not rating one of the data security attributes. In this example, the user does not rate confidentiality, since it is not a concern for them.

reconsidered in view of the user's preferences. In Figure 2.6, the user specifies their preferences, in which there is only one relationship of a security attribute much more important than another ($A1.cn \rightarrow D1.cn$). Whilst originally $B1.cn \rightarrow D1.cn$ is defined by the user as being a greater than relationship, its placement suggests that it needs to be evaluated again. If $B1.cn$ is more important than $A1.av$ and $A1.cn$, then it must also be much more important than $D1.cn$, in the same way that $A1.cn$ (which $B1.cn$ is rated as more important than) is. The relationship is redefined accordingly, which can be seen in its changed weighting afterwards.

Depending on their positions in the graph relative to other security attributes, it is possible for two security attributes to have an equal value to each other but this is only possible when they do not have a user defined preference between them. In Figure 2.6 the position of $B1.av$ and $B1.cn$ show that since they are both considered to have the same

Order	Data	Sec. Attribute
1	$A1$	Availability
2	$A1$	Confidentiality
3	$A1$	Compliance
=	$A1$	Integrity

Table 2.1: Threats from Figure 2.3 are retrieved in this order.

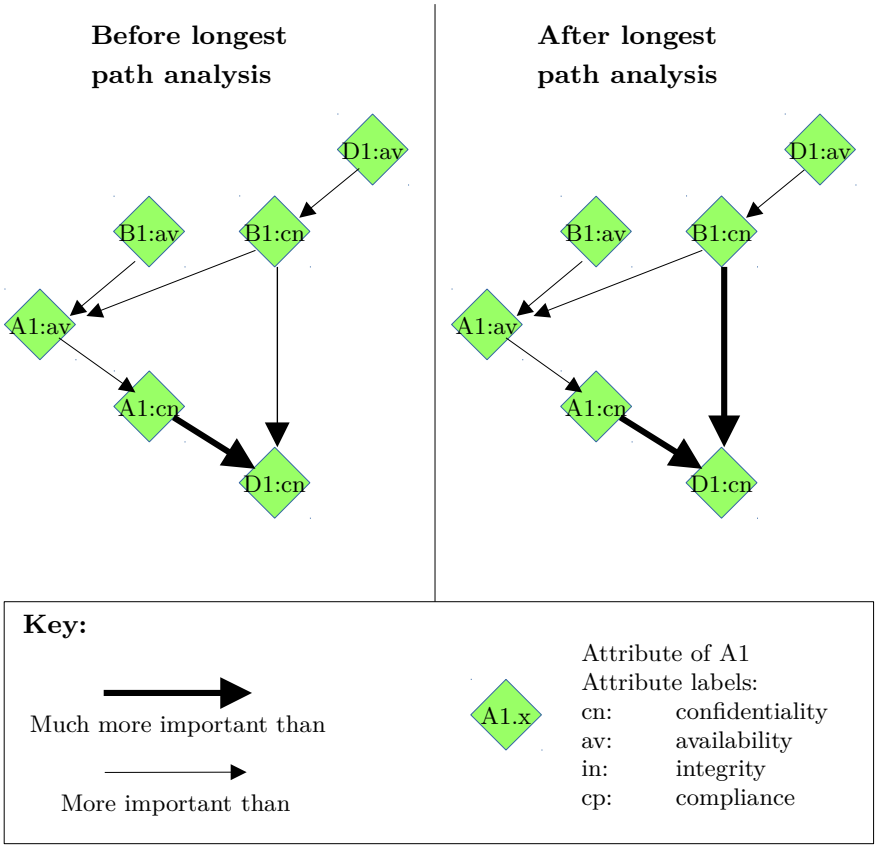


Figure 2.6: Before and after longest path analysis. Before shows a user's preferences. After shows how longest path analysis can alter the weightings of some edges (in this case, $B1.cn \rightarrow D1.cn$ changes).

relationship with $A1.av$ with no other defined preferences between them, that they are both at the same level of importance to each other. The next change to the graph is made when considering threat propagation, which is discussed in detail in Chapter 3.

2.5 Related Work

The Common Vulnerability Scoring System (CVSS) is a risk assessment methodology which uses metrics to quantify the threat posed by vulnerabilities to a system. These metrics look at the effect of the CIA triad on the severity of the impact of the vulnerability in question (in a similar way to Cloud-COVER), the environment in which it operates, and any mitigating factors [50]. Cloud-COVER considers compliance in addition to the CIA attributes, and uses user supplied comparisons about the importance of those valuations to determine the priority of threats, removing some of the ambiguity of using numeric values.

Trike is a threat modelling framework and tool which considers threats from the perspective of risk management [51]. Trike's consideration of threats involves more formality than other frameworks, asking about the interaction of actors, assets and intended actions. By relating the actors with these intended actions (for example, adding a blog entry), and considering these with regards to a number of keywords (for example, the keyword 'no' would suggest the actor is not a permitted actor), threats to the system are determined. Trike therefore involves a large amount of detail, but produces results for those patient enough to make use of it. Trike also considers threats from a different perspective to other approaches, in that all threats are categorised as being either denial of service or elevation of privilege. Cloud-COVER uses a higher perspective, and allows users to input their own attributes for modelling purposes if it fits their view of the threat modelling process.

STRIDE is a threat modelling process which works by modelling systems at low levels. Threats are determined by considering whether certain components could take advantage of a number of possibilities (Spoofing,

Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege). There are times when the threats from different classes can overlap (for example, spoofing and tampering will frequently overlap threats). STRIDE's biggest problem is the fact that it is a manual method. For this reason, Microsoft developed the SDL Threat Modelling Tool which utilised the STRIDE approach.

The Microsoft SDL Threat Modelling Tool uses a simple interface for users to enter details about their system at a low level of detail, which generates reports detailing possible threats to the system [24]. The tool is aimed primarily at developers, and identifies threats at low levels. Cloud-COVER considers organisational threats which are not covered by many threat modelling tools, and does so from a higher level perspective.

SeaSponge is a browser based threat modelling tool aimed at system administrators to help them identify threats to their systems [31]. SeaSponge works by allowing a graphical input of sections of a user's system, including data flow, and identifying the threats faced by the input system. Cloud-COVER is aimed at distributed systems, and divides threats into the categories of instance threats and organisational threats, and also considers the way in which threats propagate.

Cloud-COVER provides a coverage of higher level threats than any of the other work presented here. Additionally, all of these methods require the user to have some security knowlege in order to make use of them. By allowing more inexperienced users to perform threat modelling, Cloud-COVER providers a coverage of a greater number of users, especially those who are unlikely to have used threat modelling techniques beforehand.

Will Harwood uses alternatives to numeric values as a way of defining ratings in a trust system [52]. He uses labels assigned to those relationships (for example, the labels of trust and distrust), in order to reason about the paths linking two individuals in a web of trust. This provides a way of evaluating the relationship between any two nodes within the web of trust by referring to the provided labelled valuations. Cloud-COVER uses this approach and applies it to a ranking system in which users

provide their preferences for the priority of data security attributes.

2.6 Further Work

One of the most pressing concerns with the design of Cloud-COVER was the desire to create a more accessible level of threat modelling tool, which could be employed by a wider range of users. One of the primary ways this was achieved was by mainly asking questions about the deployment instances and their connections. There is the possibility of asking users to provide more details about their system, and this could help to aid the threat modelling process. However asking users for details like this could change the expectations of the user, and it could take longer. It would be important to understand the way in which this could impact the usability of the tool.

One other area which could be very useful to include in Cloud-COVER's model would be to consider data owners, and system users. With cloud computing allowing for organisations to share data in complex architectures, a way of modelling the interactions of users, data owners and permitted users along with connections could aid the administrators in identifying where threats in such systems could exist.

2.7 Discussion

This chapter has presented alternative ways of identifying threats and prioritising their importance for threat modelling purposes.

A novel way of rating the impact of threats to systems was introduced, in which users are able to provide relative valuations between data security attributes. This method has a number of advantages over the use of numerical valuations, not least the fact that users are not forced to provide any inherent priorities through high or low numbers. By splitting up threats into those which are relevant to four security attributes, and allowing users to prioritise those security attributes according to their

own priorities, Cloud-COVER is better able to suggest the most pressing threats to the deployments under consideration.

In addition, the viewpoint of the model is different to most other threat modelling processes. Cloud-COVER's model does not look at low level details such as threads or processes and instead considers instances, data, connections and their properties. By considering the perspective of systems Cloud-COVER is better able to identify threats which system owners and administrators might be responsible for.

Important distinctions are also made between different kinds of threats, so that users can be provided with better knowledge of the types of threats which exist, and the need for different kinds of responses to them. An important example of this is the differentiation of instance threats and organisational threats. The fact that instance threats may only exist on a single instance at any one time, and that organisational threats always exist against all instances in a deployment (and that they require a coordinated approach to defend against), is important for users to know about.

3 Threat Propagation and Ordering

3.1 Introduction

Any analysis of threats to a cloud computing deployment cannot be considered complete without an understanding of the way in which threats propagate throughout the system. Deployments with more than a single instance contain multiple attack points, making it important to think about how disparities in the levels of security between instances could prove detrimental to overall security. Attackers who find their targeted instance too hard to breach may find that they can reach their intended target by going through other less well protected instances which connect to (and has permissions to make changes to) their original target.

This issue is what is referred to here as threat propagation. Threat propagation depends on the permissions granted to the connections between the instances. Other threats could exist when attackers breach a deployment through poorly protected instances, and only then find that they are able to reach even better protected instances, subsequently using those connections to perform an even more destructive attack than they had originally planned. It is crucial therefore to understand the impact of the ways in which instances connect to each other, and what this means for threats and defending against them.

In a simplification of reality, if a system's security can only be considered to be as strong as its weakest link, then all instances within a cloud deployment would need to be as well defended as each other. This however does not consider the more complicated nature of the cloud, as users may only have limited control over some of their instances (such as

Software-as-a-Service), they may not allow those connections to have any meaningful permissions to violate security, or may not even consider certain instances to host anything valuable (and therefore not worth wasting resources to protect). Each of these are issues to be considered for any kind of threat propagation analysis. This chapter looks at how Cloud-COVER's model treats threat propagation, by looking at the connections and their permissions, and determining whether they would be able to help an attacker violate data security attributes over the connections.

This chapter is structured in the following way: threat propagation is discussed in Section 3.2, with user valuations discussed in Section 3.3, and the impact of connection permissions covered in Section 3.4. Data operations and their relevance to permissions are covered in Section 3.5, and the actual method of valuation propagation is presented in Section 3.6. The ordering of final results in presented in Section 3.7, related work is in Section 3.8, further work is in Section 3.9 and finally a discussion in Section 3.10.

3.2 Threat Propagation

It is important to define and justify what threat propagation means within the context of Cloud-COVER, before covering how threat propagation is modelled within the tool.

In order to determine the level of impact different threats could have to a deployment, Cloud-COVER uses the security attribute valuations from users to find the threats relevant to those prioritised security attributes, and to order the threats accordingly. The fact that attackers are known to use less well protected instances (beachheads) to breach deployments means that it is vital to identify any potential instances which could play such a role. To protect against them requires an understanding of the way that threats propagate from one part of the system to another.

It is tempting to think that threats propagate in a way such that they appear to be transitive. Let us take an example scenario in which 3 instances, A, B and C, are connected. Threats to C also need to be

protected against from instance B. In addition, threats to B also need to be protected against from instance A. Should it not therefore mean that threats to C need to be protected against from instance A? Not so long ago, when security practice did not have access to the number of tools and processes available today, this may have been an acceptable way to approach the situation. However times have changed, and attackers at present need to do a lot more work in order to move through connections (assuming that good security practice is being used). Also given that everyone has limited resources, it is unreasonable to aim to protect all instances at an equal level.

Use of good security practice, such as intrusion detection systems, means that attacks can often be identified before they are able to spread from their initial foothold. This helps administrators take action to lock down those instances before attackers have the opportunity to continue their actions, and prevent them from moving at will through deployments. In Cloud-COVER, one of the important ways in which security controls are represented are the connection permissions, and they allow the modelling of the threats of attackers when individual instances of a deployment are breached. They also demonstrate how and why some threats are unable to simply propagate across all connections that they come across. They will only be able to propagate across some connections, depending on what security attribute they are a threat to, and what permissions the connections have. They therefore require some analysis in order to understand where this could happen.

If the assumption about threats simply propagating across instances C to B to A held up then the simple way to protect against threats to those instances would be to value all threats to a certain security attribute (or even all security attributes) as being at the same level across all connected instances. Actually, there is some value in this approach, and it is already included in Cloud-COVER when considering organisational threats. In the previous chapter, we described how threats were split up into those threats to instances as well as threats to deployments (organisational threats). The fact is that organisational threats which

require a coordinated approach are those which are given the same value across all systems. For the threats which require instance by instance approach, it is unreasonable to expect users to prioritise all instances the same, and makes much more sense to identify specifically those instances with data security attributes of higher value, and the instances from which that data might be attacked. This allows users to dedicate their sparse resources more effectively, instead of looking to protect all instances at the highest possible level.

A more effective way to protect assets would be to protect data on an instance, and then protect that same data from being attacked *from any connections representing a threat*, instead of protecting the data and all connecting instances with the same priority. The actual way in which Cloud-COVER treats threats is seen in Figure 3.1, with Table 3.1 providing an understanding of how the threats can be prioritised. In this figure, C1's availability is valued as the most important, so these threats are considered first in priority. Then, equally as important, are threats to the same data, coming *from B to C*. Instead of protecting the whole instance of B from that threat, which seems to be a waste of resources if it the hosted data is not valued as highly, only the threats to C1's availability are protected against. Then, next most important are threats to confidentiality to C1 on C, and equally as important, from B to C1. The figure and table show how breaking down threat propagation in this way should lead to a better understanding of how resources should be allocated when assessing security.

One of Cloud-COVER's important roles therefore is to provide advice to prevent an attacker from being able to mount a successful beachhead attack. Cloud-COVER does this by identifying important connections that attackers could use for breaches, and advising users to protect such connections from those particular threats. This makes a much more effective security approach than other approaches which disregard the crucial role of beachheads in distributed systems.

Finally, it is important to provide users with a tool which does not just present them with a list of threats but which allows them to understand

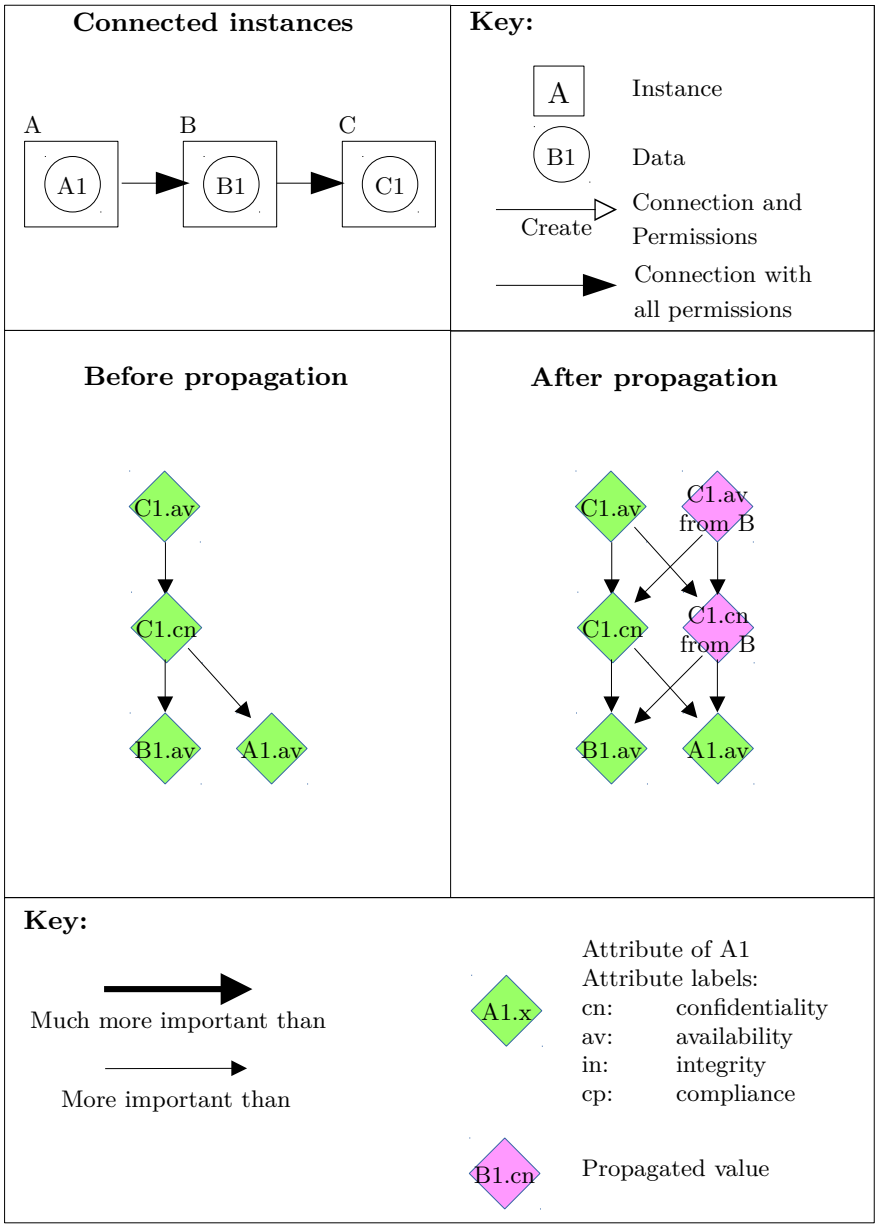


Figure 3.1: A simple example of threats propagating. There is no transitive propagation. Threats to data on C need to be protected against from instance C, and from threats on instance B.

Priority	Threat	Instance	Data
1	Threats to availability	C	C1
=	Threats to availability	C from B	C1
3	Threats to confidentiality	C	C1
=	Threats to confidentiality	C from B	C1
5	Threats to availability	B	B1
=	Threats to availability	A	A1

Table 3.1: The values propagate from instance to instance, and require data to be protected against from incoming connections. The priority of the threats is based on the values in Figure 3.1.

the implications of the choices they make. If the only thing which changed was the presence or absence of threats, and not the priority they were presented in, then users would be much less likely to value making good security choices for important issues like the connections coming in and out of their instances. By allowing the configuration of the deployment to demonstrate the implications for systems security, users can learn the importance of good security practice.

3.3 Valuations

The user's security attribute valuations represent the priority that they attach to the security of the associated data. As other instances are likely to connect to and possibly have permissions to perform operations on that data, one cannot only consider threats from the instance on which that data resides. As the valuations provided are used to prioritise threats in the presented threat rankings, they can be used as the way in which threat propagation is analysed as well.

When analysing the cloud model input by the user, the analysis will need to take place in two stages. Firstly, the analysis looks at all instances. Since threats can only propagate when connections between instances exist, the analysis then goes through each of the outgoing connections. Then the analysis looks at the two instances in question (the connect-

ing instance, and the connected-to instance), and looks at the security attribute valuations for data on each instance, and checks to see whether the correct conditions for propagations are satisfied. These conditions are described in Section 3.6.

However, not all connections are equal, and sometimes certain connections can mean that those values do not propagate. Network connections can be configured with permissions, which can be one way of creating more secure links. The granted permissions can limit users and processes at both ends of the connection, and therefore affect the way in which attackers might make use of those connections. This will mean that although some threats could propagate to all instances over their connections, others might not be able to propagate at all — this will all depend on the properties of the instances and connections.

3.4 Connection Permissions

Connection permissions are specific actions that any processes within an instance are permitted to perform on other connected-to instances. Depending on the permissions granted to the connections, only certain kinds of security attributes can be violated. For example, confidentiality cannot be broken by the destruction of data, since the destruction will mean that it is impossible for anyone to have violated confidentiality by seeing the data in question. For the other permissions, it is easy enough to think of examples of ways that confidentiality can be violated. When moving data, or requesting the forwarding of data, it is possible for the data to be passed to a user or instance who does not have the right to view that data. The creation of data may require the need to simultaneously maintain its confidentiality (*eg* with access control). The way other security attributes are all considered individually with their own combinations with security permissions is discussed in Section 3.5.

Using this analysis, the only time security attribute valuations are propagated is when those permissions are granted on the connections under analysis. It is therefore vital to consider how within a cloud

deployment permissions might affect security, and for Cloud-COVER to be able to provide such information to its users. By understanding the implications of granted permissions to each of the data security attributes, the analysis can produce a set of results which mean that only those threats which can pose a threats to a deployment are contained in the threat results presented to the users.

3.5 Operations

Within the model, an important question is how to represent the connection permissions. One example of permissions is that of regular access control, in which the controlled permissions are whether to allow users the ability to command read, write, or execute operations on a file. Although this is one way in which it could be considered, there is a difficulty involved in thinking about the permission to execute (since it could include further operations to read and write from within the file being executed). Therefore Cloud-COVER instead uses an alternative method. However, other perspectives (such as ones where alternative permissions are used) may be explored by using the extensibility feature on Cloud-COVER's inputs. These are discussed further in Chapter 4.

The fundamental operations present in all computing systems are the creation, destruction, comparison, and movement of data. Although the modification of data can also be considered as another operation, it can be done in two different ways. Modification can be a combination of the destruction of data followed by the creation of new data, or it can be when the old data is retained but moved to a new location. This means that modification does not strictly need to be considered. The destruction of data is an obvious possibility of a threat to data. Data movement is also a threat when it is moved without permission of the owner. Data creation can create problems if not created with a security process in mind (such as access control). Another important operation, the comparison of data, can be used in security for evaluating between two values, but is not an operation which is capable of creating new security threats. As a result,

it is not included among the permissions used in the model. The other three operations (creation, destruction, and movement) are all capable of creating new threats in a system when not managed correctly.

Data operations can be linked with the security attributes already contained within the model. For example, the destruction of data is of clear relevance to the compliance attribute, since the command could be used to destroy auditability data. The fact that these links could provide another level of depth to the analysis (covered in more depth below), meant that the identified operations were chosen to be the permissions included in the model.

The permissions granted to these types of operations can then be used to analyse the permissions given to connections. From this point on, we shall refer to the granting of these operations as permissions, unless specifically discussing issues related to the individual operations themselves. Therefore, permissions can be considered to be conditions which describe whether specific types of operations are allowed to be performed. The fact that each of the permissions may not allow security attributes to be violated requires a justification of how these conditions will be included within the model's analysis tool. In Table 3.2, the permissions which allow for certain security attributes to be violated over connections is presented. The permissions themselves are also discussed further.

The permissions and security attributes used in Cloud-COVER are those which are found to present a useful balance between detail and ease of input. It is easy enough to ask users to input more information about their permissions, and possible to use those for more detailed analysis, but there would be a price to pay for doing this. Resulting problems would include usability issues for users (most importantly, the amount of time to enter such information), especially those with less security experience. There is an argument to be made for including more detail, and it is easy enough to think of more permissions or security attributes (CIA is only one of a number of security models). If users are concerned enough about including other permissions this is an option

that is available to them through Cloud-COVER's extensibility feature (explored in Chapter 4).

Sec. Attribute	Threat	Reason
Data Creation		
Confidentiality	Yes	Newly created confidential data requires immediate access control
Integrity	Yes	Newly created data may require integrity check at later date
Availability	Yes	Newly created data can impact availability
Compliance	Yes	Newly created data may be subject to compliance procedures (such as auditability)
Data Destruction		
Confidentiality	No	Destroyed data cannot be accessed to violate its confidentiality
Integrity	Yes	Destroyed data cannot be verified for integrity
Availability	Yes	Destroyed data cannot be accessed
Compliance	Yes	Destroyed data could be compliance data (such as auditability data)
Data Movement		
Confidentiality	Yes	Movement of data may allow non-permitted users to access data
Integrity	No	Moving data cannot affect integrity (as long as original data is retained)
Availability	No	Moving data cannot affect availability (as long as original data is retained)
Compliance	Yes	Movement of data may be subject to compliance requirements

Table 3.2: Table describing combination of permissions, security attributes and threats. Threat column asks whether security attribute valuation propagates on a connection only featuring that permission. For example, confidentiality is not affected by data destruction, so the answer is no.

3.5.1 Creation

The creation of data across a network can be a problem in cases where the data requires specific properties upon creation. Newly created data frequently requires permissions to be created or integrity checks. Depending on the kind of instruction received through the connection, any of the data security attributes could be violated by attackers. Newly created cata can also impact availability.

Newly created data is also required in many organisations to undergo a compliance process where it needs to auditable. This would mean that it would need to be created in such a way that means that the system can keep track of what is performed on it. Creating new data, even on a secure instance, therefore means that the machine where these instructions can be sent from needs to be kept secure.

3.5.2 Destruction

Data destruction is a clear way to create problems in a system. Availability is the obvious security attribute which will be affected by data destruction, given that it depends upon the data in question existing.

Destruction can also be a problem for compliance purposes, since auditability requires that the metadata holding information on the auditing of data needs to be retained. Destruction of auditing data could therefore be violation of this requirement.

3.5.3 Movement

The movement of data on an individual machine is a copy operation from one data location to another data location on the same machine. In distributed systems, the move instruction needs to be split into two different instructions. Firstly, a machine can move its own data to another location on itself. Secondly, a machine can send a command to another machine to forward its data to another second machine. Therefore, someone who has tried and failed to capture data present on a target machine may

accomplish the same goal if they learn that another less secure instance has the right to send a forward command, and subsequently use this command to forward all the information from their original target to themselves.

There is an important note to make here. A read command from instance A to instance B can be considered to be the same thing as a forward command, and both are represented as movement. This is because if instance A is breached, and this command is sent to instance B, it violates exactly the same security attributes as if the data had been forwarded to another third location owned by the attacker.

3.6 Valuation Propagation

There are three specific conditions which must be fulfilled in order for a security attribute's value to be propagated.

1. The two instances hosting the data security attributes in question must be connected.

2. The security attribute on the connected-to instance must be rated as more important by the user.

3. The permission for the security attribute in question to propagate over that connection must satisfy those described in Table 3.2.

This is done as an iterative process, by going through each instance and value of the data security attributes on each. After these conditions have been satisfied, the value of the more highly rated security attribute is passed over. These propagations continue until each of the connections has been exhausted, ensuring that no more propagations can occur.

An example of threat propagation not occurring due to violation of the first and second conditions is in Figure 3.2. Figure 3.3 is an example showing some threat propagation happening, and other propagations not happening due to non-fulfilment of the three conditions. An example

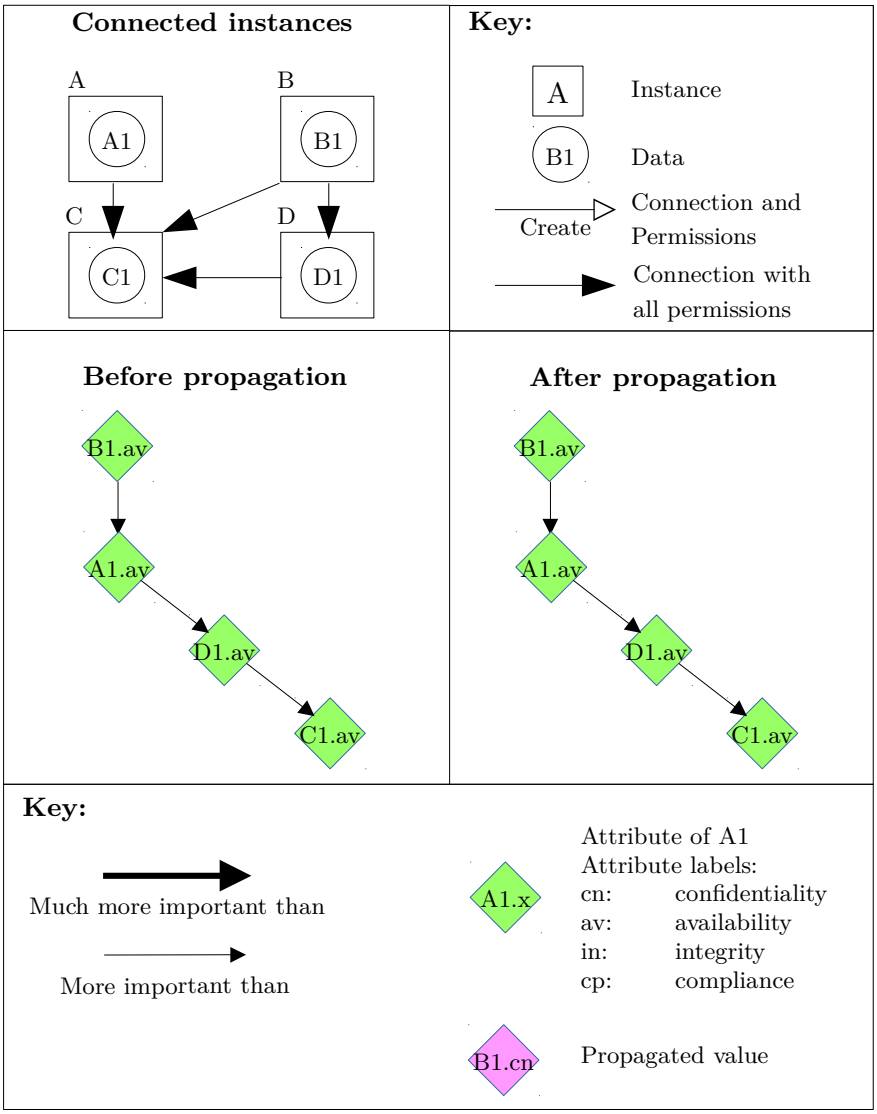


Figure 3.2: Values only propagate backwards on connections. Here, none of the values propagate as the connections mean that either their values are not higher than data on instances connecting to theirs (the second of the three conditions explained in Section 3.6), or their instances do not connect to instances whose data they are valued higher than (the first condition). For example, instances C and D are both connected to by instance B, but the values of data on C and D are lower than B, so no propagations can occur.

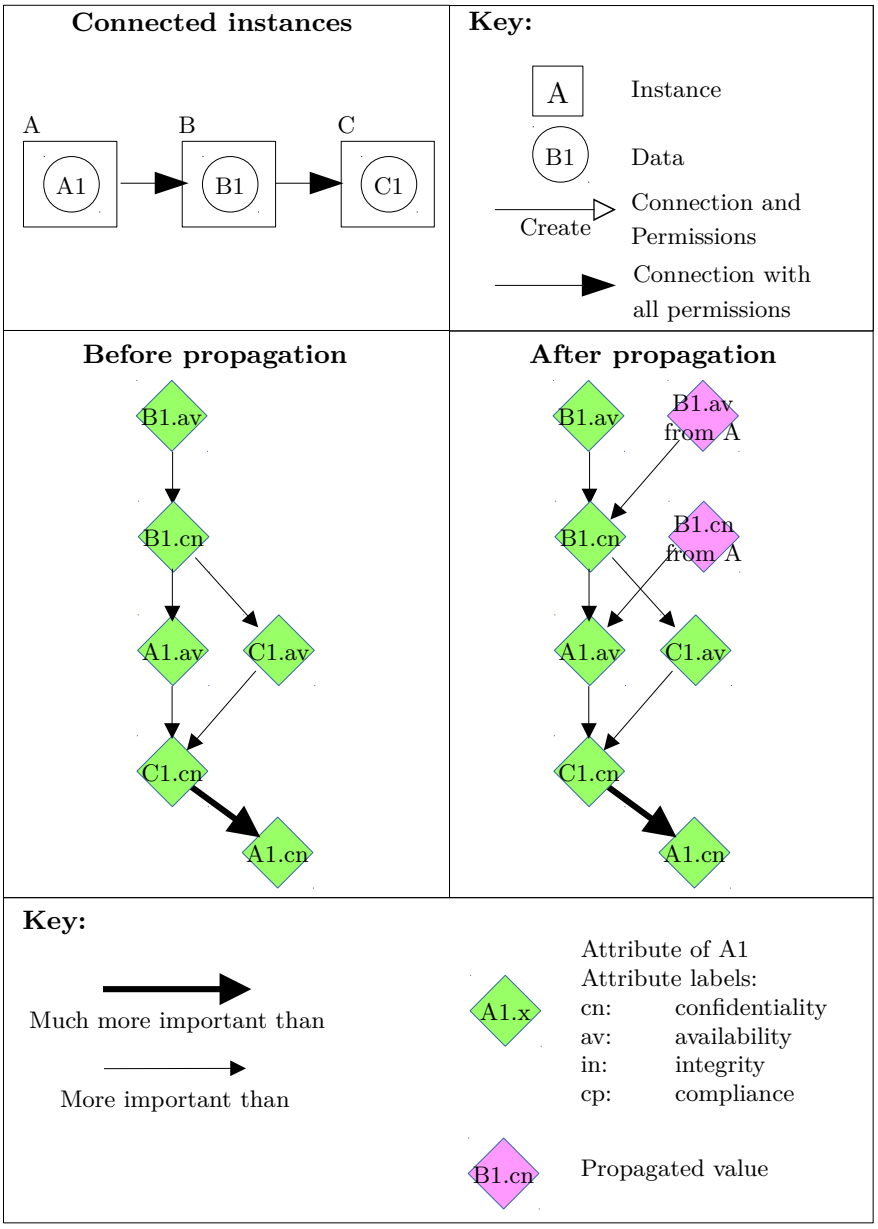


Figure 3.3: B1's values can only propagate to A1, so there is also a threat to *B1.av* from A. This value is represented next to *B1.av*, as *B1.av* from A. C1's values, although higher than A1's values, cannot propagate to instance A, since instance A connects only to B (the first of the three conditions for propagation explained in Section 3.6).

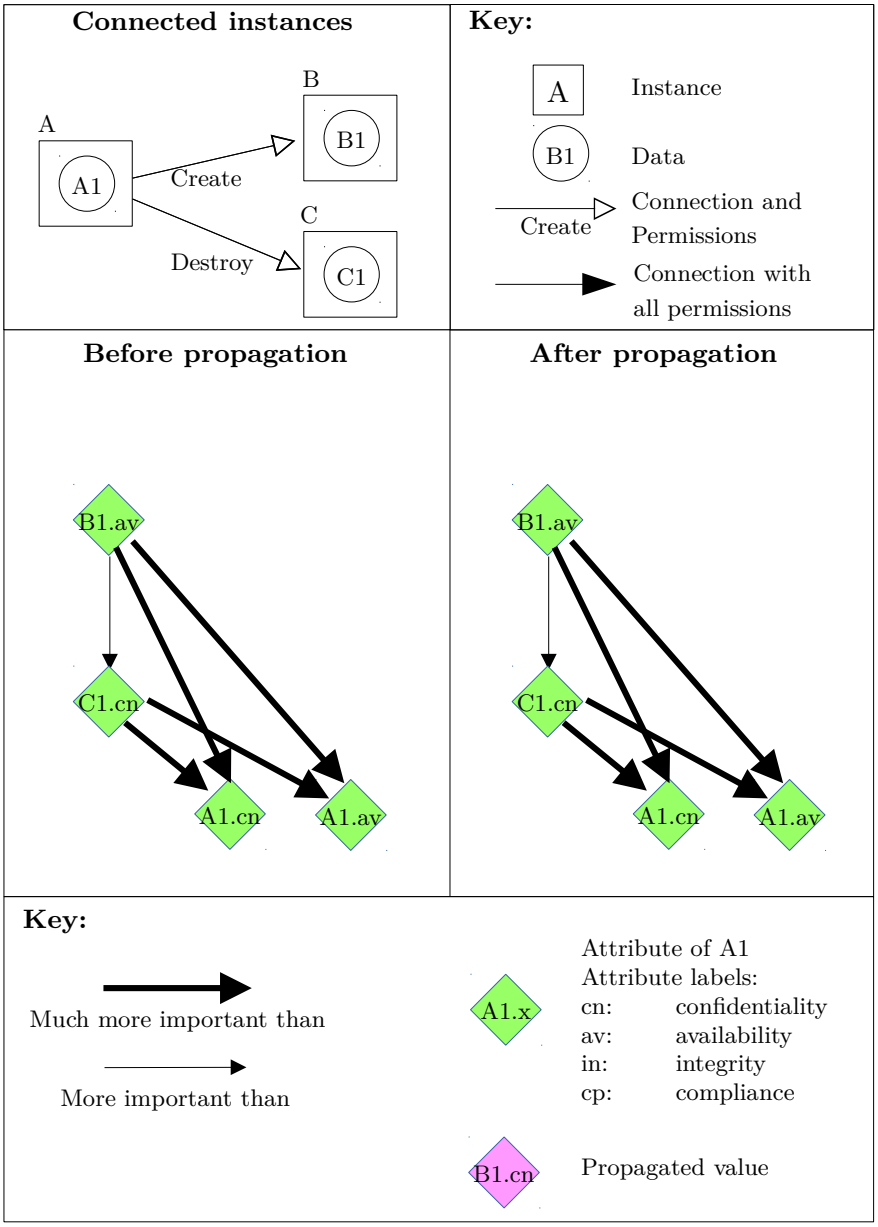


Figure 3.4: None of the values propagate, because the connection permissions mean those security attributes cannot be affected. This is the third of the three conditions explained in Section 3.6.

of threat propagation not happening due to the third condition not being satisfied is given in Figure 3.4. In this example, there are two security attribute types, availability and confidentiality. The availability attributes are linked by the connection between instance A and B, with the connection only granted the ability to create data. Although $B1.av$ has a larger value than $A1.av$, the connection means that availability attributes cannot propagate their value. To do so, the connection would require the permission to destroy. Likewise, $C1.cn$ cannot propagate its value to instance A, because confidentiality requires the connection to create or move data, which it does not have.

An example of these values being propagated can be seen in Figure 3.5 and Figure 3.6. Assuming all permissions are granted on the connections, we can see that values can only propagate from C and B to A, or from A to D. Values from B and C cannot propagate to each other, or to D. In Figure 3.5 the availability attribute of C1 is higher than that of A1, so this value propagates to A, as $C1.avfromA$. $A1.av$ position remains the same. C1's confidentiality however, is lower than $A1.cn$, so no propagation needs to take place. In Figure 3.6, $A1.cn$ is checked against another of the connected-to security attribute valuations, $B1.cn$. Since B is connected to from A, and B1's valuation is higher, the value propagates over. Since D's security attributes are the highest valued, its values could only propagate to incoming instances. Since there are none, its values do not propagate to any others.

3.7 Ordering

After the propagation analysis, the user's threat prioritisation does not need any further analysis. Instead, the remaining work is the ordering of the threats relevant to each of those security attributes.

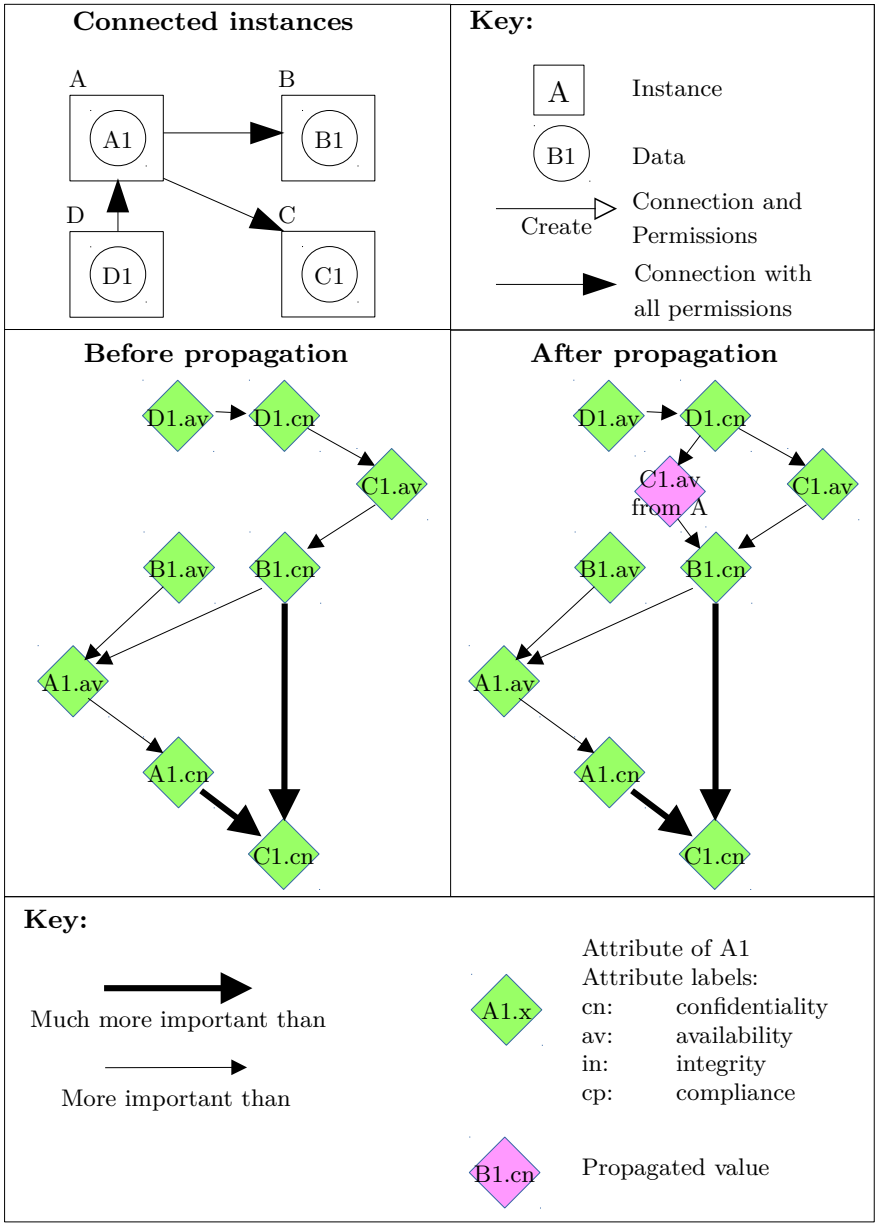


Figure 3.5: Part 1 of an example step by step propagation. Here threats to C1.*av* propagate to instance A.

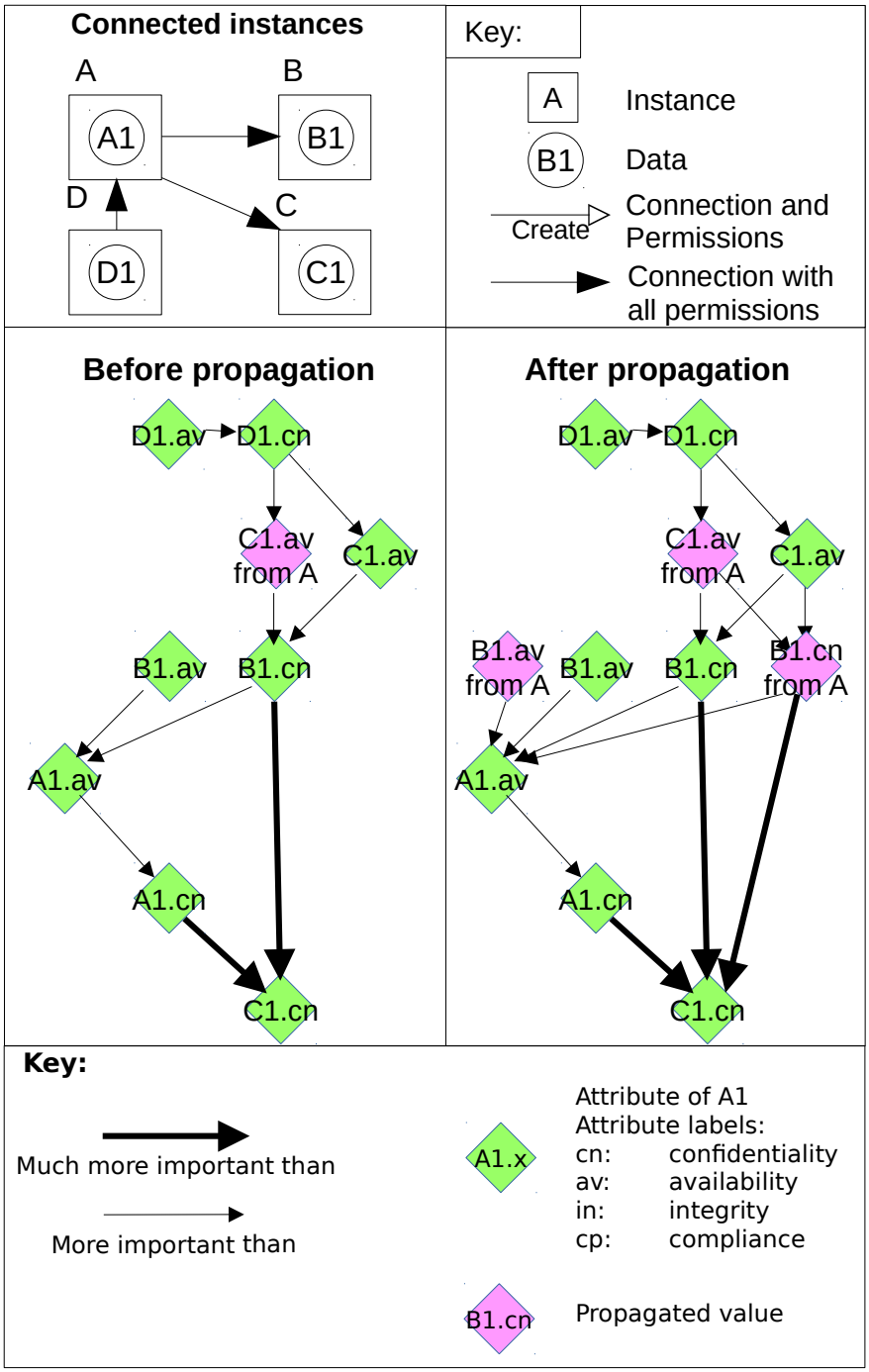


Figure 3.6: Part 2 of an example step by step propagation. Here threats to *B*1.*cn* and *B*1.*av* propagate to instance A.

Rank	Data	Sec. Attribute	Threat	Probability	Instance
1	D1	Availability	Denial of service	High	D
2	D1	Availability	CMS content deletion	Medium	D
3	D1	Availability	Virtualisation environment breach	Low	D
4	D1	Confidentiality	SQL Injection data view	High	D
5	D1	Confidentiality	Unautharised user data view	Medium	D
6	D1	Confidentiality	Virtualisation environment breach	Low	D
7	C1	Availability	SQL Injection deletion	High	C
=	C1	Availability	SQL Injection deletion	High	C (from A)
...	...	...	...	...	...
22	C1	Confidentiality	SQL Injection data view	High	C
23	C1	Confidentiality	Unautharised user data view	Medium	C
24	C1	Confidentiality	Virtualisation environment breach	Low	C

Table 3.3: Initial ordering using security attribute preferences based on Figure 3.6

Rank	Data	Sec. Attribute	Threat	Probability	Instance
1	D_1	Availability	Denial of service	High	D
2	D_1	Availability	CMS content deletion	Medium	D
=	D_1	Confidentiality	SQL Injection data view	High	D
4	D_1	Availability	Virtualisation environment breach	Low	D
=	D_1	Confidentiality	Unautharised user data view	Medium	D
=	C_1	Availability	SQL Injection deletion	High	C
=	C_1	Availability	SQL Injection deletion	High	C (from A)
8	D_1	Confidentiality	Virtualisation environment breach	Low	D
...	...	...	...	...	...
22	C_1	Confidentiality	SQL Injection data view	High	C
23	C_1	Confidentiality	Unautharised user data view	Medium	C
24	C_1	Confidentiality	Virtualisation environment breach	Low	C

Table 3.4: Subsequent ordering considering relationships and probabilities, using attribute preferences based on Figure 3.6

3.7.1 Probability

There is no way to know for certain the techniques an attacker may use to attempt a breach. Two attackers may use completely different techniques to attack identical deployments, using their strengths as attackers to choose their tactics, rather than the weaknesses of their targets. Deployments may also have multiple points to attack, and it will not be possible to anticipate which one will have the higher likelihood of being targeted. There is even the possibility that a coordinated attack could target several points at once. Exploits of previously unknown vulnerabilities (often called 'zero-day attacks') can also not be anticipated, and can cause massive disruption.[1] The fact that some vulnerabilities are immediately disclosed, whilst some are kept secret, only adds to the problems facing system owners.

Victims of data breaches are not particularly keen on publicising attacks. This may be due to damage to their reputation, poor security practice, or any number of other reasons. This leaves knowledge of these breaches hard to come by. The problem is that ideally, data about these breaches are exactly what is needed in order to understand how to better defend against them. Knowledge of previously successful attacks can provide help in determining the most vulnerable and targeted parts of systems.

Despite the lack of data on this topic, there are some resources which are able to help shed some light on the topic of successful attacks. The statistics from these resources are useful to understand the general pattern attacks have taken in the past few years. For example, the internet security firm Trustwave, who investigate system breaches around the world, compile annual reports detailing the statistics of these attacks [43]. Some statistics from these reports can be seen in Figure 3.7.

These statistics show a general trend in which a large number of attacks

[1] Although the very first victims of zero-day-attacks have no chance of protecting themselves, most people can still attempt to defend their systems from these attacks in the period following their discovery (technically they may no longer be zero-day-attacks at this point, but the threat will still exist) by using good security practice and updating vulnerable software.

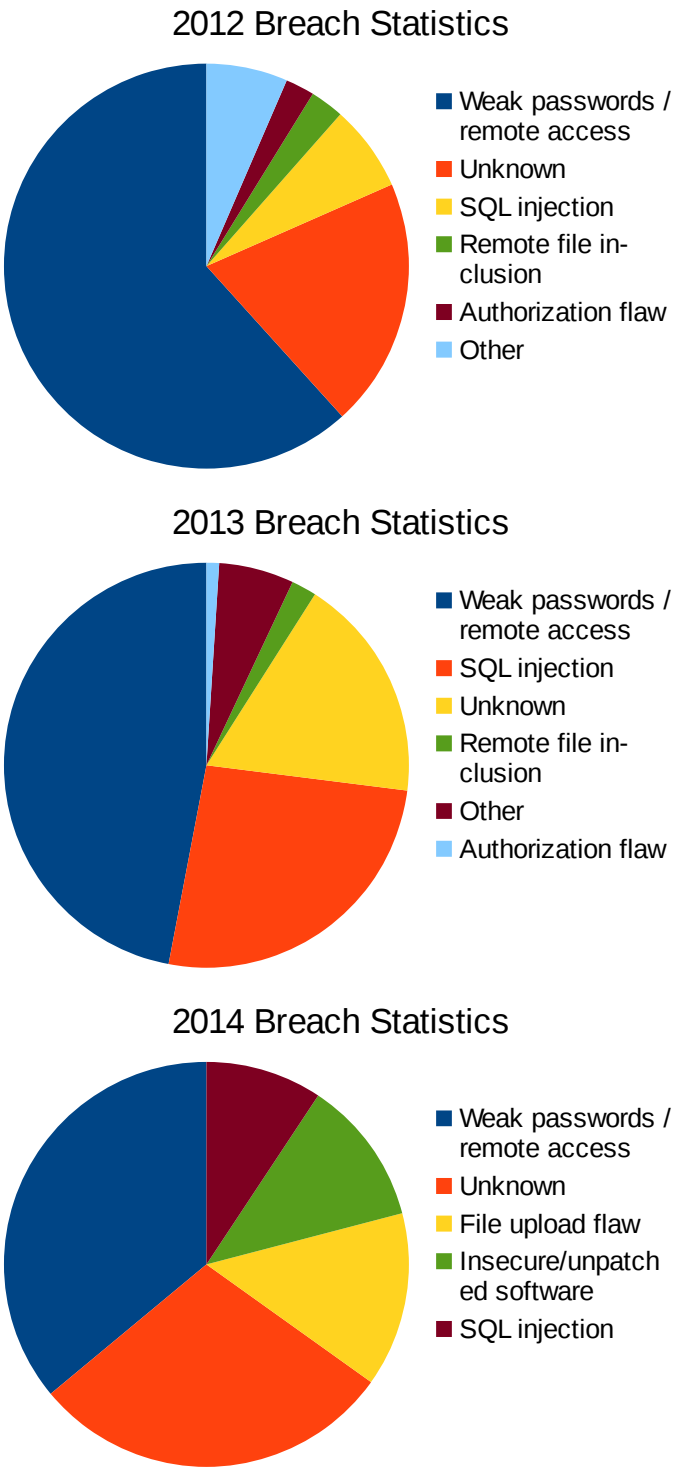


Figure 3.7: Trustwave statistics on security breaches for 2012 - 2014.

take similar approaches. Over several years, the same attacks tend to feature prominently (in particular poor passwords, SQL injection, and vulnerable off the shelf software), with other attacks also regularly showing up. Despite the huge number of vulnerabilities which exist, the fact that these attack techniques stay in use demonstrates their usefulness and prevalence. This suggests that attackers tend to concentrate on proven techniques instead of looking for new methods of attacks. Although statistics of the past do not assure us of the pattern of future attacks, they suggest that the likelihood is that most attackers will try to make use of a limited number of well proven attacking techniques that they are familiar with, similar to those used in the recent past.

One interesting thing to note from the statistics is the number of attacks classified as 'unknown'. This is worrying, but also suggests a need for a tool like Cloud-COVER. By advising its users on security defences, which can include intrusion detection mechanisms, and logging and auditing systems, users can be better prepared to investigate security breaches. Although it could be possible that the attacks could not be identified using those precautions, it would seem likely that they could help to identify at least some of the unknown attack techniques.

Although some of the threats in the cloud come from direct attacks on deployments, there are also many that do not. As discussed in Chapter 2, some problems come from exposures which also need to be protected against. However, finding statistics about the prevalence of these issues is much more difficult. Instead of relying on statistics to understand the likelihood of these threats, the analysis of cloud security experts has been used, such as literature from ENISA [53].

Using these resources, the threat categories are split into three classes, those with high, medium and low probabilities of occurring. Those threats which are repeatedly seen most often are categorised as high, with the less popular ones categorised as medium probability. Those categorised as low probability are rarely seen but are still known to exist. For this reason, very few threats are categorised as low probability. These probabilities are then used, along with the valuations provided by the

user, to order the results. The probabilities are ultimately decided upon not by any reference to a single source but on a combination of expertise derived from the kinds of sources mentioned. Although this approach is open to criticism, there is no obvious solution which can take away from the fact that no resource has enough data from which to answer such questions with total certainty.

One criticism which could be made of using probabilities in this way is to differentiate between opportunistic and motivated attackers. Whilst opportunistic attackers will simply use any openings they find through analysis of their target, motivated attackers will use a variety of techniques following an assessment of their target. By concentrating on informing Cloud-COVER users of the areas attackers are most likely to target, those techniques which happen to be most likely to be used by any attackers can be identified and defended against. Although attackers will carry on only as long as their motivation lasts, defenders need to allocate enough resources to ensure their defences last at least that long. The presentation of the threats to the deployment in the order of a user's priority can be a good way for them to determine the best way to balance their resources for this purpose. By considering their own threshold for what they would be prepared to accept from an attack, the results can help them better decide whether they need to devote more resources to security.

3.7.2 Final Ordering

After the valuation propagation has been completed, the final stage is the ordering of the threats. The likelihood of the various threats and exposures are not all equal. Even if an availability security attribute is valued as most important, the different kinds of threats to availability are not all worthy of being given the same amount of attention. Clearly, the most important ones to pay attention to are those which are more likely than the others. The classified probabilities are used as a basis from which to order the threats for the final results.

Given three threats with the probability classifications of high, medium and low respectively, there is an obvious need to pay attention to the threats in order of their likelihood. The main question to answer is how to order threats when many security attributes are valued closely in priority by the user.

Cloud-COVER's approach

Cloud-COVER includes two different preference valuations for users to make between security attributes. This allows the ordering of the results to be made according to different preferences. It is important to note that the way in which the results are affected by the valuations could be changed if it was found to be much more useful. For Cloud-COVER, the threat results are ordered in the following way.

Separating threats only according to user security attribute valuation is not ideal. This is precisely because of what the probabilities suggest. Given two security attribute valuations for a datum A, confidentiality and availability, if confidentiality is valued to be higher than availability, there is a difficulty in suggesting that all of the confidentiality threats should be valued higher than the availability threats. Since there are likely to be some threats to confidentiality which are low probability, and some threats to availability which are high probability, there needs to be a way to re-order the threats to take this into account.

The approach taken here is to order the values differently based on their relationships to other security attributes. The threats from a security attribute much more important than another security attribute are all ranked above the lesser valued security attribute. The threats between two security attributes, A and B, where A is only more important than B, are a bit more complicated. A's high probability threats are all ranked above B's threats. But A's medium ranked threats are ranked at the same level as B's high probability threats. Instead of ranking them differently, it makes sense to present them at the same level of priority and let the user prioritise between them (and present them with other useful

information at the same time, such as the probability of each of those threats, something useful the user might wish to consider when deciding between them). If the user had wanted all A's threats to be ranked above B, they could have chosen to rank A as much more important than B. The rankings continue to be ranked in the same way with the lower ranked threats, through the remaining threats and security attributes.

Examples of these can be seen in Table 3.3 and Table 3.4, which show an example before and after ranking of results. Q's confidentiality security attribute, which is ranked as much less important than the others, are all ranked below the other threats. The other threats are all mixed in with each other, but ultimately depend on the preferences between them, and their probabilities. Also Table 3.5 is an example of organisational threats, which are presented to the user separately from the instance threats.

Repeating Threats Removal

Many threats are categorised as belonging to more than one security attribute. A threat may violate integrity as well as confidentiality, or a combination of any of the other security attributes. A list of threats retrieved for the security attributes on each instance means that the threats retrieved for the next security attribute might contain repeating information. Even if the threat is to a different security attribute, the countermeasures to those threats are exactly the same, which would mean suggesting a user repeat each action unnecessarily. These threats are removed so that the same threat is not repeated for each instance. This allows users to have a more manageable list of threats when presented with their results.

3.8 Related Work

Feng *et al* develop a Security Risk Analysis Model (SRAM) to identify the causal relationships among the components of distributed systems, in order to find the paths most likely to result in risk [54]. The presented

Bayesian network uses Ant Colony Optimisation (ANO) to analyse the graphs of the networks under observation. Using databases and probabilities determined from historical observations and domain experts, a security risk treatment plan is developed which identifies the most pressing paths of vulnerabilities which need to be defended. The evaluation is performed according to security ratings applied to given properties of the system (examples include data secrecy and data integrity policies being rated as high, medium or low). The interactions of these ratings determine the final results. Cloud-COVER concentrates on higher level views of the system, and considers a user's valuation of security attributes to determine the most pressing security priorities to systems.

Butler creates a structured approach to risk analysis called Security Attribute Evaluation Method (SAEM), an approach in which risks are analysed by quantifying threats based on several important indicators [55]. This includes the likelihood of the outcomes, potential hours lost to risk and potential revenue lost, following which SAEM considers the cost benefit analysis of taking care of each risk. SAEM assumes an initial security evaluation has taken place, requiring the knowledge of some security expertise in order to provide many of the required values. Cloud-COVER only assumes knowledge of system set up, and allows users to rate the potential impact that different threat categories could have on their deployment.

Kondakci creates a causal model between assets and threats, in order to analyse the logical relations between them [56]. Bayesian Belief Networks are used to analyse the dependencies, causalities, and risk propagation within the network. This analysis produces quantitative risk values which can then be used to determine where defences within a network should be prioritised. However, this model too requires previous evaluation of the network in order to identify assets, threats, the causal links between them, and likelihoods of risks. Cloud-COVER does not need any input of threat likelihood, as it contains such information in its default inputs. Cloud-COVER also consider causes of threats to be connection permissions, as it uses a higher level perspective.

3.9 Further Work

Although we have discussed ways of ordering the countermeasures, there are other ways to take the ordering of the countermeasures even further. As we have demonstrated that some attacks are much more likely to occur with regularity than others, it would also be useful to know how many of those attacks are conducted using the same vulnerabilities. This would help in being able to recommend the countermeasures which respond to those vulnerabilities to users first. The problem, as in other areas, is the lack of resources in this area. In an ideal world, by having access to details not only of the kind of attack used, but more specific details, such as the actual way in which systems are compromised, it would be possible to build a much better profile of how attackers take advantage of different kinds of threats. Detail of more statistics of successful attack would be one of the most useful resources to take this work even further. The idea of using statistics of previous attacks could be taken further by using Bayesian inference. Bayesian statistics is a way of using more recent information to revise understanding of the probability of events. It makes sense to use a statistical method which has been proven to demonstrate the ability to apply statistics to historical observations and use them to good effect. The statistics gathered using Bayesian methods could then be used to order the final priority of results.

Another important area which could be useful would be in seeing how using alternative valuations could affect the usability of Cloud-COVER. There may be changes which could be made with regard to the number of valuations used (more than two types of valuations, or to have an equality comparison between security attributes), or to have the valuations effect on the final results as being different. User reaction to these changes would be an important way to make Cloud-COVER more responsive and useful to user demands. This would involve analysing the way that users choose to value their security attributes when presented with more or fewer valuation levels, and in trying to understand what this means for the type of valuation used in Cloud-COVER.

3.10 Discussion

This chapter demonstrated a technique of modelling threat propagation within a cloud deployment. The important contribution of this chapter is the way in which the threat propagation is modelled, using an analysis of permitted operations given to connections, and specific security attributes, to understand when two connecting instance's (or rather, specific types of security defence on those instances) security defences need to be secured with the same priority. By abstracting away much of the detail (such as the specific nature of connections or instances) this model could be easily applied to other distributed systems in order to provide threat analysis as well. The priority of the threats, and propagating threats, are determined using the unique method of relative valuations provided by the user, which are covered in more detail in the previous chapter.

This chapter also presents the way in which the final ordering of results in Cloud-COVER are determined. The method presented here is one which allows users to distinguish between valuations which allow threats to be more equally weighted depending on probability (a more important than valuation) or one where none of the threats between the two security attributes can mix (a much more important than valuation).

Rank	Data	Sec. Attribute	Threat	Probability	Instance
1	-	Availability	Quality of service	High	All
=	-	Availability, confidentiality	Social engineering threats	High	All
3	-	Availability, confidentiality	Malicious Insider (cloud side)	Medium	All
=	-	Availability, confidentiality	Cloud service takeover/shutdown	Medium	All
5	-	Availability	Economic denial of service	Low	All

Table 3.5: An example of organisational threats, presented separately from the instance threats.

4 Providing Extensibility to Cloud-COVER

4.1 Introduction

Continual assessment is one of the important recommendations for a good threat modelling process.[1] Scandariato notes that the perspective of threat modelling should be applied depending on the assets being considered [58]. With every system being unique, this suggests the need to change perspective for every new threat modelling process, as this enables people to think about issues which previously have not been thought about [35]. The fact that threat modelling processes usually take one of three different perspectives anyway (attacker-centric, software-centric or asset-centric) demonstrate the fact that even within the various methods there is no single view of the best way to think about threats to systems.

By making some of Cloud-COVER's inputs extensible to users, this could allow them to make use of the ability to think about threats in ways that are most relevant to them and their circumstances. Another important reason to do this is because there are always likely to be threats which cannot be found within a given representation model. It makes sense that threats which may make sense when considered from a high-level perspective would not make sense at a low-level one, and that they only make sense at the perspective that a user understands them from. Only by thinking about ideas in different ways can users consider as

[1] Adam Shostack frequently repeats the mantra 'You are never done threat modelling' [57]. However he does concede that for organisations needing to ship products, they have to make a decision on when they are actually done threat modelling.

many possibilities as possible, if they have the time to do so.

Cloud-COVER's model is one in which many features of systems are abstracted away, and the analysis of the model concentrates on the threat propagation in which three main inputs determine the results presented to the user. These inputs are the threats to the deployment, user valuations of security attributes, and connection permissions. By allowing each of these inputs to be extensible, users can change zero, one or more of these inputs to shift the perspective of the model in ways that could allow them to consider things that otherwise they would not be able to model. For example, the data security attributes which are used as default include the CIA triad and compliance. However, these are not the only security attributes which could be used. For example, alternative security attributes could include authenticity. Allowing users to model threats using some of these, or even other security attributes they consider important, can allow for some important distinctions between threats to be made.

In Cloud-COVER's results, users are also presented with countermeasures to threats, and may need some help in distinguishing these countermeasures from each other. This is partly due to the fact that administrators are known to apply their own thinking regarding the amount of effort needed to protect particular machines, which influences which instances they protect. Heimann and Nochenson find that administrators generally have tipping points after which they choose not to protect machines [59]. Cloud-COVER's intention to be a well rounded and engineered tool means that at least some effort should be made to provide users with more knowledge of countermeasures to help their decision making.

This chapter presents extensibility features to Cloud-COVER, features which allow users to input their own threats, data security attributes and connection permissions. These features allow users to shift the perspective of the tool to one which better suits their own circumstances. By allowing users to change the model in ways that they may not have previously considered, Cloud-COVER emphasises the need to re-evaluate

previously completed threat modelling processes. This encourages an important feature of a good threat modelling process, of never thinking that one is finished. Instead users need to continue to think about the assumptions, considerations and viewpoints when threat modelling their system. This chapter also includes work on categorising countermeasures, in order to aid users in understanding which are best suited to their own circumstances.

This chapter is structured in the following way: we cover the reasons for allowing security attributes to be changed and examples of these in Section 4.2, we cover the same points for permissions in Section 4.3, and again for threats in Section 4.4. The implementation of this extensibility is discussed in Section 4.5, and the implications are included in Section 4.6. Related work is in Section 4.7, further work is covered in Section 4.8, and finally a discussion for the chapter is in Section 4.9.

4.2 Security Attributes

In addition to the more well known CIA security attributes, Cloud-COVER uses the security attribute of compliance. This was chosen because of the fact that cloud services may be based anywhere in the world, and issues such as data location are an important consideration for a variety of industries, such as defence and healthcare. The fact that none of the CIA security attributes covered this issue meant that it was important to include in any representation of cloud services. Figure 4.1 shows an example of valuations which include compliance, with Table 4.1 showing the threats from those valuations. However, compliance can cover more than a single issue and it is possible to consider splitting compliance into more than one security goal. A good example of the usefulness of the ability to add more security attributes to Cloud-COVER can therefore be demonstrated by splitting up compliance to consider additional security goals.

In this example, we split compliance up to consider three additional but separate security attributes. However, depending on one's perspective,

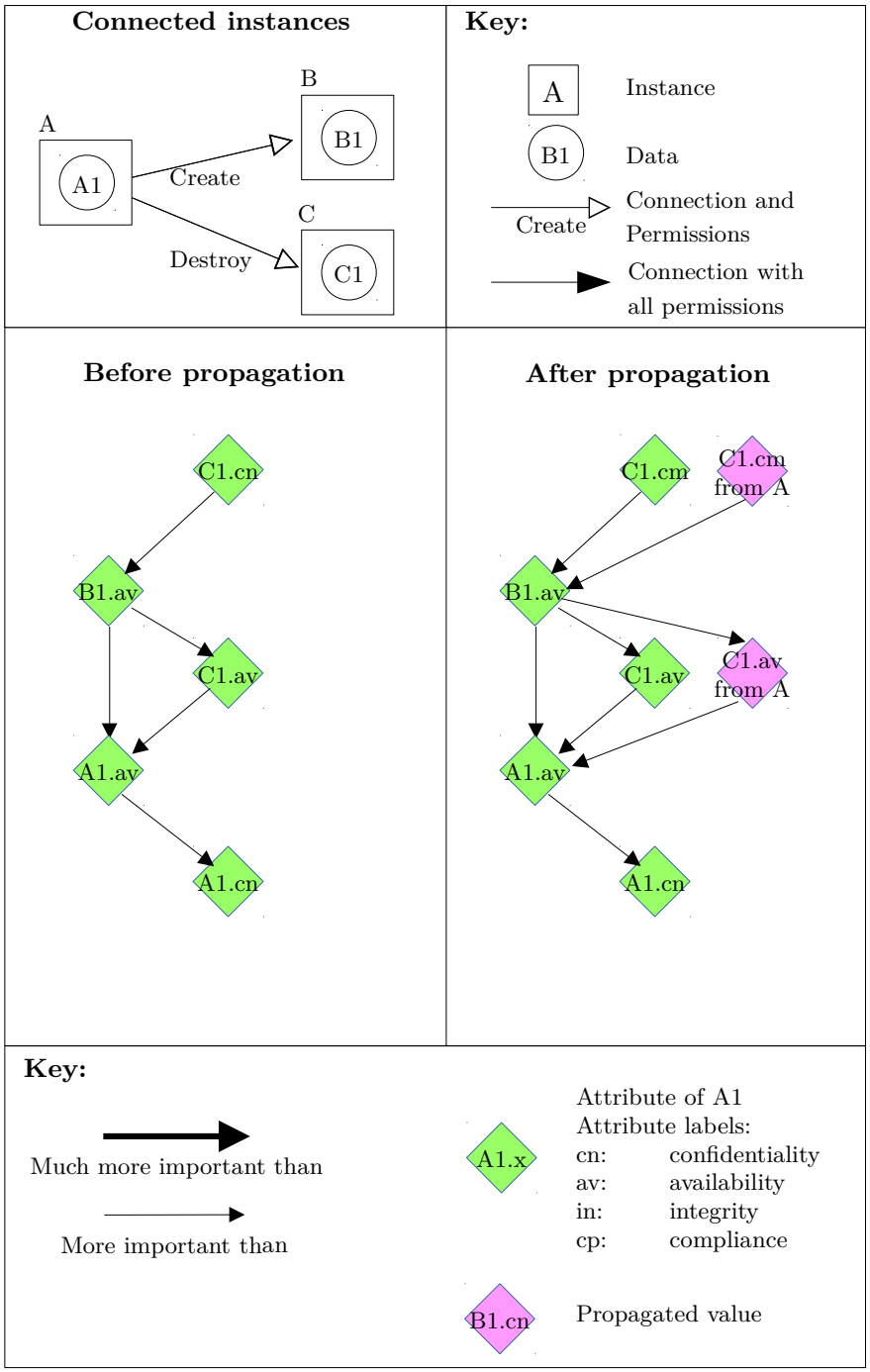


Figure 4.1: A figure considering compliance and its propagation, in contrast with Figure 4.2 where compliance is considered as three separate security attributes.

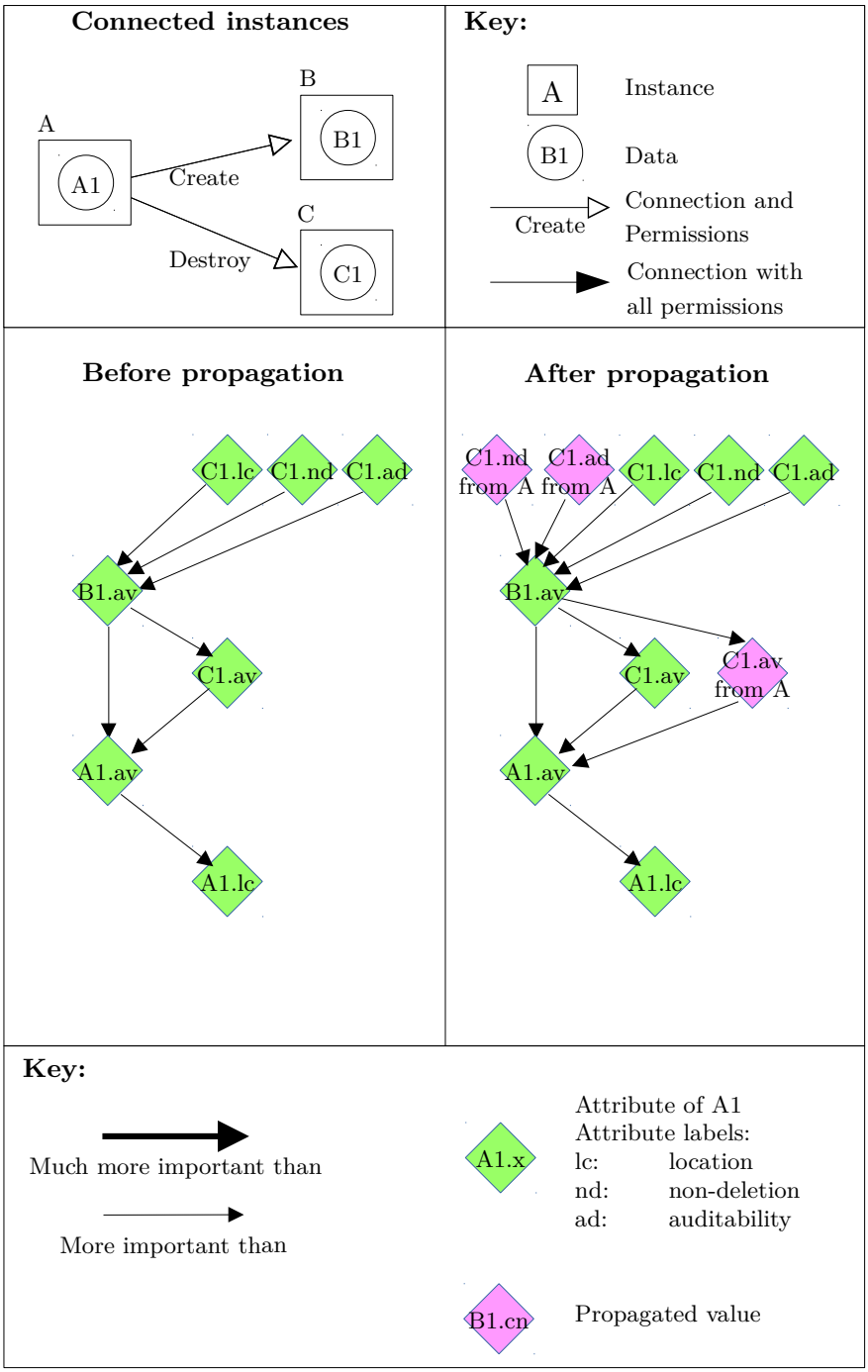


Figure 4.2: A figure considering compliance split into three additional security attributes and its propagations. One of the security attributes, location, does not propagate whilst the others do

Priority	Threats	On Instance	Data
1	Threats to compliance	C	C1
=	Threats to compliance	C (from A)	C1
3	Threats to availability	B	B1
4	Threats to availability	C	C1
=	Threats to availability	C (from A)	C1
6	Threats to availability	A	A1
7	Threats to confidentiality	A	A1

Table 4.1: The order of threats produced from Figure 4.1

it could be said to include even more. Data location, auditability, and non-deletion can each be considered to be a part of compliance. Data location specifically relates to the goal of keeping data within a certain geographic area or legal jurisdiction. Auditability refers to the ability to conduct monitoring on all actions performed on data, in order to ensure that no unacceptable actions take place. Non-deletion refers to the need for data never to be deleted for as long as it is needed. Each of these are clearly distinct and important considerations for an organisation to control.

Given the fact that these security attributes are different to each other, it makes sense that a user may want to make use of the extensibility feature to include them in their model. Table 4.3 shows the relevance of the newly defined data security attributes to the default connection permissions of Cloud-COVER. This shows how the threat propagations would occur with the newly included security attributes. When the table presents the security attribute/permission result as being positive, it means that the value of that security attribute can propagate. As can be seen in the table, data location is affected when the permission to move data is given to the connection. When the analysis is checking to see if the value propagates in that circumstance, it finds that it can and the value propagates. However when data location is being considered and the only connection permission is to create data, then the value does not propagate, since creating data at the other end of a connection does not

Priority	Threats	On Instance	Data
1	Threats to auditability	C	C1
=	Threats to auditability	C (from A)	C1
=	Threats to non-deletion	C	C1
=	Threats to non-deletion	C (from A)	C1
=	Threats to location	C	C1
6	Threats to availability	B	B1
7	Threats to availability	C	C1
=	Threats to availability	C (from A)	C1
9	Threats to availability	A	A1
10	Threats to location	A	A1

Table 4.2: The order of threats produced from Figure 4.2

affect its location. In Figure 4.2, only two of the three newer security attributes propagate, based on the values from Table 4.3.

The additional security attributes represented in the model, and the threats which may be relevant to each of them, would therefore appear in the results presented to the user. An example of this is can be seen in Table 4.2, which presents results from Figure 4.2. The fact that additional results have been found compared to Table 4.1 shows that this approach could yield useful results for dedicated threat modellers. The fact that the extra security attributes can be added to the model would mean that users could use them, or other security attributes, whenever the circumstances mean that including them would be useful to their threat modelling.

Any change to the security attributes considered by the model will also mean that the threats included need to be re-evaluated to see which security attributes they are able to violate. A security attribute like compliance which is split into a number of different security attributes for example, will mean that each additional security attribute will need to be considered separately. For example, Cloud-COVER needs to be told whether all existing (or new) threats are relevant to the three new security attributes in Table 4.3.

Sec. Attribute	Threat	Details
Data Creation		
Location	No	Data creation does not change its location
Auditability	Yes	Data creation needs to be created within process to make it auditable
Non-deletion	No	Data creation will not affect ability to non-deletion
Data Destruction		
Location	No	Data destruction does not move it to a new location
Auditability	Yes	Data destruction can destroy auditing data
Non-deletion	Yes	Data destruction violates non-deletion
Data Movement		
Location	Yes	Data movement can violate location security attribute
Auditability	No	Data movement can send on auditing data but does not destroy it
Non-deletion	No	Data movement cannot not affect non-deletion

Table 4.3: Table describing combination of permissions, security attributes and threats for new security attributes of location, auditability and non-deletion.

4.3 Permissions

Cloud-COVER's model considers the permissions given to connections, which represent the actions that the processes connecting from one instance are permitted to perform on another instance. The permissions used as default in Cloud-COVER (data creation, destruction and movement) are operations, and limit the number of threats that can be considered within the model, as they only represent a minimal set of permissions which could be controlled. In actual fact, any number of permissions may be controlled, depending on the connecting processes and the control that users have over them. The fact that other connection permissions could be used instead therefore means that this should be another Cloud-COVER input which should be extended to users, as there may well be circumstances in which particular permissions will change the way in which threats propagate.

For example, Cloud-COVER did not include data modification within its model. This choice was made because it is essentially either data destruction followed by creation of new data, or a movement of old data to a new location followed the creation of new data in the location of the moved data. Either way, the fact that it is a combination of operations which were already included meant that it was not included. For example, the CRUD functions considered in databases (Creation, Read, Update, Delete) are very similar, but consider the update category to be separate and required [60]. A user may therefore prefer to include this as one of the connection permissions. Alternatively, they may choose to include other permissions which are more relevant to their circumstances. Another alternative is to consider the permissions to read, write and execute. This can be understood more naturally by users since these permissions are regularly used to secure user accounts. These roles can also be defined with regard to data security attributes.

An example of the read, write and execute permissions considered with regard to security attributes is given in Table 4.4. This again shows how a users perspective can allow them to change Cloud-COVER's analysis

depending on their preferences. Again, the threat propagations would take place depending on the results found in this table. However, the fact that the execute permission can be seen to violate all security attributes demonstrates that in some situations, other actions may be more useful to consider.

Permissions can include any kind of limitations prescribed on a process. For example, a process may only be permitted to change things within a single folder, and not in others. This may prevent the destruction of particular data that a user considers important. These kinds of modelling considerations can be of use to users who want to include more details with regard to the permissions they have configured for their connections. The only thing that is important is that when considered together with the included security attributes, a user is able to state whether an attacker making use of those connections would be able to violate any security attributes.

4.4 Threats

Cloud-COVER's threats are an obvious input which a user might wish to change. Cloud-COVER's default mode is to consider higher level threats relevant from a systems perspective, as other threat modelling tools do not provide this. Additionally, lower level threats are better considered by threat modelling tools which take in and are therefore able to understand lower level details of systems. Lower level details include details such as data flow between threads and processes, and trust boundaries between these items. Cloud-COVER's modelled trust boundaries are between instances, and do not consider data flows within single instances, meaning that low level threats are more difficult to reason about. Nevertheless, the fact that Cloud-COVER's analysis always operates in the same way, means that users may wish to make use of changing this input too. Users can provide their own threats to the system, which they themselves have considered to be more relevant to Cloud-COVER's threat propagation.

Sec. Attribute	Threat	Details
Read		
Confidentiality	Yes	If attacker command a request to read, confidentiality can be violated
Integrity	No	Reading data does not violate its integrity
Availability	Yes	Reading data does deny its availability
Compliance	Yes	Requesting instance may lie outside of permitted locations
Write		
Confidentiality	Yes	Newly created data may require confidentiality
Integrity	Yes	Newly created data may require integrity check at later date
Availability	Yes	Newly created data can impact availability
Compliance	Yes	Newly created data may be subject to compliance requirements (such as location)
Execute		
Confidentiality	Yes	Execute may include any number of operations on data
Integrity	Yes	Execute may include any number of operations on data
Availability	Yes	Execute may include any number of operations on data
Compliance	Yes	Execute may include any number of operations on data

Table 4.4: Table describing combination of permissions, security attributes and threats. This example considers the user making use of extensibility, and including connection permissions of read, write and execute. Differences with Table 3.2 demonstrate the changes which can be made to these features.

Table 4.5: Input to Cloud-COVER about threats

Threat	Location	Auditability	Non-deletion
SQL Injection Modification	False	False	True

4.4.1 Additional Threats

As users can add security attributes, they may also add threats that they consider the default version of Cloud-COVER to have missed. This makes sense, as people may either add new threats that may not have been around or known about when Cloud-COVER's default threats were first included. Alternatively, they may consider that a threat which is listed in Cloud-COVER's default mode to be two distinct threats, as they may be able to defend against one without worrying about the effects of the other threat. These are all valid reasons to do this, and Cloud-COVER provides an easy structure through which the input file takes threats.

4.4.2 Re-evaluated Threats

When adding alternative security attributes to Cloud-COVER, the existing threats also need to be re-evaluated with regard to those new security attributes. We can consider the three alternative security attributes presented in Section 4.2, data location, auditability and non-deletion. Let us consider threats from SQL Injection modification, which is a threat to data because when this threat exists changes to data can be made by making use of vulnerabilities in the SQL language. This threat would only be a threat to non-deletion since the data would be deleted upon being changed. The data location would not change, since the modification could not change its location. Users must input the data to Cloud-COVER with the simple representation of the order of security attributes entered, similar to that of Table 4.5.

4.5 Implementation

The threats used in Cloud-COVER are taken from two input files. These files are used for two different but related purposes. The first input file asks users whether system data or an instance contains certain properties, which determine whether the threat they are related to exists. The second input file contains details of the threat itself, such as which security attributes it violates.

The two files are text files, open to be edited by users, only requiring adherence to the structure of the inputs (and sufficient security knowledge of the user). This allows users to tailor the tool in a way that suits their needs.

When analysing the input model, Cloud-COVER first needs to determine whether or not the property required for the threat to exist is present. This depends on the user entering the properties of the data, instances and connections. There are also some threats which exist without the need for a property to be present, such as the threats from not updating software. These threats exist without the need to check for their properties, and are loaded regardless of user input. For the analysis to determine that a threat exists, the entries need to be tied by a linking term (Threat ID, which is in string form), which is the final entry in both their structures, presented below.

The structure of the threat in the input file is:

Threat Name (String), Probability (Integer: 1 lowest, 3 highest), Propagating Threat (Boolean), Confidentiality Threat (Boolean), Integrity Threat (Boolean), Availability Threat (Boolean), Compliance Threat (Boolean), Threat ID (String)

The propagating threat boolean states whether the threat is one which propagates, whilst the data security attribute properties are used to specify if it can violate the data security attributes in question. The threat ID links a threat to the question file, which asks users whether the property which means the threat exists is present.

The structure of the data or instance question is: Threat property

question (String), Threat ID (String)

The threat property question only needs to establish the existence of the threat. For example, for SQL injection attacks, the question asks whether data is SQL data. A positive answer from the user confirms that the threat from an SQL injection attack exist, and this threat is then assigned to that instance. The threat ID must be identical in both input files for the match to be made by the analyser.

4.6 Implications

In the examples presented in this chapter, only one input at a time is changed whilst leaving the other inputs in their default states. It is perfectly acceptable for users to change zero, one or more inputs, depending on the input that they wish to change.

Each input is an important part of the analysis in Cloud-COVER. Allowing users to change aspects of the model can be useful in giving them the freedom to explore threats in ways they may not have thought much about before. By emphasising particular properties of systems or data, it is possible to find out more about the security issues of the systems being modelled.

The extensibility itself has one important caveat, which is that users must understand the way in which their modified inputs reflects their own understanding of Cloud-COVER's model. Providing users with this extensibility is to give them the trust that they can understand and use the model correctly for their own purposes. As long as users understand that if their understanding of security means that the security attribute/permission table makes sense to them for their own modelled system, then it makes sense for them to alter the model in that way to analyse threats.

One criticism which could be made would be to say that users should not be allowed to extend a model in such a way, as this may end in incorrect or unsafe advice being presented to them. However, the fact is that threat modelling does not just exist in the form of automated

tools, but also as manual processes. People who feel that they have a sufficient understanding of security issues should be entitled to provide their own inputs to Cloud-COVER if that is what they wish, since they might be able to use other threat modelling processes to accomplish the same results anyway.

Although Cloud-COVER has been adapted for extensibility, the limit of this extension is the way in which threats propagate. The analysis of user valuation and threat propagation is unique to Cloud-COVER, and is one of the reasons it is useful as an alternative to other threat modelling tools and techniques. By keeping this scope in place, users have knowledge of how the impact to instances is determined, and how the threat propagation takes place. Changing the way in which the propagation takes place has therefore not been considered.

4.7 Related Work

CORAS is a risk management methodology for system stakeholders to identify security flaws [61]. It has a semi-formal process, coming from its structured process but one which allows a degree of autonomy with regard to the perspectives taken to analyse the system. The formality comes from the use of documentation to ensure good communication between stakeholders (regarding goals, scope of analysis, and other issues) and for modelling (which is done using UML to identify security flaws). The modelling and risk identification stays loyal to a regular threat modelling approach. In contrast to other approaches however, CORAS suggests using a workshop of multiple participants to carry out a step-by-step walk-through, to use their different perspectives to identify risks in the system. Afterwards, the level of these risks, and whether they are acceptable or unacceptable is determined, along with the possible precautions to take. Cloud-COVER allows users to adapt its own model instead of advocating the use of multiple alternative processes, enabling users to reuse the same process in order to model an identical system from different perspectives.

Don Parker provides alternative security attributes in his so called 'Parkerian Hexad' [62]. These alternatives include possession, authenticity, and utility in addition to the CIA triad. Possession refers to ownership, since loss of ownership can bring about concern over the use of data. Authenticity refers to to veracity of origin, which separates it from integrity which is mainly concerned with proving that changes have not occurred. Utility refers to usefulness, since some data cannot be made use of without other things (such as encrypted data with no decryption key). Although Cloud-COVER has default security attributes of the CIA trio along with compliance, users are able to input whatever goals they have for their data, as long as it makes sense when considered along with how connection permissions affect threat propagation to other instances (like in Table 4.3).

4.8 Further Work

Although a part of this chapter looked at the presentation of countermeasures, they are an important point for any threat modelling process. Providing a better understanding of countermeasures, their relationships to one another, and most importantly, their effectiveness, is one of the most important things users need to be informed about. Sometimes, countermeasures are shared between threats. Other countermeasures also make sense to do at the same time as other ones. This is particularly useful for those which involve educating people within an organisation on important security matters. It would be useful therefore to explore ways in which the cost-benefit analysis of countermeasures could be presented to users, which would enable even better use of their resources.

4.9 Discussion

This chapter has presented extensibility features to Cloud-COVER's model, which allows users to input threats, data security attributes and connection permissions. In addition to making Cloud-COVER much

more useful by enabling users to shift the perspective of the analysis, this feature also shows the strength of Cloud-COVER's underlying model. The fact that Cloud-COVER is able to reason about threat propagation using only a few inputs is a demonstration of its well engineered nature.

5 Case Study

5.1 Introduction

Cloud-COVER was developed to solve specific issues with regard to the lack of threat modelling tools for systems (apart from SeaSponge). The previous chapters have outlined the underlying theory of Cloud-COVER, but it is also important to show that Cloud-COVER is able to provide useful output.

This chapter presents a case study looking at two users employing Cloud-COVER, one a security lecturer and another a security novice. For the security lecturer, a scenario is examined using two different methods (one being Cloud-COVER), with the major points of comparison between the two highlighted. For the security novice, the examination concentrates on usability, another of the original aims of this work. Although this case study is small in scope, it aims to demonstrate that Cloud-COVER is a useful vehicle for threat modelling purposes.

5.1.1 Cloud-COVER Usage

A typical session in which Cloud-COVER would require a user to know about the structure of their cloud deployment (including the connection permissions) and the data they wish to prioritise. The user would also need to have knowledge about data security attritubtes, in order to distinguish between those which they would need to provide preferences for. Although users do not need to have security knowledge, some security knowledge could help them to understand how the connection permissions and attributes would affect how results are gathered for the

final results, and how they can change their deployment in order to better protect against threats. After providing the input of the deployment, and then their preferences for the data security attribtes, Cloud-COVER then works on its analysis and presents users with the results. Cloud-COVER allows users to save their progress, so at any point during the input process, users can save the input and return to it later.

For the security lecturer, this knowledge could be assumed and he was able to use the tool without asking many questions. The security novice required a little advice on the mentioned issues before he was able to use the tool.

This chapter is structured in the following way: we present a case study comparing the approach of a security lecturer to the results obtained by Cloud-COVER in Section 5.2, and we present a usability study in Section 5.3. Possible further work is covered in Section 5.4 and we provide a discussion of the chapter in Section 5.5.

5.2 Security Lecturer

Frank

Frank is a security lecturer, who has had some experience in online security. His main experience is in penetration testing and securing websites, with some recent experience in cloud services and security. The example scenario presented here is an adaptation of a tool he has been working with as part of a team.

5.2.1 Frank's Approach

Frank's threat modelling approach consists in using a number of techniques, most of which involve utilising experience he has developed from securing systems he has worked on. There is no one particular process he uses, but rather he uses a combination of processes until he feels he has performed a rigorous analysis of the system. This can involve any of the following:

- Thinking conceptually about the system and its constituent parts and identifying the most obvious attack points.
- Extending this analysis to separating out smaller parts of the system from which launching other attacks could be useful (similar to threat propagation).
- Visiting security and exploit portals from which he is able to gather the latest information on threats.
- Using notes taken from previous security reviews, and combining these with his experience to think about possible attacks and threats.
- Using the above to think about assigning an overall rating of how secure each instance is (but not in a formal way).
- Attacking the system himself to find any other possible threats.

Frank applies the process to the instances within a deployment, and considers the connections and possible attacks which they could be used for. He uses personal security notes from previous systems he has helped to secure, and considers whether they contain any relevant threats. The use of security and exploit portals make him feel he is up to date in terms of his knowledge on threats to security and that he can use this to protect his system. In terms of threat modelling techniques, he has used manual threat modelling techniques like STRIDE in the past, but finds them too cumbersome to use each time for a new analysis.

There is no explicit consideration of security attributes, although he approaches this issue from the perspective of the purpose of each component or instance. Frank's primary consideration of possible attacks is in terms of their potential impact. His rating relies upon the degree of control the attack could exert over an instance, so that those gaining root access to systems would be the ones likely to have the biggest impact. This makes sense, and is an alternative to thinking about security in terms of data security attributes. The issue of probability is also considered,

but is mostly in the back of his mind, with potential impact being the main consideration.

The prioritisation of approaching the protection of the system happens on an ad-hoc basis, by observing the final produced analysis. Frank works his way through it by prioritising in his own mind what he thinks the most pressing needs are. He does this by selecting a few of the most important issues he identifies, and then repeats this process after fixing the initially identified issues. After having fixed the identified issues, he then uses automated tools for other things such as penetration testing to further look into security issues. The process outlined above is one which fits the general idea of a threat modelling process, although it is not a formal defined version with a given name.

5.2.2 Scenario

The scenario is an adaptation of a system which Frank was partially responsible for. The system is a cloud deployment for an advance flood warning website, which uses 4 cloud instances, as can be seen in Figure 5.1. Two instances are used as repositories for data (CW and HD in the figure), which are connected to a third instance (WM). WM is an instance using powerful computation to model the weather, producing results for flood warnings. The data for the warning system is forwarded to WS, which is a website available to the public.

- Instance CW provides regular updates about current weather data. Important concerns about the data from CW will be integrity and availability.
- Instance HD operates mainly as a data repository which includes historical data about the weather for the whole country. Integrity and availability will be big concerns, but compliance and confidentiality will not.
- Instance WM is a high performance computing service which models the weather using inputs from both CW and HD to try to under-

stand what the implications of the weather are likely to be. Each of the security attributes is important on this instance, although availability and integrity are the most important.

WM links to HD to request data when needed. CW's link to WM is not a request from WM, but rather CW pushing its data over when it has newer weather data.

- Instance WS is a website presenting the data produced for flood warnings to the public. The most important aspect of the data on this instance will be availability, and integrity of data from WM is also a concern.

  WM pushes data to WS to update the flood warning system, without WS requesting the data.

The valuations given for data security attributes are in Figure 5.2.

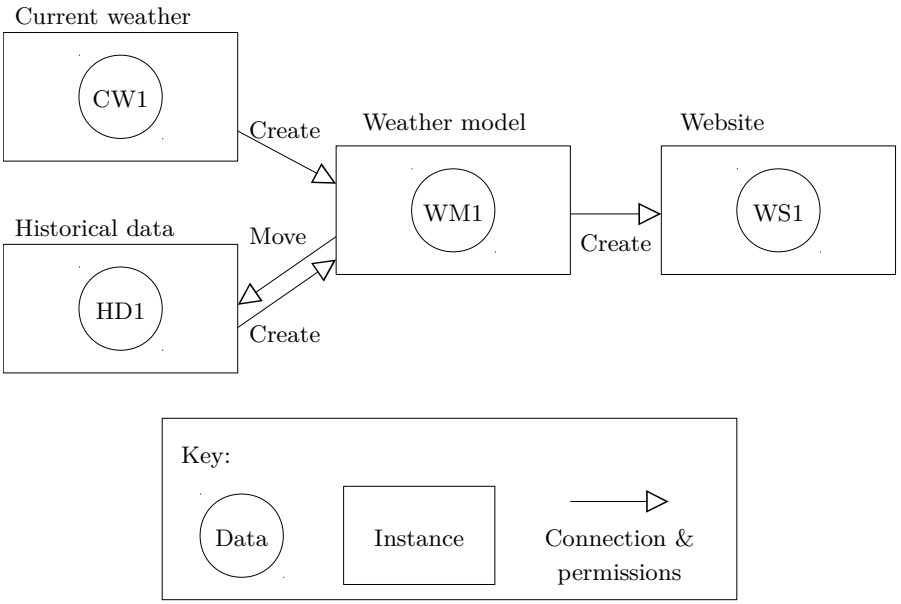


Figure 5.1: The model for the case study scenario. The scenario involves 4 instances, passing data between each other in order to produce a website for a flood warning system.

5.2.3 Cloud-COVER

Frank's input to Cloud-COVER took about 10 minutes, some time of which was spent considering the preference values for the data security attributes. The input of the deployment did not present any problems, with Frank working through the entry with relative ease.

Frank worked his way through the input stages, first entering information on the instances and their properties and then going through the connections and data. The longest time he spend was when considering the data security attributes and his preferences for them. With this input Cloud-COVER was able to produce the final list, with the top results visible in Table 5.1.

When entering details about the deployment, Frank did not like how threat modelling was not able to consider alternatives at the same time. For example, the current weather instance could be much more easily replaced than the instance storing historical data if necessary to do so. However, it did not mean that the availability of information was any less important. Although the human mind is easily able to comprehend this information, it was not obvious how this could be included in the model.

The results were presented in a way that Frank found easy to understand (as can be seen in Table 5.1), with the explicit prioritisation very clearly seen in the presentation. The highest priority was given to the threats emanating from instances CW and HD. Also the organisational threats were presented separately from the instance threats, which Frank was not used to. This is something which sets Cloud-COVER apart from other threat modelling tools. In terms of the results, the presented results make sense and look reasonable when compared to Frank's security attribute valuations. A screenshots of the top results can be seen in Figure 1 in the Appendix.

5.2.4 Frank's Approach

Frank's analysis works using a different perspective to Cloud-COVER, in which he essentially thinks about the most likely attacks to a system.

Threat	To Data On	Sec. Attribute	Propagates From
Most important			
Denial of service	CW	Availability	
Denial of service	HD	Availability	
SQL injection malware	HD	Availability	
SQL injection modification	HD	Availability	
SQL injection insertion	HD	Availability	
SQL injection deletion	HD	Availability	
Next Most important			
Insecure APIs	WM	Availability	
Denial of service	WM	Availability	
SQL injection malware	WM	Availability	
SQL injection modification	WM	Availability	
SQL injection insertion	WM	Availability	
SQL injection deletion	WM	Availability	
Encrypted communications hack	CW	Availability	
Unautharised user data deletion	CW	Availability	
Unautharised user code execution	CW	Availability	
Encrypted communications hack	HD	Availability	
Unautharised user data deletion	HD	Availability	
Unautharised user code execution	HD	Availability	

Table 5.1: The list of top threats to Frank's deployment from Cloud-COVER's results. The remaining threats are covered in the Appendix in Tables 4-7.

Attacks and groups of similar attacks are similar to what Cloud-COVER thinks of as threats. For example, an SQL injection attack is a threat to SQL and its data, and can be performed in many ways. Frank would list SQL attack as one type of attack on that instance, and then consider other attack types he has knowledge of. In order to provide a comparison, these attacks have been grouped together as threats when presenting the results of his analysis in Table 5.2, although it is perfectly reasonable to highlight that there are differences between the two perspectives despite the similarities.

Frank's analysis took around 90 minutes, time during which he was using internet resources and his own notes to identify possible threats. Also Frank was clear that the initial analysis is unlikely to be the final one, as additional analysis is a frequent issue with threat modelling. Input to Cloud-COVER took around 10 minutes, a significant time saving. Any additional analysis needed from changes made to the system could also provide a significant time saving.

Frank concentrated on the three data hosting and producing instances, leaving the website to be the least most important in his consideration. The table shows that the threats he considers to be a priority are similar to the ones listed in Cloud-COVER, apart from his desire to check the software on the instances to make sure that they are not making the system unsafe. Cloud-COVER does not check individual items of software, but asks whether out of date software exists on the system. Although Cloud-COVER does consider this issue, Frank is more intent on specifically checking this issue himself by looking at individual applications. However, Frank was made aware that for users who wish to do so, the extensibility feature would allow him to enter individual items of software to Cloud-COVER as threats if he wanted to do so.

In terms of threats identified, one attack type which was not listed in Frank's analysis is the virtualisation environment breach. Although the probability of this threat is low, the past couple of years have actually started seeing increasing virtualisation platform patches being applied. This is because this type of threat seems to be becoming more real, and

Threat	To Data On	Frank's Priority
Denial of service	CW	Yes
Communications attacks	CW	No
Software or OS out of date	CW	Yes
Unautharised user attacks	CW	No
Denial of service	HD	Yes
SQL injection attacks	HD	Yes
Communications attacks	HD	No
Software or OS out of date	HD	Yes
Unautharised user attacks	HD	No
Insecure APIs	WM	No
Denial of service	WM	No
SQL injection attacks	WM	No
Software or OS out of date	WM	No

Table 5.2: The top threats found using Frank's method.

the timing of these patches have caused some problems for cloud users [63] [64]. This demonstrates that Frank's personal experience was not able to compensate for this, although at the same time the types of attacks he identified covered almost all of the same ones identified by Cloud-COVER.

Finally, Frank's analysis does not perform an explicit consideration of security attributes with regard to the connections and the possible range of attacks this either enables or disables. The fact that Frank acknowledges that each instance's outgoing connections have to be judged based on the possibilities opened up by the connections means that there is an awareness of threat propagation. But by dealing with them as they come around, the comparison between the Cloud-COVER analysis and Frank's analysis is difficult to evaluate.

5.2.5 Cloud-COVER Advantages

As a person with security knowledge, Frank produced a security analysis which covered a large amount of the threats identified by Cloud-COVER.

There were some factors however which demonstrate the value of Cloud-COVER when compared with his manual threat modelling results.

Time Frank's analysis took around 90 minutes, which was much longer than the 10 minutes it took for the whole process of inputting the deployment and getting the results from Cloud-COVER. Frank was very impressed with how quickly an analysis was able to be produced on Cloud-COVER.

With regard to the presented countermeasures, both methods leave users much work to do in terms of finding out how to take action against those threats. Although Cloud-COVER presents countermeasures to users, users still need to do research of their own to find out details regarding implementation on their own individual systems.

Cloud-COVER Perspective Cloud-COVER provides an alternative vantage point from which to judge threats to systems, and with the extensibility feature even allow users to change the model's perspective. Frank's method of finding specific kinds of attacks and working to protect them does not mean that he will have identified other similar attacks which may need different defences. At least these can all be grouped together under a single heading in Cloud-COVER, while it is plausible that Frank could miss important examples of attacks with his methods.

Also, Frank did not classify organisational threats separately in his analysis, but was aware of the differences in his own mind. Some of the organisational threats are definitely not ones which would necessarily be considered threats in non cloud systems, so are important to inform users about. For example, data loss is an easy problem to solve in a regular computer, as one can retrieve the physical hardware. Since this is not possible in the cloud, it is important that any cloud specific issues are not forgotten about, since many users (especially cloud newcomers) who may be

unfamiliar with them.

Threat Knowledge Frank was not aware of a couple of listed threats among the organisational threats presented: cloud provider non-deletion and economic denial of service attack (EDoS). Although the EDoS attack is still a low probability one, it would still be a good idea to implement protections for it. Frank was more concerned about what non-deletion of data could mean for information meant to be confidential.

This suggests that even for people who consider themselves to be up to date, the area of security is one which moves forward so quickly that even knowledgeable people can miss important information.

Explicit prioritisation Frank's method meant identifying threats and then working through them in a methodical way. Cloud-COVER identified the same threats and presented them with an initial priority attached. Frank observed that although he did not disagree with any of the highest priority threats presented (in terms of his own priorities), he would still be likely to go through any threats presented with his own method (of picking the most important, fixing them, then returning to see what remained).

However Frank also stated that one of the useful points of this feature would be to present the results to a superior in the workplace, without needing to justify that the results were influenced largely by the desire for more resources to protect with (although the user could still manipulate the results, in theory).[1] This could be done, for example, by using Cloud-COVER in front of the person (and agreeing about the given input ratings) and demonstrating the importance of those security issues.

[1] The issue of workplace superiors rarely giving enough resources to with which to secure systems with is a frequent complaint in the security section of The Register. Trevor Pott, a contributor who regularly discusses his experiences as a system administrator, states that he was never given the correct resources to do his job in his 20 years in such roles [65].

5.2.6 Cloud-COVER Disadvantages

Frank produced a detailed analysis of threats, and some of this analysis was done using methods which are not available through the medium of a threat modelling tool.

Human nuances The fact that Cloud-COVER was not able to consider alternatives at the same time was described earlier.

Even if Cloud-COVER included the possibility of including alternatives, it is just as likely that other equally difficult concepts (easy for humans, difficult for models) exist. In other words, it is important to acknowledge that the importance of knowledge and experience in security cannot simply be made redundant by a threat modelling tool.

Threat consideration Although Cloud-COVER does ask users about whether they include software which is not regularly updated on each instance, Frank's approach is one in which he personally chooses to check this for himself, and thought that Cloud-COVER's questioning may be a bit lacking in this regard. Frank's approach makes sense, but at the same time could be included in Cloud-COVER by making use of the extensibility feature.

This question is about how much detail Cloud-COVER asks of its users, and there is a case to make that a bit more detail could make for a more useful set of results. This will not be a problem for Cloud-COVER's model to cope with, and is something which makes sense to consider. The problem is that software which is insecure today, becomes secure tomorrow through features such as software updates. It is a difficult balancing act, but one which could be countered by, for example, constantly updating Cloud-COVER's default list of inputs. But the question of just how much information users are asked about is still an important one, and is one of the main areas that could be worth exploring for the future.

5.2.7 Not Reviewed

Extensibility Although the extensibility feature was not used in this case, it was demonstrated to Frank to see his reaction to it. Frank appreciated the value of the feature, especially as someone with an interest in exploits and a slightly different perspective to the one Cloud-COVER uses.

Complexity Using a scenario involving 4 instances, with 3 connections between those instances, still took Frank much longer to consider using his own process than it took Cloud-COVER. The reality is that given a larger deployment, with even more connections (and instances with any more than one outgoing connection), threat analysis will take much longer (for both methods).

Frank agreed that given a scenario involving any additional things to consider, it would take a huge amount of thinking and work in order to properly identify threats to each part of the system.

5.2.8 Case Study Thoughts

Frank's approach to threat modelling is similar in process to what one would expect. The problem for Frank is that he does not have a structured approach to threat modelling, and instead has a rather ad-hoc approach. Although a lot of people would use a regular structured approach, such as a specific process, there are also many who, like Frank, would not. Whether using a structure approach or not however, in manual processes there is always the possibility of people missing threats. Even single threats are important to cover, and there were a few examples of threats not being covered by Frank's analysis, even if a couple of them are considered to be low probability threats.

If we also consider perfect analysis which does not miss any threats and produces all the threats that Cloud-COVER also does, Cloud-COVER can still be of use. Cloud-COVER's extensibility should allow for parts of almost any process to be automated, but it would require users taking

the time to work on the inputs. Once finished however, it should still be able to produce useful results. It would also allow for people to think about how else they would be able to identify threats by making use of this feature.

The most obvious advantage that Cloud-COVER can provide to anyone, even the best threat modelling experts, would be to save them time. Frank was extremely impressed by this aspect of Cloud-COVER, especially since the extensibility feature could allow him to adapt it to something closer to his own perspective of security. The fact that he had some different ideas with regard to Cloud-COVER meant that he thought this could be a very useful tool. The other useful aspects of the time saving would be money saving for businesses and individuals. Also, the ability to quickly use it to demonstrate the security needs of a deployment to any workplace superiors being hesitant about providing money for security.

5.3 Usability Study

5.3.1 Cloud User

Gabriel is a researcher who has been using cloud computing for several years. His experience is mainly in cloud portability, and he has used a variety of cloud platforms and services. He has little knowledge of security, but occasionally needs to use security features (such as single sign on).

Gabriel was taken through the use of Cloud-COVER, and some basic security issues were explained in order for him to understand how to use the tool. This comprised of the following steps:

- Basic security knowledge. The idea of a beachhead in attacking a distributed system was explained, and why deployments can be defended more effectively by analysing them to find where threats exist. Also the idea of data security attributes, why they make a difference, and the need to provide values for them later.

- The way to enter instances, data, and connections in order to input the deployment.
- The way to provide preferences between the data security attributes for the final rankings.
- The way to understand the presentation of the results.

Before using the tool, Gabriel was very inquisitive and asked several details about how it worked. He entered the details of his deployment, then looked at the presented results. Since Gabriel knows so little about security, there was no other threat modelling analysis that Cloud-COVER's results could be compared to. Instead, his overall thoughts as a security novice and use of Cloud-COVER are covered below.

5.3.2 Positives

- Gabriel did not know about data security attributes before, but commented on how he appreciated how and why they would be valuable to differentiate between the kinds of threats faced on the cloud. This suggests that these concepts can be easily explained to people with minimal security knowledge, and is something which can be easily included in a Cloud-COVER use guide.
- Gabriel found input to be very easy, something he found surprising. As a security novice, he expected to have some difficulty with entering system information and the language used when doing so. He expressed the thought that he might need guidance during initial use (and was especially wary of coming across technical terms), but seemed to find entry self explanatory.
- Gabriel found the results presentation to be clear. He also liked the presentation of the countermeasures and the fact that he could navigate between them by their categorisation. This was an important issue to him as a security novice.

- Gabriel expressed a preference for the relative preference valuations when asked if he would prefer numeric valuations instead.

5.3.3 Negatives

- Gabriel did not like some of the data and instance property descriptions, due to the mix of using both positive and negative statements to confirm the existence of threats. Although positive confirmations made more sense to him (such as "Data is SQL data"), he did not like the negative confirmations (for example, "Instance data is not regularly backed up"), which he found a bit confusing.

  This issue was one which was considered during design, and ultimately left in due to the way the extensibility feature works. By linking the confirmation of a property (such as data being SQL data) with the existence of a threat (such as SQL injection attacks), users can easily add security threats to Cloud-COVER's analysis. Changing the method of extensibility would make it more difficult to use. For this reason, although the issue is a valid one to raise, it is not something which was changed.

- Gabriel felt that connection permissions could have been better explained on screen. This is an example of how the less knowledgeable users need to be considered more in terms of the user interface.

- Gabriel felt that the input could be improved, by allowing for instances with identical properties to simply be cloned and renamed, without needing to repeat the input process. This is an interesting feature which would be a useful addition to Cloud-COVER's usability.

5.3.4 Neutral

- Gabriel did not need any of the features of extensibility, and expressed satisfaction as long as predefined threats were loaded by

the tool.

- Although he seemed to understand how the probabilities affected the rankings, he did not use both valuation types because he was unsure about how getting very different rankings resulting from their use would mean for protecting the deployment overall. The advantages of using the different valuations may need to be explained in more detail to Cloud-COVER users.

5.3.5 Usability Study Thoughts

Although Gabriel expressed some issues with the tool, most of the concerns had either already been considered, or were smaller user interface issues which can be dealt with during possible further development. This suggests that the user interface has some issues which could be made better. In terms of analysis and results, Gabriel seemed to find the tool easy to use, and understood the way the results were presented.

One of the original aims of Cloud-COVER's development was to be able to provide a usable threat modelling tool for security novices. Although Gabriel is a single user and not a representative sample of novice users, his ability to easily input his system and understand the presented results suggests that the original aims have largely been addressed.

5.4 Further Work

Given more time, it would have been useful to have produced a bigger case study, demonstrating the way in which Cloud-COVER could aid users of larger deployments than the one presented. This could have demonstrated the amount of time and effort Cloud-COVER could save, and threats which could easily have been missed in such a complex analysis. Cloud-COVER's scalability has been tested on deployments with values for 10, 20, 30, 40, and 50 data items, with the processing and output time found to increase linearly. However, it would be a good

idea to test this with increasingly larger scenarios. In addition, Cloud-COVER's extensibility features, allowing for users to shift the perspective of the threat modelling process, would also be very important to include in any extensive case study.

Also, a single user is not a representative enough sample to prove that the tool's usability is good enough. It would be helpful to test Cloud-COVER's usability on a larger number of people. One interesting angle would be to see how cloud computing novices (not just security novices) would find it. Although the examples presented in this case study are useful, it is also clear that there are other features of Cloud-COVER which would be worthwhile to highlight in a larger case study.

5.5 Discussion

This case study has presented some important findings. Although people with security knowledge can produce a very thorough analysis, and find the vast majority of threats that Cloud-COVER comes up with, there are still small examples which might slip through. However, it has also demonstrated factors such as how human thought will always be an important part of a process such as threat modelling, with the issue of human nuances coming up. This shows how human factors can never be ignored in important issues like security. Given that there was not a massive difference in terms of the amount of threats found by the two approaches, one of the most important findings was the vast difference in time in order to produce the results for both, with 80 minutes saved when using Cloud-COVER, and without the need to consult additional resources.

The real positive point for Cloud-COVER is when considering more complexity, Cloud-COVER would be able to perform the same role but for a much larger scenario. A larger scenario would require a much more complicated analysis to figure out the connections and the way in which they should affect the way that defences should be prioritised across any deployment. The other major benefit is that security novices would

also be able to use Cloud-COVER to produce a similar threat modelling analysis as people with a good level of knowledge on security.

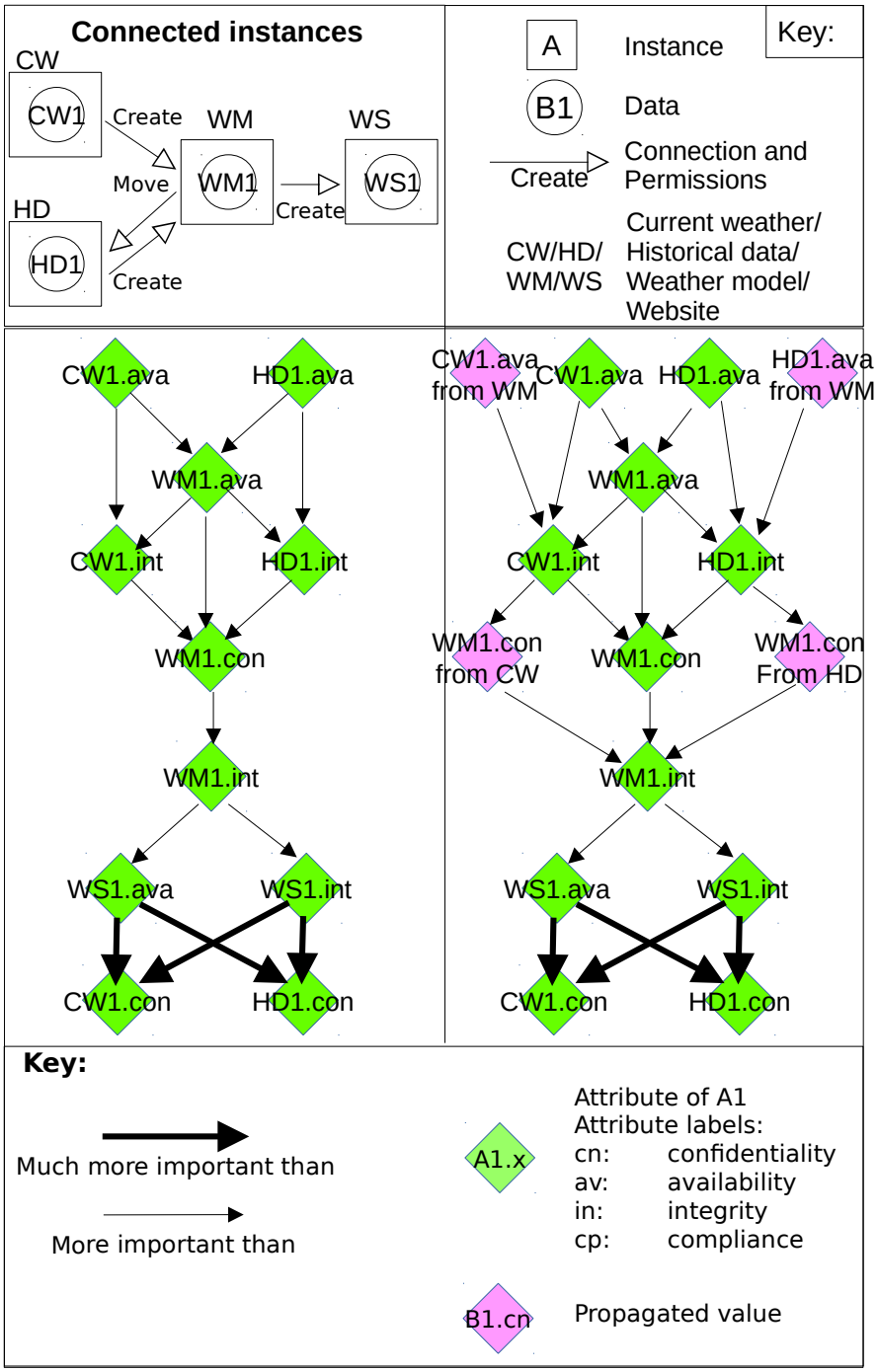


Figure 5.2: The supplied values for the data security attributes, and the values before and after propagation.

6 Conclusions

6.1 Introduction

This thesis presents Cloud-COVER and its underlying model and threat propagation analysis. Cloud-COVER is a threat modelling tool for cloud computing, which presents users with threats to their cloud deployments, threats identified by analysing the input of their deployment. By applying the threat modelling process at a higher level than the low-levels that it has traditionally been used for, Cloud-COVER is able to cover an area that other tools have not.

Cloud-COVER has a default model which analyses the way threats propagate depending on their relevance, and ability to propagate, based on three things: threats, user valuations of data security attributes, and connection permissions. However, Cloud-COVER has been modelled in such a way that any of those three inputs can be adapted by the user. The model has been abstracted in a way that the analysis always takes place in the same way, but the inputs can let users change the perspective of the analysis, allowing for a more useful threat modelling process.

There is the possibility of taking the work presented in this thesis further, some of which we have covered in the penultimate sections of each chapter. However, the lack of other methods which allow users to determine threats to systems using the methods presented here, of analysing threat propagation, and of using relative valuations, suggests that Cloud-COVER should be immediately useful for anybody looking for a tool capable of providing such a service.

Each section below provides an overview of the contributions made in each of the chapters presented in this thesis.

6.2 Cloud-COVER

Chapter 2 introduces two important parts in the design of Cloud-COVER, the model used to represent the cloud deployments and the user supplied valuations of data security attributes.

The chapter presents the underlying model of Cloud-COVER and how it is used to represent user's deployments. It is an important part of how threat propagation analysis takes place, and also features prominently when discussing Cloud-COVER's extensibility and adaptability in Chapter 4.

This chapter also presents the way in which Cloud-COVER takes input from users based on their relative preferences between data security attributes. By using this alternative valuation system instead of numerical valuations, Cloud-COVER is better able to justify the ordering of the presented threats, in contrast to some of the more ambiguous interpretations which result when using a numerical system. This is a unique and novel way in which to rate threats to a system, and solves some of the problems known to exist when using numerical valuations.

6.3 Threat Propagation

Chapter 3 presents the way in which Cloud-COVER analyses threat propagation, and builds on the idea of the input values from Chapter 2. Using the user valuations, and depending on connection permissions and the security attributes being considered, Cloud-COVER determines whether certain kinds of threats exist over the links connecting instances. If they do represent a threat, the priority given to the propagating threat is taken from the user supplied valuations. This allows a consistent priority to be given to the threats when the results are presented to the users.

The way in which threat propagation is analysed is unique to Cloud-COVER, with its consideration of the combination of connection permissions and data security attributes. By looking at these issues from a high

level perspective, this analysis is able to provide a useful viewpoint from which to understand threat propagation in cloud computing.

6.4 Providing Extensibility to Cloud-COVER

Chapter 4 considers another important part of the threat modelling process, of considering threats from alternative perspectives. Although Cloud-COVER's default operation is one we are satisfied provides users with worthwhile information about threats to their system, there are additional changes which could make it even more useful. The constant need to re-evaluate threats to systems, and from different perspectives, is an important part of a good threat modelling process. Allowing extensibility of system inputs to the user can allow for this to take place.

This is done by allowing the three inputs of threats, permissions, and data security attributes to be changed by the user. The chapter presents specific examples in which additional threats could be found depending on the inputs which are changed. The way in which different perspectives can mean additional, or reduced numbers of threats, depending on what is being considered, suggests not only the usefulness of this feature, but also the well engineered nature of Cloud-COVER and its underlying model.

6.5 Case Study

Chapter 5 presents a case study featuring a security lecturer and a security novice. For the security lecturer, the chapter compares the results of the tool and the results from the lecturer's own threat modelling process, and highlights the advantages and disadvantages of Cloud-COVER. For the security novice, Cloud-COVER's usability is covered.

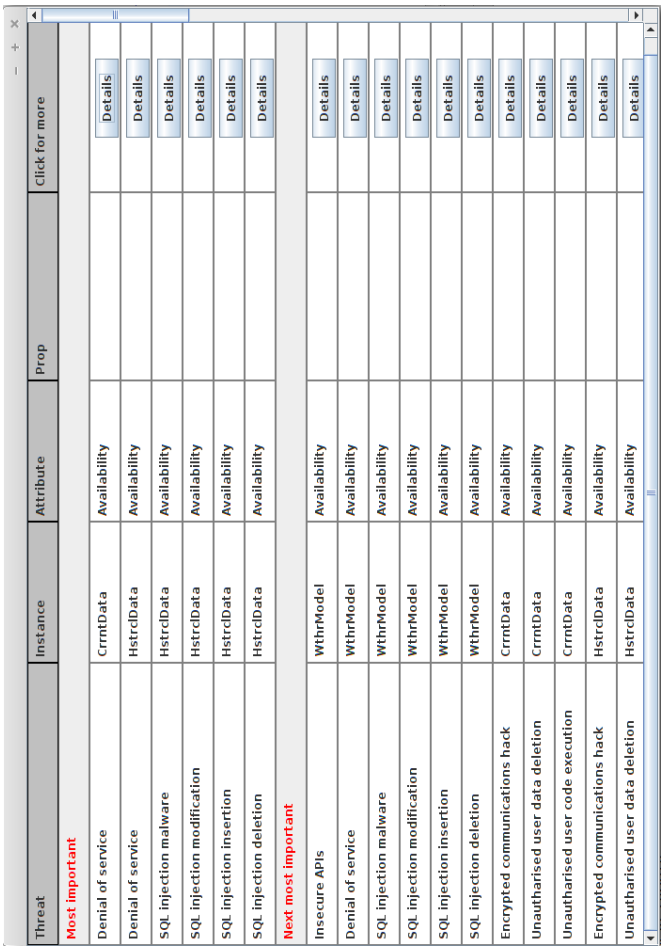
This case study demonstrates that the main goals envisioned when originally developing the tool, of helping to identify the important threats to cloud deployments, and of being accessible to people with little security knowledge, has been achieved.

6.6 Final Thoughts

This thesis has presented research making significant contributions to the area of computer security, and in particular, threat modelling. Although threat modelling processes and tools are useful, they are not widely used, do not offer much help to those looking to protect systems, and have not been applied to cloud computing. By taking the general idea of threat modelling, adapting it to cloud computing, and subjecting it to academic rigour, we have developed and presented the underlying theories and model of Cloud-COVER. Cloud-COVER is a threat modelling tool for cloud computing developed to aid the identification of threats to cloud deployments.

By developing a structured and extensible tool, we have strengthened the theory behind threat modelling whilst keeping to the original threat modelling process. We have also developed novel ways of solving specific issues relating to the valuation of threats in such processes, by allowing users to value data security attributes using relative preferences. In addition, Cloud-COVER has been intended to be easy to use for those less knowledgeable on security matters. By aiming the tool at users including security novices, it is hoped that security knowledge can be made accessible to more people, which can hopefully help in reducing some of the anxiety many people have of using cloud computing.

Appendix



Threat	Instance	Attribute	Prop	Click for more
Most important				
Denial of service	CrrntData	Availability		Details
Denial of service	HstrclData	Availability		Details
SQL injection malware	HstrclData	Availability		Details
SQL injection modification	HstrclData	Availability		Details
SQL injection insertion	HstrclData	Availability		Details
SQL injection deletion	HstrclData	Availability		Details
Next most important				
Insecure APIs	WthrModel	Availability		Details
Denial of service	WthrModel	Availability		Details
SQL injection malware	WthrModel	Availability		Details
SQL injection modification	WthrModel	Availability		Details
SQL injection insertion	WthrModel	Availability		Details
SQL injection deletion	WthrModel	Availability		Details
Encrypted communications hack	CrrntData	Availability		Details
Unautharised user data deletion	CrrntData	Availability		Details
Unautharised user code execution	CrrntData	Availability		Details
Encrypted communications hack	HstrclData	Availability		Details
Unautharised user data deletion	HstrclData	Availability		Details

Figure 1: A screenshot of the results for the case study.

Table 1: The values of organisational threats and the attributes they are a threat to.

	Probability	Confidentiality	Integrity	Availability	Compliance
XSS/Browsing threats	High	Yes	Yes	Yes	Yes
Social engineering threats	High	Yes	Yes	Yes	Yes
Poor security practice	High	Yes	Yes	Yes	Yes
Account hijack	High	Yes	Yes	Yes	Yes
Cloud service takeover/shutdown	Medium	Yes	Yes	Yes	Yes
Malicious Insider (cloud side)	Medium	Yes	Yes	Yes	Yes
Data Lock In	Medium	No	No	No	Yes
Cloud Data Non-Deletion	Medium	Yes	No	No	Yes
Economic denial of service	Low	No	No	Yes	Yes
Quality of service	Low	No	No	Yes	Yes
Location and legal risks	Medium	No	No	No	Yes

	Probability	Propagates	Confidentiality	Integrity	Availability	Compliance
SQL injection deletion	High	Yes	No	No	Yes	No
SQL injection insertion	High	Yes	Yes	Yes	Yes	No
SQL injection modification	High	Yes	No	Yes	Yes	No
SQL injection malware	High	Yes	Yes	Yes	Yes	Yes
Unautharised user code execution	Medium	Yes	Yes	Yes	Yes	Yes
Unautharised user data viewing	Medium	Yes	Yes	No	No	No
Unautharised user data modification	Medium	Yes	No	Yes	No	No
Unautharised user data deletion	Medium	Yes	No	No	Yes	Yes
CMS content deletion	Medium	No	No	No	Yes	No
CMS content insertion	Medium	No	No	No	No	Yes
CMS content modification	Medium	No	No	Yes	No	No

Table 2: 1 of 2. The values of threats and the attributes they are a threat to.

	Probability	Propagates	Confidentiality	Integrity	Availability	Compliance
Denial of service	High	No	No	No	Yes	No
Privilege escalation	High	Yes	Yes	Yes	Yes	No
Insecure software	High	Yes	Yes	Yes	Yes	Yes
Insecure APIs	High	Yes	Yes	Yes	Yes	Yes
Undetected intrusions	High	No	Yes	Yes	Yes	Yes
Attacker Account Access	High	No	Yes	Yes	Yes	Yes
Insecure communications	High	No	Yes	Yes	Yes	Yes
Secure communications breach	Medium	No	Yes	Yes	Yes	Yes
Data Loss	Low	No	No	No	Yes	Yes
Virtualisation environment breach	Low	No	Yes	Yes	Yes	No

Table 3: 2 of 2. The values of threats and the attributes they are a threat to.

Threat	To Data On	Attribute	Propagates From
Most important			
Denial of service	CrrntData	Availability	
Denial of service	HstrclData	Availability	
SQL injection malware	HstrclData	Availability	
SQL injection modification	HstrclData	Availability	
SQL injection insertion	HstrclData	Availability	
SQL injection deletion	HstrclData	Availability	
Next Most important			
Insecure APIs	WthrModel	Availability	
Denial of service	WthrModel	Availability	
SQL injection malware	WthrModel	Availability	
SQL injection modification	WthrModel	Availability	
SQL injection insertion	WthrModel	Availability	
SQL injection deletion	WthrModel	Availability	
Encrypted communications hack	CrrntData	Availability	
Unautharised user data deletion	CrrntData	Availability	
Unautharised user code execution	CrrntData	Availability	
Encrypted communications hack	HstrclData	Availability	
Unautharised user data deletion	HstrclData	Availability	
Unautharised user code execution	HstrclData	Availability	

Table 4: 1 of 4. Cloud-COVER's list of threats to Frank's deployment in Chapter 5

Threat	To Data On	Attribute	Propagates From
Next Most important			
Encrypted communications hack	WthrModel	Availability	
Unautharised user data deletion	WthrModel	Availability	
Unautharised user code execution	WthrModel	Availability	
Virtualisation environment breach	CrrntData	Availability	
Virtualisation environment breach	HstrclData	Availability	
Next Most important			
Insecure APIs	CrrntData	Confidentiality	WthrModel & any other cons
SQL injection malware	CrrntData	Confidentiality	WthrModel & any other cons
SQL injection insertion	CrrntData	Confidentiality	WthrModel & any other cons
Insecure APIs	HstrclData	Confidentiality	WthrModel & any other cons
SQL injection malware	HstrclData	Confidentiality	WthrModel & any other cons
SQL injection insertion	HstrclData	Confidentiality	WthrModel & any other cons
Unautharised user data modification	CrrntData	Integrity	
Unautharised user data modification	HstrclData	Integrity	
Virtualisation environment breach	WthrModel	Availability	

Table 5: 2 of 4. Cloud-COVER's list of threats to Frank's deployment in Chapter 5

Threat	To Data On	Attribute	Propagates From
Next Most important			
Unautharised user data viewing	WthrModel	Confidentiality	
SQL injection modification	CrrntData	Integrity	WthrModel & any other cons
SQL injection modification	HstrclData	Integrity	WthrModel & any other cons
Unautharised user data viewing	CrrntData	Confidentiality	WthrModel & any other cons
Unautharised user code execution	CrrntData	Confidentiality	WthrModel & any other cons
Unautharised user data viewing	HstrclData	Confidentiality	WthrModel & any other cons
Unautharised user code execution	HstrclData	Confidentiality	WthrModel & any other cons
Next Most important			
Denial of service	Website		
Unautharised user data modification	CrrntData	Integrity	WthrModel & any other cons
Unautharised user data modification	HstrclData	Integrity	WthrModel & any other cons
Unautharised user data modification	WthrModel	Integrity	

Table 6: 3 of 4. Cloud-COVER's list of threats to Frank's deployment in Chapter 5

Threat	To Data On	Attribute	Propagates From
Next Most important			
CMS content modification	Website	Integrity	
Unautharised user data modification	Website	Integrity	
Encrypted communications hack	Website	Availability	
CMS content deletion	Website	Availability	
Unautharised user data deletion	Website	Availability	
Unautharised user code execution	Website	Availability	
Next Most important			
Virtualisation environment breach	Website	Availability	
Next Most important			
Unautharised user data viewing	CrrntData	Confidentiality	
Unautharised user data viewing	HstrclData	Confidentiality	

Table 7: 4 of 4. Cloud-COVER's list of threats to Frank's deployment in Chapter 5

List of References

[1] M. Aydin and J. L. Jacob, "Cloud-COVER: Using User Security Attribute Preferences and Propagation Analysis to Prioritise Threats to Systems," in *2015 European Intelligence and Security Informatics Conference, Manchester, United Kingdom, September 7-9*, 2015, pp. 53–60.

[2] ——, "Providing Extensibility of Attributes, Permissions and Threats for Cloud-COVER's Threat Modelling Tool," in *2016 European Intelligence and Security Informatics Conference, Uppsala, Sweden*, 2016, Submitted for consideration.

[3] Raya Jalabi, [9 July 2015], "OPM hack: 21 million people's personal information stolen, federal agency says," [Online] Available: http://www.theguardian.com/technology/2015/jul/09/opm-hack-21-million-personal-information-stolen, Accessed May 2016.

[4] Chris Welch, [7 November 2013], "Over 150 million breached records from Adobe hack have surfaced online," [Online] Available: http://www.theverge.com/2013/11/7/5078560/over-150-million-breached-records-from-adobe-hack-surface-online, Accessed May 2016.

[5] B. Schneier, *Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World*. WW Norton & Company, 2015.

[6] T. Garfinkel and M. Rosenblum, "When Virtual is Harder Than Real: Security Challenges in Virtual Machine Based Computing Environments," in *Proceedings of the 10th conference on Hot Topics*

in Operating Systems - Volume 10. Berkeley, CA, USA: USENIX Association, 2005, pp. 20–20.

[7] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds," in *Proceedings of the 16th ACM conference on Computer and communications security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 199–212.

[8] R. Yeluri and E. Castro-Leon, *Building the Infrastructure for Cloud Security: A Solutions View*. Apress, 2014.

[9] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, "Controlling Data in the Cloud: Outsourcing Computation Without Outsourcing Control," in *Proceedings of the 2009 ACM workshop on Cloud computing security*. ACM, 2009, pp. 85–90.

[10] Bill Bitner and Susan Greenlee, [18 April 2016], "z/VM âĂŞ A Brief Review of Its 40 Year History ," [Online] Available: http://www.vm.ibm.com/vm40hist.pdf, Accessed May 2016.

[11] S. Garfinkel and H. Abelson, *Architects of the Information Society: 35 Years of the Laboratory for Computer Science at Mit*. Cambridge, MA, USA: MIT Press, 1999.

[12] Y. Chen, V. Paxson, and R. Katz, "Whats New About Cloud Computing Security," *University of California, Berkeley Report No. UCB/EECS-2010-5 January*, vol. 20, no. 2010, pp. 2010–5, 2010.

[13] A. Belapurkar, A. Chakrabarti, H. Ponnapalli, N. Varadarajan, S. Padmanabhuni, and S. Sundarrajan, *Distributed Systems Security: Issues, Processes and Solutions*. Wiley, 2009.

[14] F. Daniel, F. Casati, V. D'Andrea, E. Mulo, U. Zdun, S. Dustdar, S. Strauch, D. Schumm, F. Leymann, S. Sebahi, F. d. Marchi, and M.-S. Hacid, "Business Compliance Governance in Service-Oriented Architectures," in *Proceedings of the 2009 International Conference on*

Advanced Information Networking and Applications. Washington, DC, USA: IEEE Computer Society, 2009, pp. 113–120.

[15] D. Birk and C. Wegener, "Technical issues of forensic investigations in cloud computing environments," in *Systematic Approaches to Digital Forensic Engineering (SADFE), 2011 IEEE Sixth International Workshop on*, May 2011, pp. 1–10.

[16] Bruce Schneier, [30 January 2013], "People, Process and Technology," [Online] Available: https://www.schneier.com/blog/archives/2013/01/people_process.html, Accessed May 2016.

[17] M. Harkins, *Managing Risk and Information Security: Protect to Enable*, 1st ed. Berkely, CA, USA: Apress, 2012.

[18] ISO/IEC 27005:2011, "[1 June 2011] Information technology — Security techniques — Information security risk management," [Online] Available: http://www.iso.org/iso/catalogue_detail?csnumber=56742, Accessed May 2016.

[19] G. Box and N. Draper, *Empirical Model-building and Response Surfaces*. Wiley, 1987.

[20] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 2008.

[21] Microsoft, "The STRIDE Threat Model," [Online] Available: https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx , Accessed May 2016, 2002.

[22] M. Howard and D. LeBlanc, *Writing Secure Code*. Pearson Education, 2003.

[23] Larry Osterman, [4 September 2007], "Threat Modeling Again, STRIDE," [Online] Available:

https://blogs.msdn.microsoft.com/larryosterman/2007/09/04/threat-modeling-again-stride/, Accessed May 2016.

[24] Microsoft, "SDL Threat Modeling Tool," [Online] Available: https://www.microsoft.com/en-us/sdl/adopt/threatmodeling.aspx, Accessed May 2016.

[25] Information Security Exchange, [17 May 2015], "Threat modelling tools and alternative threats," [Online] Available: http://security.stackexchange.com/questions/89441/threat-modelling-tools-and-alternative-threats , Accessed May 2016.

[26] M. Abi-Antoun, D. Wang, and P. Torr, "Checking threat modeling data flow diagrams for implementation conformance and security," in *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*. ACM, 2007, pp. 393–396.

[27] Firesheep, [Online] Available: http://codebutler.com/firesheep/?c=1, Accessed May 2016.

[28] Wireshark, [Online] Available: https://www.wireshark.org/, Accessed May 2016.

[29] B. Schneier, "Attack Trees," *Dr. Dobb's Journal*, vol. 24, no. 12, pp. 21–29, 1999.

[30] P. Meland, D. Spampinato, E. Hagen, E. Baadshaug, K. Krister, and K. Velle, "SeaMonster: Providing Tool Support for Security Modeling," 2008, nISK-2008 conference.

[31] SeaSponge, [Online] Available: https://github.com/mozilla/seasponge, Accessed May 2016.

[32] I. A. Tøndel, M. G. Jaatun, and P. H. Meland, "Security Requirements for the Rest of Us: A Survey," *Software, IEEE*, vol. 25, no. 1, pp. 20–27, 2008.

[33] W. Yurcik, G. A. Koenig, X. Meng, and J. Greenseid, "Cluster Security as a Unique Problem with Emergent Properties: Issues and Techniques," in *5th LCI International Conference on Linux Clusters: The HPC Revolution 2004*, 2004, pp. 18–20.

[34] David LeBlanc, [14 August 2007], "Threat Modeling Again, STRIDE," [Online] Available: https://blogs.msdn.microsoft.com/david_leblanc/2007/08/14/dreadful/, Accessed May 2016.

[35] A. Shostack, "Experiences threat modeling at microsoft," 2008, Modeling Security Workshop. Dept. of Computing, Lancaster University, UK.

[36] ——, *Threat Modeling: Designing for Security*. Wiley, 2014.

[37] R. Hasan, S. Myagmar, A. J. Lee, and W. Yurcik, "Toward a threat model for storage systems," in *Proceedings of the 2005 ACM workshop on Storage security and survivability*. ACM, 2005, pp. 94–102.

[38] Trustwave, "2012 Global Security Report," [Online] Available: https://www.trustwave.com/gsr, Accessed May 2016.

[39] M. Aydin, "Cloud-COVER," [Online] Available: https://github.com/mustafadoe/CloudCOVER, Accessed May 2016.

[40] Dana Epp, [17 January 2012], "The Evolution of Elevation: Threat Modeling in a Microsoft World," [Online] Available: https://technet.microsoft.com/en-us/security/hh778966.aspx, Accessed May 2016.

[41] Bruce Schneier, [27 February 2007], "CYA Security," [Online] Available: https://www.schneier.com/blog/archives/2007/02/cya_security_1.html, Accessed May 2016.

[42] OWASP, [19 April 2013], "OWASP Testing Guide," [Online] Available: https://www.owasp.org/images/1/19/OTGv4.pdf, Accessed May 2016.

[43] Trustwave, "Trustwave Global Security Report," [Online] Available: https://www.trustwave.com/gsr, Accessed May 2016.

[44] Cisco, "Cisco Security Reports," [Online] Available: http://www.cisco.com/c/en/us/products/security/security-reports.html, Accessed September 2016.

[45] Symantec, "2016 Internet Security Threat Report," [Online] Available: https://www.symantec.com/security-center/threat-report, Accessed September 2016.

[46] Websense, "2015 Threat Report," [Online] Available: https://www.websense.com/assets/reports/report-2015-threat-report-en.pdf, Accessed September 2016.

[47] S. Feruza and T.-h. Kim, "IT Security Review: Privacy, Protection, Access Control, Assurance and System Security," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 2, no. 2, pp. 17–32, 2007.

[48] UK National Archives, [16 July 1998], "Data Protection Act 1998," [Online] Available: http://www.legislation.gov.uk/ukpga/1998/29/contents, Accessed May 2016.

[49] U.S. Government Printing Office, [30 July 2002], "Sarbanes-Oxley Act of 2002," [Online] Available: https://www.gpo.gov/fdsys/pkg/PLAW-107publ204/html/PLAW-107publ204.htm, Accessed May 2016.

[50] P. Mell, K. Scarfone, and H. Romanosky, "S. A Complete Guide to the Common Vulnerability Scoring System Version 2.0," [Online] Available: https://www.first.org/cvss/cvss-guide, Accessed May 2016.

[51] P. Saitta, B. Larcom, and M. Eddington, "Trike v.1 Methodology Document," [Online] Available: http://octotrike.org/papers, Accessed May 2016.

[52] W. T. Harwood, J. A. Clark, and J. L. Jacob, "Networks of Trust and Distrust: Towards Logical Reputation Systems," 2010, Workshop on Logics for Security (Copenhagen).

[53] ENISA, [8 December 2012], "Cloud Computing: Benefits, risks and recommendations for information security," [Online] Available: https://resilience.enisa.europa.eu/cloud-security-and-resilience/publications, Accessed May 2016.

[54] N. Feng, H. J. Wang, and M. Li, "A security risk analysis model for information systems: Causal relationships of risk factors and vulnerability propagation analysis," *Information Sciences*, vol. 256, pp. 57 – 73, 2014, business Intelligence in Risk Management.

[55] Butler, "Security Attribute Evaluation Method: A Cost-Benefit Approach," in *Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference on*, May 2002, pp. 232–240.

[56] S. Kondakci, "A Causal Model for Information Security Risk Assessment," in *Information Assurance and Security (IAS), 2010 Sixth International Conference on*, Aug 2010, pp. 143–148.

[57] Adam Shostack, [15 October 2014], "BruCON 0x06 - Keynote [Video File]," [Online] Available: https://www.youtube.com/watch?v=-2zvfevLnp4, Accessed May 2016.

[58] R. Scandariato, K. Wuyts, and W. Joosen, "A Descriptive Study of Microsoft's Threat Modeling Technique," *Requir. Eng.*, vol. 20, no. 2, pp. 163–180, Jun. 2015.

[59] C. F. L. Heimann and A. Nochenson, "Identifying tipping points in a decision-theoretic model of network security," *CoRR*, vol. abs/1203.2824, 2012.

[60] Maarten Mullender, "CRUD, Only When You Can Afford It," [Online] Available: https://msdn.microsoft.com/en-us/library/ms978509.aspx , Accessed May 2016, 2004.

[61] M. S. Lund, B. Solhaug, and K. Stølen, *Model-Driven Risk Analysis: The CORAS Approach*. Springer Science & Business Media, 2010.

[62] D. B. Parker, *Fighting Computer Crime: A New Framework for Protecting Information*. New York, NY, USA: John Wiley & Sons, Inc., 1998.

[63] Neil McAllister, [1 October 2015], "Patch NOW: VMware vCenter, ESXi can be pwned via your network," [Online] Available: http://www.theregister.co.uk/2015/10/01/vmware_patches/, Accessed May 2016.

[64] Chris Williams, [29 October 2015], "Patch this braXen bug: Hypervisor hole lets guest VMs hijack hosts," [Online] Available: http://www.theregister.co.uk/2015/10/29/xen_security/, Accessed May 2016.

[65] Trevor Pott, [9 July 2015], "I cannae dae it, cap'n! Why I had to quit the madness of frontline IT," [Online] Available: http://www.theregister.co.uk/2015/07/09/why_i_quit_it_sysadmin_overloads/?page=2, Accessed May 2016.