DEGREE PROJECT IN TECHNOLOGY,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2022

Goat Farm: Learning by hacking

Jonas Hulthén & Thea Nöteberg

KTH ROYAL INSTITUTE OF TECHNOLOGY
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Authors

Thea Nöteberg <theanot@kth.se> and Jonas Hulthén <jhulthen@kth.se>
Information and Communication Technology
KTH Royal Institute of Technology

Place for Project

Stockholm, Sweden

Examiner

Pawel Herman Place
KTH Royal Institute of Technology

Supervisor

Viktor Engström

Stockholm

KTH Royal Institute of Technology

Abstract

Cloud computing has seen a big increase during the last decade. Many companies are seeing the benefits of cloud computing and are migrating over to the cloud however, migrating to the cloud raises customer-related security risks. Amazon web services (AWS) is the biggest cloud provider and a majority of the AWS accounts are at the risk of data breaches. Companies need to learn how to defend against hackers and one of the tools that can increase knowledge surrounding cloud cyber security is CloudGoat. CloudGoat gives the defender hand-on hacking experience by providing a handful of hacking challenges however, these hacking challenges are all static. The hacking community is always evolving and to not fall behind this report proposes GoatFarm, a tool to automatically assemble hacking challenges from predefined building blocks. Goatfarm used one of the scenarios presented in CloudGoat as a starting point and generated two new similar hacking challenges. The hacking challenges were set up to verify the generated scenarios and the hacking challenge was traversed. The results showcased that the generated hacking challenge was traversable which meant that it possible to generate hacking challenges however, GoatFarm only generated two scenarios. These generated hacking challenges can be used as a learning experience but more hacking challenges need to be generated to create a complete learning experience. The verification could also be more effective by using a tool to assess the generated hacking challenges automatically.

Keywords

CloudGoat, hacking challenges, cloud computing, generated, learning by hacking, cyber security, AWS.

Sammanfattning

Molntjänster har ökat signifikant under det senaste decenniet. Många företag ser fördelarna med molntjänster och migrerar över till molnet, men att migrera till molnet ökar kundrelaterade säkerhetsrisker. Amazons webbtjänster (AWS) är den största molnleverantören och en majoritet av AWS-kontona löper risk för dataintrång. Företag behöver lära sig att försvara sig mot hackare och ett av verktygen som kan öka kunskapen kring cybersäkerhet i molnet är CloudGoat. CloudGoat ger försvararen en hand-on hackerfarenhet genom att tillhandahålla en handfull hackningsutmaningar, men dessa hackningsutmaningar är alla statiska. Hacking grupper utvecklas hela tiden och för att inte hamna på efterkälken föreslår den här rapporten GoatFarm, ett verktyg för att automatiskt sammanställa hackningsutmaningar från fördefinierade byggstenar. Goatfarm använde ett av scenarierna som presenterades i CloudGoat som utgångspunkt och lyckades generera två nya liknande hackningsutmaningar. För att verifiera de genererade scenarierna sattes hackningsutmaningarna upp och hackningsutmaningen stegades igeom. Resultaten visade att den genererade hackningsutmaningen var överkomlig vilket innebar att det var möjligt att generera hackningsutmaningar, men GoatFarm genererade bara två scenarier. Dessa genererade hackningsutmaningar kan användas som en inlärningsupplevelse, men fler hackningsutmaningar måste genereras för att skapa en komplett inlärningsupplevelse. Verifieringen kan också bli mer effektiv genom att använda ett verktyg för att automatiskt bedöma de genererade hackningsutmaningarna.

Keywords

CloudGoat, hackingutmaningar, molntjänster, generering, lärande av hacking, cyber säkerhet, AWS.

Acronyms

AWS	Amazon Web Service
EC2	Amazon Elastic Compute Cloud provide computing power in the form of a virtual machine.
VPC	Amazon virtual private cloud virtual network that can handle different AWS resources.
IAM	Identity access management authenticates certain users in the cloud environment.
S3	Amazon simple storage service is used to store data (amazon doc)

Contents

Chapter 1

Introduction

The concept of the internet of things has seen a big increase during the last decade. Everyday items such as cars, sensors, and TVs are being connected to the internet at a rapid pace which could change the way we work, live, and communicate. [17] Cloud computing is one of the emerging techniques, which is used in things we do daily such as playing games, writing documents, storing data, and more. Cloud computing differs from traditional computing resource storage by keeping everything on the internet which enables customers to access the cloud from anywhere and on any device [8]. Many companies are seeing the benefits of using cloud computing resources but as more companies migrate to the cloud, customer-related security becomes more important.

One of the biggest cloud providers, Amazon web services (AWS) has vulnerabilities in a majority of the existing AWS accounts. Ermetic points to the fact that misconfigurations in identities could expose 90% of amazon Simple Storage Service (S3) buckets [9]. This makes a majority of companies vulnerable to potential security breaches. One example of this is when WizCase security team found that SeniorAdvisor had a misconfigured S3 Bucket that exposed three million senior citizens info [6]. Another incident was in 2017 when the personal data of 123 million American households were exposed due to a misconfigured S3 bucket [2].

At the same time, the trend of customer-related breaches is not slowing down. Gartner estimates that in 2025, 99% of all cloud breaches will be caused by customer-related issues [12]. This means that companies have to learn how to prevent these vulnerabilities.

To prevent potential hackers from abusing vulnerabilities in the cloud, the defender needs to understand how the hackers are able to attack their environment. One way to do this is to conduct a penetration test. Penetration tests are done to find vulnerabilities in a company which could then be addressed and fixed [7]. To conduct a penetration test information needs to be gathered surrounding the object that will be tested. However, cloud computing environments can be different from customer to customer. This makes it hard for penetration tests to be conducted in each and every environment.

As a solution to the problem, there are several tools to assess security. One example is Amazon Inspector which automatically scans the AWS workload for software vulnerabilities [1]. Another tool that can be used to assess the security of the cloud is the Meta Attack Language (MAL). MAL is an attack simulation tool developed at KTH to test attack graphs. MAL uses a domain-specific attack language that uses code to simulate the attack logic of a specific domain. [14].

These tools can help identify vulnerabilities, however, the best way to educate cloud customers on security is hands-on learning. One study done at the University of Washington showed that students in a class with traditional learning were 1,5 times more likely to fail the course than students who had hands-on learning [10]

Rhino security labs created the hacking learning tool Cloud Goat which can give cloud customers a hands-on learning experience about cloud computing security [5]. Cloud goat is an Amazon web service deployment tool that deploys “vulnerable by design” cloud environments (Rhino Security Labs, 2022). It is created as a learning experience where the user identifies vulnerabilities in different cloud computing environments to practice their cybersecurity skills. However, these hacking challenges are all static.

1.1 Problem Statement

The problem with the static hacking challenges is that they are always the same. In the real world, new exploits and vulnerabilities are discovered which means that cybersecurity needs to adapt and change in order to compete. One way to adapt to this is to have cloud computing environments generated, therefore our plan is to investigate if static cloud hacking scenarios can be replaced by generated cloud hacking scenarios.

The generated cloud hacking environments will be based on the scenarios presented in CoudGoat. We plan on answering the question:
How can traversable cloud hacking scenarios be generated?

1.2 Scope

This project will focus on generating hacking scenarios by modifying one of the AWS resources. Cloudgoat has multiple cloud computing environments and each environment has multiple AWS resources that can be changed. Trying to generate new environments for each CloudGoat scenario and each AWS resource would imply work outside of the scope of this project. To reduce the complexity of the project this report will be limited to the CloudGoat scenario cloud_breach_s3 and only focus on modifying the IAM instances. The project will work as a proof of concept where a successful generation of an environment with modified AWS resources implies that more environments can be automatically generated. Therefore the aim of the project is to create a tool that could generate hacking scenarios with an achievable goal. However, the goal of this project is not to identify weaknesses in the AWS service but rather to identify vulnerabilities in misconfigured cloud environments.

1.3 Outline

2. *Background* - In this chapter, a technical background with concepts such as Cloud Computing, AWS, and threats to cloud computing. It then describes Cloud Goat and ties it all together with related work.

3. *Approach* - In this chapter, we describe how our experiment was conducted and our method for verifying.

4. *Result* - In this chapter, the scenarios are presented and how they are verified to have a goal.

5. *Discussion* - In this chapter, a discussion is made based on the results and the related work.

6. *Conclusion* - In this chapter, a conclusion is presented based on the report and how the research can be continued.

Chapter 2

Background

2.1 Technical background

2.1.1 Cloud Computing

Cloud computing is the concept of providing access to computational resources that can be accessed anywhere and with any device through the internet [18]. The cloud service uses a 'pay-per-use' policy where customers pay for the time they use a computing resource. This 'pay-per-use' policy has multiple benefits. Some customers could not enter certain IT markets due to high budgets to maintain large data centers. With the introduction of cloud computing services, the cost of a data center is shared among all of the cloud customers. This led to some customers being able to enter certain IT markets they weren't able to before.

Cloud services are offered in three different service models, Infrastructure as a Service (IaaS), Platform as a Service(PaaS), and Software as a Service (SaaS) [18]. IaaS is one of the fastest-growing fields in cloud computing [15]. This service model lets the customer deploy and run the software by giving them computing resources such as processing, storage, and network [18]. This is made possible by letting the customer use small virtual resources from a large physical resource. Amazon Web Services uses IaaS by giving their customer a virtual machine that can be modified as a physical server. This model gives the customer the most control but requires the most amount of work [13]. Amazon Web Service is the world's most used cloud computing platform. They are used by everything from large enterprises, leading government agencies, and growing startups [4]. Amazon provides many different cloud computing services with AWS.

The ones used in this project are EC2, VPC, S3, and IAM.

Cloud computing has its fair share of benefits but there are also risks and challenges[18]. Traditional computational resources were stored on-site which enabled high control over the resources. But with cloud services the control is transferred to the provider instead of the customer. This means that the customer has no control over the security of the cloud system as well as no administrative power. Cloud services have several cybersecurity challenges to address.

2.1.2 Threats to Cloud Computing

In the report ”Top Threats to Cloud Computing” several threats to cloud computing are presented [2]. Some of the threats explained in this report are listed below.

IAM

IAM stands for Identity access management, it authenticates that a user has permission to access certain cloud resources. But flaws in the IAM systems can lead to malicious actors being able to abuse these vulnerabilities to gain access to data they were not authorized to see.

Misconfigured resources

Cloud computing resources are highly complex to configure which makes misconfigured resources a common occurrence. Misconfigured cloud computing resources is when some resources are wrongly set up which can cause vulnerabilities in those resources.

In traditional computing resources data centers changes could take several days but with the introduction of cloud computing these changes could take seconds. This is a consequence of cloud computing only being an abstraction of the traditional computational resources in the hardware data centers. This raises the complexity and makes the cloud computing approaches more dynamic which makes it harder to design an absolute change control.

Lack of Cloud Security

Many companies transfer to the Cloud without a proper cloud security strategy. Companies see the benefits of cloud computing and transfer to the cloud. When doing this companies need to have a security strategy to protect their resources from malicious actors.

The knowledge concerning good Security Strategy practice in the cloud is still new and not known by many companies. This results in companies using traditional security control practices when migrating IT resources to the cloud which leads to vulnerabilities that can be abused by malicious adversaries.

At the same time companies often favor completing the transfer fast instead of establishing a secure architecture and strategy before doing the transfer. In order to attempt a transfer from traditional computational resource storage to cloud storage, a company needs to implement proper cloud security architecture and strategy.

Data breaches

Data Breaches are when a non-certified person gains access to sensitive or confidential information. A data breach can be caused by vulnerabilities in applications, human error, or bad security practices. Example of this is in 2016 when an AWS account belonging to Uber was hacked which led to the personal info of 57 million people being stolen.

2.2 CloudGoat

CloudGoat is an Amazon web service deployment tool that deploys “vulnerable by design” cloud environments [5]. It is created as a learning experience where the user identifies vulnerabilities in different scenarios to practice their cybersecurity skills. These scenarios are statically created by Rhino Security Lab.

2.2.1 Cloudgoat scenarios

Many hacking scenarios exist in CloudGoat. The relevant scenario to this project is Cloud_breach_S3.

Cloud_breach_S3 - The goal of this scenario is to access the sensitive data stored in

the S3 bucket [5]. This scenario includes 1 VPC containing one EC2 instance and one S3 bucket. The scenario starts off with the attacker acquiring the IP address of the EC2 instance. The EC2 is running a misconfigured reverse-proxy server which enables the attacker to acquire credentials of the IAM instance profile connected to the EC2. With the IAM instance profile, the attacker is able to download the sensitive data in the S3 bucket.

2.3 Related work

2.3.1 Security scenario generator

An experiment was made by the University of Birmingham where they created virtual machines (VMs) based on generated scenarios with vulnerabilities, these were made for people to learn to hack [3]. They were inspired by hacking challenges such as Catch the flag (CTF), which is an activity where big groups of people compete to reach a hacking goal. It has been used in education to give students a more hands-on experience to learn hacking skills. With CTF as an inspiration, they created the generated scenarios in the form of vulnerable VM's that was used in a CTF event.

The CTF event was branded as a positive learning experience, however, the number of participants was limited. The report created a CTF event using their VMs generator and out of the 18 participants, 81% expressed that their knowledge of cyber security increased because of the event. Even though the number of participants is few, this report still points to the fact that generated hacking challenges can be used as a learning experience.

This experiment shares similarities with our experiment. Both experiments generate hacking challenges, however, this experiment only focused on creating VMs with vulnerabilities while we want to generate cloud environments with vulnerabilities. Furthermore, they tested their creation with external parties to see how effective the challenges were.

2.3.2 Automated Security Assessment of AWS environment

Experiments have been conducted to examine if it is possible to create a domain-specific modeling language for AWS [8]. The language is able to automatically

construct attack graphs that can be used by the user to assess the security with minimal experience in security. To validate the language, the domain-specific modeling language was tested on AWS environments deployed by CloudGoat. The result was compared to a similar open source assessment tool.

The results showed that the domain-specific language could identify vulnerabilities in the cloud environment, however, some assumptions had to be made to reach some goals. In some cases, the DSL could not complete the path, but the report points to the fact that the partial results can showcase the usefulness of the domain-specific language.

The results were only tested on artificial scenarios provided by CloudGoat. This meant that the report did not cover how the domain-specific language would perform in real-life scenarios. At the same time, the results show that the DSL could not identify some of the more advanced attacker methods.

2.3.3 Ethical hacking: Issues and Challenges

At the American University of Beirut, a survey about ethical hacking was conducted to see what consequences ethical hacking has. The paper discussed how the lack of security in most cases is caused by bad planning and coding configurations [19]. They also explored ethical hacking and the pen testing domain with the belief that employees need to constantly be educated in this area furthermore, the report argues that there need to be more ethical hackers. Methods like pen testing can help companies in solving security flaws.

The survey showed that ethical hacking and pen testing is needed and that more money needs to be put into ethical hacking. It's an evolving area that is needed to create secure internet. They also concluded that more resources is needed to allow ethical hackers to gain more knowledge about hacking.

Chapter 3

Method

To research our question, design science research was conducted. Design science research is done to find IT solutions for organizational problems [16]. This report used design science to find a solution to the lack of dynamic learning tools for cloud security.

3.1 Aim

To achieve the goal we decided that the hacking scenarios that were generated would all have the same end goal. This meant that the scenarios could have a goal of obtaining sensitive data but the attack path to gain access to the sensitive data could vary.

Having a chosen end goal would make it possible to generate working hacking scenarios with an obtainable goal but a different attack path. Hacking scenarios with different goals would be hard to generate because generating without a chosen goal could result in a hacking scenario with an unattainable goal.

3.2 Information gathering

To gain knowledge concerning what created a traversable hacking scenario we studied all the scenarios in CloudGoat. The idea of a modified AWS resource in a scenario that could result in a similar cloud environment with the same end goal came to mind. Each CloudGoat scenario covers different vulnerabilities in the cloud computing environment and has varying degrees of difficulty and size (size refers to the amount of

AWS resources set up). CloudGoat uses the scale easy, moderate, and hard to classify difficulty in a scenario. Size is divided into small, medium, and large.

The moderate difficulty and small size scenario cloud_breach_S3 was chosen as the scenario to base our research on. We decided that a moderate difficulty and small size would be a good starting point because it limits the amount of AWS resources that can be modified.

3.3 Utility

This project used the tools AWS, Terraform, and python to generate cloud environments. AWS was chosen as the cloud provider for this project because many companies are moving their computational resources to AWS. The AWS cloud environment was built with the tool Terraform. Terraform is a command line interface that can deploy cloud environments. It had support for multiple cloud providers which made it a good deployment tool for this project. Python can be used to write small scripts which made it a good pick for this project.

3.4 Modifying Cloud_Breach_S3

Cloud_breach_s3 acted as a base starting point for this project (section CloudGoat scenarios). The different AWS resources in the Cloud_breach_s3 scenario were modified to achieve a different hacking scenario. If the hacking scenario was traversable the modified scenario was saved otherwise, another AWS resource was modified (Figure 3.4.1).

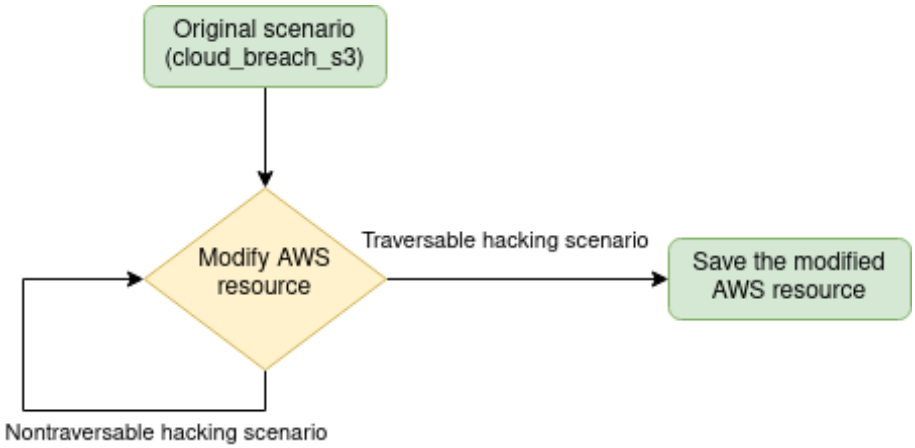

Figure 3.4.1: Flowchart of the workflow used to generate new scenarios

3.5 Verification

To verify the generated cloud environments the cloud environments were set up with Terraform and the hacking scenario was manually traversed. Each hacking challenges had a different attack path to achieve the goal and each attack path was traversed to verify that the hacking challenge was possible.

Chapter 4

Result

The AWS cloud environment was built by combining the predefined AWS resources VPC, EC2, S3, and IAM. If the AWS resources were not predefined, the generation would be unpredictable and could result in an unsolvable hacking scenario. The best approach to achieve a generated solvable hacking scenario was to create an AWS cloud environment from predefined building blocks. In this case, the building blocks were the different AWS resources configured in different ways (Appendix 2 Terraform file defining an AWS resource).

A python script was used to combine multiple Terraform files that each defined a single AWS resource into a complete Terraform file that could be deployed. Terraform files were created where each contained one AWS resource. For example, one Terraform file defined the EC2 instance while another defined the S3 bucket.

This enabled the deployment tool to generate a cloud environment from building blocks made up of Terraform files that declared AWS resources. Terraform files that described the same AWS resources with different configurations were picked randomly which resulted in different cloud environments being generated each time. This means that one of two IAM AWS resources that are defined in two different Terraform files is randomly picked. The python script will always have one of each AWS resource needed to create a working hacking challenge which ensures that the cloud environment is traversable. When the python script had built a Terraform file, Terraform could be executed to deploy the cloud environment (Figure 4.1).

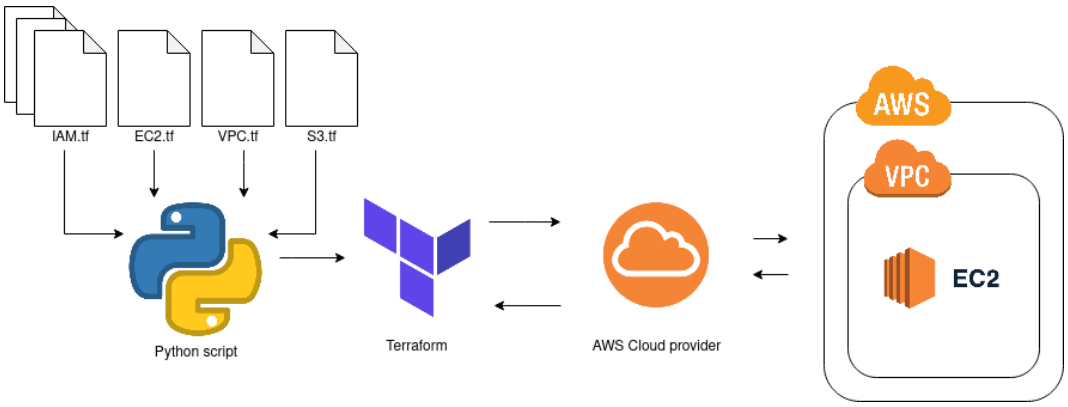

Figure 4.0.1: Diagram showcasing how the tools used in this project worked together to generate a cloud environment

4.1 Scenarios

The IAM instance profile in this scenario played a crucial role in achieving the scenario goal which made it the AWS resource that this project focused on to achieve generated scenarios. Therefore the first modification was to the privileges of the IAM instance profile. The IAM instance profile gained different privileges which did result in a new scenario, however, the IAM instance profile did not have access to the S3 bucket. This meant that the attacker did not have a way to gain S3 access privileges which meant that only modifying privileges is not enough.

One way to achieve higher IAM privileges is through exploitation. The scenario iam_privesc_by_rollback presented a way to gain higher privileges through a privilege escalation exploit. The IAM instance profile in cloud_breach_s3 was modified to also contain a privilege escalation exploit. An IAM instance profile with S3 access privilege could still be achieved but with a different attack path.

Once the first modified scenario was confirmed to be traversable the IAM AWS resource was modified again. This time an additional new IAM user was added as well as the IAM role connected to the EC2 profile was modified. The attack path of the generated scenarios is presented in Figures 4.2.1-4.2.3.

4.2 Verification

The first scenario is the same as the scenario cloud_breach_s3 (Figure 4.1). cloud_breach_s3 has explained in section 2.1.1 above.

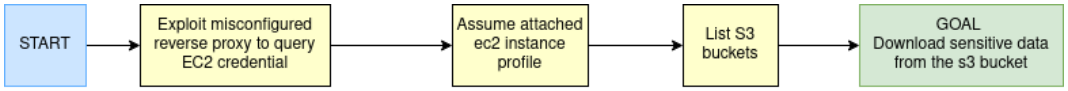

Figure 4.2.1: Attack graph of the first scenario that is identical to Cloud_breach_s3

The second scenario plays out like the cloud_breach_s3 scenario but includes a privileges escalation exploit (figure 4.2.1). Once an attacker has accessed the IAM instance profile a privileges escalation exploit can be performed to gain higher privileges. By analyzing the privileges of the IAM instance profile the attacker can discover the privilege SetDefaultPolicyVersion. This privilege enables the attacker to set an old policy version as the default policy version (the policy version decides the privileges of a IAM profile). This scenario contains one old policy with full admin privileges. The attacker can use the SetDefaultPolicyVersion to set the policy version with full admin privileges as default which gives the attacker full admin access. After performing the privilege escalation exploit the scenario plays out the same as cloud_breach_s3.

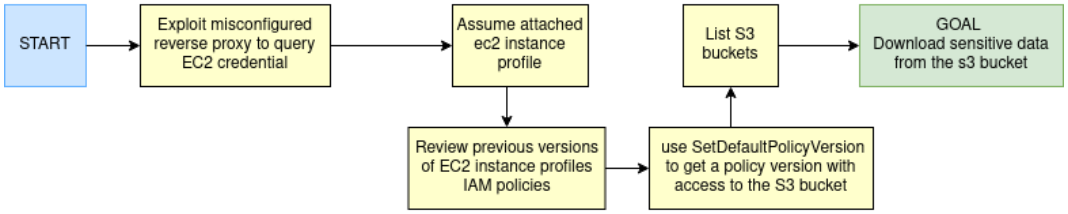

Figure 4.2.2: Attack graph of the second scenario that features a privileges escalation exploit

The last scenario contains an attack graph that is completely different from the cloud_breach_s3 (Figure 4.2.3). The commonly used path of accessing the IAM instance profile is a dead end which means that another path needs to be taken. This scenario contains an IAM user in combination with an IAM role with a full admin policy. The attacker can access the IAM user and then swap the IAM role with full admin policy onto the existing IAM instance profile. From here the attacker can create a new EC2 key pair which is used in the creation of a new EC2 instance. By attaching the IAM instance profile with full admin access to the EC2 instance, the attacker can SSH into the EC2 to use AWS CLI commands. Once an EC2 instance with full admin access has been created, the attacker can list the S3 buckets and then download the sensitive data.

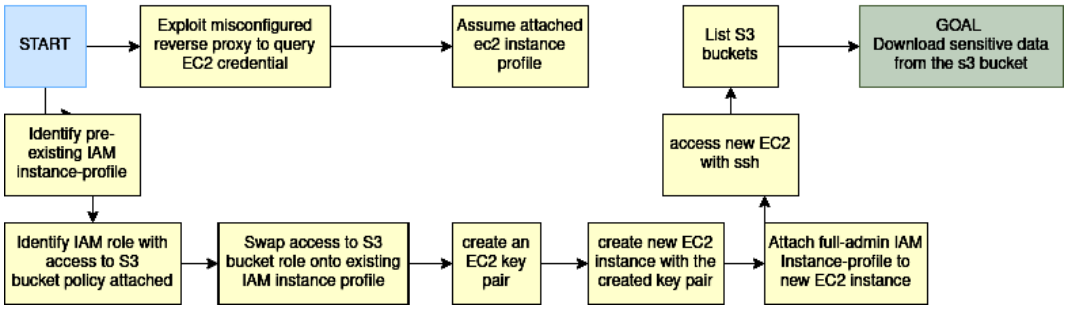

Figure 4.2.3: Attack graph of the third scenario that features a different attack path

Chapter 5

Discussion

5.1 Learning by hacking

The newly generated hacking challenges presented in the result are verified to have an achievable goal. This means that they are traversable and can be used in a learning environment to gain more knowledge about cloud security. If used by cloud customers they can increase their knowledge concerning vulnerabilities in a cloud environment and what the consequences of a misconfigured environment are. By using the generated hacking challenges presented in the result as a learning experience, cloud customers can create cloud environments with fever vulnerabilities. This in turn increases cloud security and lowers the chance of a data breach.

Hacking challenges are a more effective way of learning cloud security than reading about the subject since it gives hands-on learning. A hands-on learning experience is more effective than traditional learning (Section Introduction). Furthermore, the study mentioned in section 2.3.1 concluded that the participants felt that they knew more about hacking after participating in the event. By using the hacking challenges presented in this report in a similar manner, cloud computing customers would have a better knowledge surrounding cloud security.

These hacking challenges are used as a good learning experience, however, malicious actors could also use them to learn about potential vulnerabilities that can be abused in malicious ways. This points to the fact that the result of this project could give the opposite effect of what it was constructed to do. This drawback needs to be considered, however, if cloud customers use these challenges to learn how to correctly set up an

environment without vulnerabilities the drawback is negated. A study made by the US Military Academy concluded that hacking competitions can be used for educational purposes to learn about security if the correct ethical tone is set [11]. Furthermore, these challenges could help educate ethical hackers. Ethical hackers could learn about flaws by hacking cloud environments which could lead to these flaws being removed (Section 2.3.3). This means that hacking challenges can have an overall good effect if used with good intent.

If more people know how to correctly construct a cloud environment more companies would feel safe to change over to the cloud. Which would make companies use the exact amount of resources they need because of the 'payperuse' policy (Section 2.1.1). This is more sustainable digitization than companies buying big data centers with more computing resources than they need.

5.1.1 Limitations

This report focused on generating new scenarios by modifying one hacking challenge and one AWS resource, however, multiple scenarios and multiple AWS resources could be modified to create more generated scenarios. The results of this report show that a cloud hacking challenge can be generated, however, this report only generated two new scenarios. The hacking community is always evolving with new hacking methods and for the defenders to keep up multiple hacking scenarios need to be generated. The results of this project work as a proof of concept, however, more could be done in this area to generate more hacking challenges. This project presents the generation of cloud environments by combining building blocks. By creating more of these building blocks we would be able to generate more challenges.

The verification of the results in this project was done by traversing the hacking challenges and achieving the end goal, however, this is a slow verification process. Traversing the hacking challenge means that the attacker needs to understand the hacking scenario. To avoid having to understand each generated hacking challenge to verify them, a meta attack language tool could be used to verify them (Section 2.4.2). This could lead to an automated security assessment of each generated hacking scenario which would make the verification easier. This was not done in this report, however, future work in this area could verify generated scenarios using a meta attack language tool.

Chapter 6

Conclusion

The purpose of this report was to investigate how traversable cloud hacking scenarios can be generated. By deploying cloud environments using several tools, two new hacking challenges with an achievable goal were generated. These challenges have the possibility to educate people on how to set up a secure cloud environment, however, it is limited by how few hacking challenges were created.

6.1 Future work

We only changed one of the AWS resources to generate the new challenges. To improve this project further, several versions of the other resources could be created to see if it generates more functional challenges. The verification was done by traversing the hacking challenges. To expand this project further a verification tool could be used, for example, a meta attack language

Bibliography

[1] *Amazon Inspector*. https://aws.amazon.com/inspector/. Accessed: 2022-03-21.

[2] Brook, JM, Getsin, A, Jensen, G, Jameson, L, Roza, M, Thethi, N, Kurmi, A, Levy, S, Shamban, S, Hargrave, V, et al. "Top Threats to Cloud Computing: The Egregious Eleven". In: *Cloud Security Alliance* (2019).

[3] Chothia, Tom and Novakovic, Chris. "An Offline Capture The {Flag-Style} Virtual Machine and an Assessment of Its Value for Cybersecurity Education". In: *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)*. 2015.

[4] *Cloud computing with AWS*. https://aws.amazon.com/what-is-aws/. Accessed: 2022-03-21.

[5] *CloudGoat*. https://github.com/RhinoSecurityLabs/cloudgoat. Accessed: 2022-03-21.

[6] *Data Breach: Millions of Senior Citizens' Names, Emails, and Phone Numbers Compromised in Senior Care Review Website Database Breach*. https://www.wizcase.com/blog/senioradvisor-breach-report/h. Accessed: 2022-05-12.

[7] Edman, Johan and Ågren, Wilhelm. *Legal and Security Issues of Data Processing when Implementing IoT Solutions in Apartments*. 2020.

[8] Engström, Viktor, Johanson, Pontus, Ringdahl, Erik, Wällstedt, Max, and Lagerstöm, Robert. "Automated Security Assessments of AmazonWeb Service Environmen". In: (2022).

[9] *Ermetic finds majority of AWS accounts surveyed are vulnerable to ransomware due to misconfigurations*. https://ermetic.com/news/ermetic-finds-majority-of-aws-accounts-surveyed-are-vulnerable-to-ransomware/. Accessed: 2022-03-21.

[10] Freeman, Scott, Eddy, Sarah L, McDonough, Miles, Smith, Michelle K, Okoroafor, Nnadozie, Jordt, Hannah, and Wenderoth, Mary Pat. “Active learning increases student performance in science, engineering, and mathematics”. In: *Proceedings of the national academy of sciences* 111.23 (2014), pp. 8410–8415.

[11] Irvine, Cynthia. “hacking Competitions and their untapped Potential for security education”. In: ().

[12] *Is the Cloud Secure?* `https://www.gartner.com/smarterwithgartner/is-the-cloud-secure`. Accessed: 2022-03-21.

[13] Lisdorf, Anders. *Cloud Computing Basics: A Non-Technical Introduction*. 2021.

[14] *MAL (the Meta Attack Language)*. `https://www.kth.se/cs/nse/research/software-systems-architecture-and-security/projects/mal-the-meta-attack-language-1.922174`. Accessed: 2022-03-21.

[15] Manvi, Sunilkumar S and Shyam, Gopal Krishna. “Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey”. In: *Journal of network and computer applications* 41 (2014), pp. 424–440.

[16] Peffers, Ken, Tuunanen, Tuure, Rothenberger, Marcus A, and Chatterjee, Samir. “A design science research methodology for information systems research”. In: *Journal of management information systems* 24.3 (2007), pp. 45–77.

[17] Rose, Karen, Eldridge, Scott, and Chapin, Lyman. “The internet of things: An overview”. In: *The internet society (ISOC)* 80 (2015), pp. 1–50.

[18] Sunyaev, Ali. *Internet computing: Principles of Distributed systems and emerging internet-based technologies*. Springer Nature, 2020.

[19] Yaacoub, Jean-Paul A, Noura, Hassan N, Salman, Ola, and Chehab, Ali. “A Survey on Ethical Hacking: Issues and Challenges”. In: *arXiv preprint arXiv:2103.15072* (2021).

Appendix A

Python Script

Python script to generate cloud scenarios

```
import os
import random
# assign directory
directory = 'terraform'
newFileContent = ''
# iterate over files in
# that directory
open('Combined.tf', 'w').close()
for filename in os.listdir(directory):
    f = os.path.join(directory, filename)
    # checking if it is a file
    if os.path.isdir(f):
        list = os.listdir(f)
        randomChoice = random.choice(list)
        while os.path.isdir(os.path.join(f,randomChoice)):
            randomChoice = random.choice(list)

        f = os.path.join(f,randomChoice)
        f = open(f, "r")
        newFileContent = newFileContent +'\n'+ f.read()
        f.close()
    elif os.path.isfile(f):
```

```
        f = open(f, "r")
        newFileContent = newFileContent +'\n'+ f.read()
        f.close()
f = open("Combined.tf", "a")
f.write(newFileContent)
f.close()
```

Appendix B

IAM resources

Terrafrom file defining the IAM resource

```
#IAM Role
resource "aws_iam_role" "cg-banking-WAF-Role" {
  name = "cg-banking-WAF-Role-${var.cgid}"
  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
  tags = {
      Name = "cg-banking-WAF-Role-${var.cgid}"
      Stack = "${var.stack-name}"
      Scenario = "${var.scenario-name}"
```

```
  }
}

resource "aws_iam_policy" "cg-banking-WAF-Role-policy" {
  name        = "cg-banking-WAF-Role-policy-${var.cgid}"
  description = "cg-banking-WAF-Role-policy"
  policy      = "${file("./terraform/IAM/policies/v1.json")}"
}

#IAM Role Policy Attachment
resource "aws_iam_role_policy_attachment"
        "cg-banking-WAF-Role-policy-attachment-s3" {
  role = "${aws_iam_role.cg-banking-WAF-Role.name}"
  policy_arn = "${aws_iam_policy.cg-banking-WAF-Role-policy.arn}"
}

#IAM Instance Profile
resource "aws_iam_instance_profile" "cg-ec2-instance-profile" {
  name = "cg-ec2-instance-profile-${var.cgid}"
  role = "${aws_iam_role.cg-banking-WAF-Role.name}"
}

resource "null_resource" "cg-create-iam-user-policy-fullAccess" {
  provisioner "local-exec" {
      command = "aws iam create-policy-version --policy-arn
      ${aws_iam_policy.cg-banking-WAF-Role-policy.arn}
      --policy-document file://./terraform/IAM/policies/v5.json
      --no-set-as-default --profile ${var.profile}
      --region ${var.region}"
  }
}
```