Two threat patterns that exploit "Cloud Based Leaky Buckets" and "Malware Laden Buckets"

ROHINI SULATYCKI, Dun & Bradstreet
EDUARDO B. FERNANDEZ, Florida Atlantic University

We present two threat patterns that describe attacks involving cloud bucket based storage. These patterns provide insight into the complexity involved in securing cloud buckets and the ease with which these buckets can start "leaking" sensitive data. We also provide insight on the misuse of cloud buckets by attackers to distribute malware to unsuspecting victims. Threat patterns describe how a threat is realized from the point of view of the attacker, indicating how the threat can be stopped or mitigated.

Categories and Subject Descriptors: D.2.11 **[Software Engineering]** Software Architecture – Patterns; D.5.1 D.4.6 **[Security]**

General Terms: Design

Key Words and Phrases: Misuse patterns, security patterns, application security, vulnerabilities, exploits, cloud, buckets and sensitive data

1. INTRODUCTION

The National Institute of Standards and Technology (NIST) defines five essential characteristics of cloud computing: on-demand self-service, broad network access, resource pooling, rapid elasticity or expansion and measured service. Cloud based storage enables on-demand access to a shared pool of configurable storage that can be rapidly provisioned and accessed with minimal management effort or service provider interaction. Users can access cloud storage from any location or device with access to the internet.

The Cloud storage ecosystem is a complex mix of virtualization, management interfaces, web services and workflows. Most cloud storage has an application programming interfaces (APIs) that are protected by API keys. Securing this environment is a difficult endeavor requiring configuring policies and access control lists and any security misconfiguration can lead to "leaky buckets" or storage that inadvertently leaks data. Additionally, cloud storage itself might be used to deliver malware to victims due to lack of anti virus scanning. In this paper we examine how this complex ecosystem gives rise to security issues.

To design secure cloud storage , we first need to understand possible threats to the storage (Braz et al, 2008). To do this, we apply *misuse patterns* (Fernandez et al. 2007 and 2009), to describe how a misuse is performed; some misuses can be produced as a result of different threats and we extended this idea to define *threat patterns* (Uzunov and Fernandez 2013), which describe specific generic threats. A misuse/threat pattern describes how a misuse/threat is performed from the point of view of the attacker. It defines the environment where the attack is performed, countermeasures to stop it, and provides forensic information in order to trace the attack once it happens. These patterns are useful for developers because once they determine that a possible attack can happen in the

Authors' addresses: Rohini Sulatycki (corresponding author), Dept. of Computer and Electrical Eng. and Computer Science, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL33431, USA; email: rohini.sulaticky@gmail.com; Eduardo B. Fernandez, Dept. of Computer and Electrical Eng. and Computer Science, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL33431, USA, email: ed@cse.fau.edu.

A preliminary version of this paper was presented in a writers' workshop at the 12th Latin American Conference on Pattern Languages of Programs (SLPLoP). SLPLoP'18, November 20-23, Valparaíso, Chile. Copyright 2018 is held by the author(s). HILLSIDE 978-1-941652-11-4

environment, a corresponding misuse pattern will indicate what security mechanisms are needed as countermeasure. Also, misuse patterns can be very useful for forensic examiners to determine how a misuse is performed, and where they can find useful evidence information after an attack. Note that the misuse pattern describes a complete misuse, e.g. stealing information from a database, not just specific steps used to perform the misuse, such as SQL injection or buffer overflow; we consider these threats, which would be described by threat patterns. Misuses do not provide aspects to be avoided when building the system, so they are not antipatterns although they have common aspects (An antipattern indicates practices that should be avoided (Mowbray and Hudson, 1998)).

Misuse patterns include the steps leading to misuses so several misuse patterns may be derived from one threat pattern, e.g., SQL injection could lead to data leakage, data destruction, or DoS. Threat patterns take advantage of specific vulnerabilities and can be described with respect to the corresponding vulnerability. We present a pattern that takes advantage of leaky buckets and a pattern that takes advantage of malware-laden buckets.

Section 2 presents a template used to describe misuse/threat patterns (Fernandez, 2013). In Section 3, we present a pattern that takes advantage of cloud-based leaky buckets to reach unauthorized data, while in Section 4 we show a pattern that uses malware-ladden buckets to attack other sites. Iin Section 5 we offer some conclusions and discuss possible future work.

2. TEMPLATE FOR MISUSE/ THREAT PATTERNS

2.1 Name

The name of the pattern should correspond to the generic name given to the specific type of misuse in standard attack repositories.

2.2 Intent or thumbnail description

A short description of the intended purpose of the pattern (what problem it solves for an attacker).

2.3 Context

It describes the generic environment including the conditions under which the attack may occur. This may include minimal defenses present in the system as well as typical vulnerabilities of the system.

2.4 Problem

From an attacker's perspective, the problem is how to find a way to attack the system. The forces indicate what factors may be required in order to accomplish the attack and in what way; for example, which vulnerabilities can be exploited.

2.5 Solution

This section describes the solution of the attacker's problem, i.e., how the attack can reach its objectives and the expected results of the attack. UML class diagrams show the system under attack. Sequence or collaboration diagrams show the exchange of messages needed to accomplish the attack.

2.6 Known uses

Specific incidents where this attack occurred are preferred but for new vulnerabilities, where an attack has not yet occurred, specific contexts where the potential attack may occur are enough.

2.7 Consequences

Discusses the benefits and drawbacks of a misuse pattern from the attacker's viewpoint. The enumeration includes good and bad aspects and should match the forces.

2.8 Countermeasures and Forensics

It describes the security measures necessary in order to stop, mitigate, or trace this type of attack. This implies an enumeration of which security patterns are effective against this attack. From a forensic viewpoint, it describes what information can be obtained at each stage tracing back the attack and what can be deduced from this data in order to identify this specific attack.

2.9 Related Patterns

Discusses other misuse patterns with different objectives but performed in a similar way or with similar objectives but performed in a different way.

3. Taking advantage of Cloud Based Leaky Buckets

3.1 Intent

An attacker may be able to take advantage of leaky buckets to view, delete or modify sensitive data. Depending on the data obtained, the attacker might be able to elevate their privilege and access other systems.

Leaky buckets are defined as cloud based storage whose contents become accessible to a wider audience than intended.

3.2 Context

One of the advantages of cloud storage is that it can be rapidly provisioned and accessed with minimal management effort or service provider interaction. However, this means that a number of users are able to use the storage without necessarily understanding the complexity of the environment. In this environment it is easy for a user to introduce a security misconfiguration by allowing public access to their storage. Data within cloud storage and the storage itself might become vulnerable if credentials such as API keys get compromised.

3.3 Problem

Attacks can be performed by taking advantage of the following vulnerabilities within the cloud storage environment:

- Access Control Policies may not be understood by users. Even if they are fine grained users do no apply them properly and allow too much access to their data.
- Users uploading data to cloud storage may not be aware of the sensitive nature of the data and therefore might not apply the correct security controls
- Cloud storage systems are often provisioned programmatically and best practices require using a source code control system that can be cloud based. Often API access keys are uploaded to source code repositories. Bots have been created that scan cloud based source code control systems for access keys.
- Default or weak passwords are enabled which can be used to compromise systems within and potentially outside the cloud storage environment.
- The emergence of vulnerability markets (Böhme, R. 2005) provides an economic incentive for researchers to search for and to disclose information on vulnerabilities. These include vulnerability disclosures related to security misconfigurations.

3.4 Solution

An attacker can identify leaky buckets and take advantage of these vulnerabilities to threaten the data and/or other systems.

3.4.1 Structure

Figure 1 shows a class diagram for compromising leaky buckets. A **User** requests the **Provider** for access to **Cloud Storage** and uses the **Management Interface** to configure the storage. The user can also obtain access keys to access the cloud storage **Account** programmatically. The Attacker operates on the **Cloud Storage.**

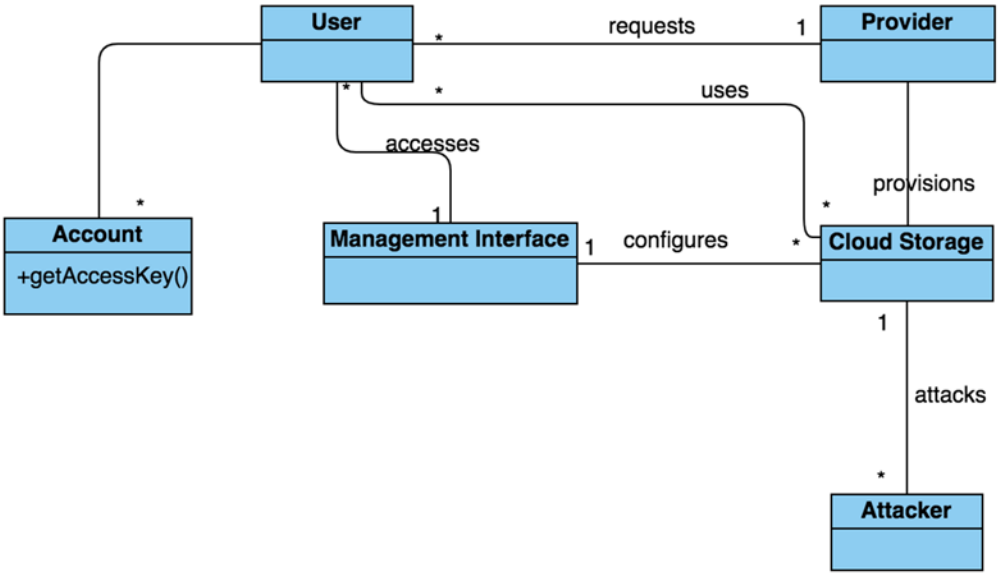

Figure 1. Class Diagram for Leaky Buckets Misuse Pattern

3.4.2 Dynamics

The following use case describes threats that take advantage of leaky buckets.

UC1: Compromising cloud storage with security misconfigurations (Fig. 2)

Summary: The Attacker compromises a leaky bucket whose security is misconfigured.

Actor: Attacker

Precondition: The Attacker must have access to the leaky bucket.

Description:

- The attacker scans cloud storage and finds the leaky bucket.
- The attacker then misuses the leaky bucket to compromise the data stored in the bucket.

Postcondition:
The data is compromised.

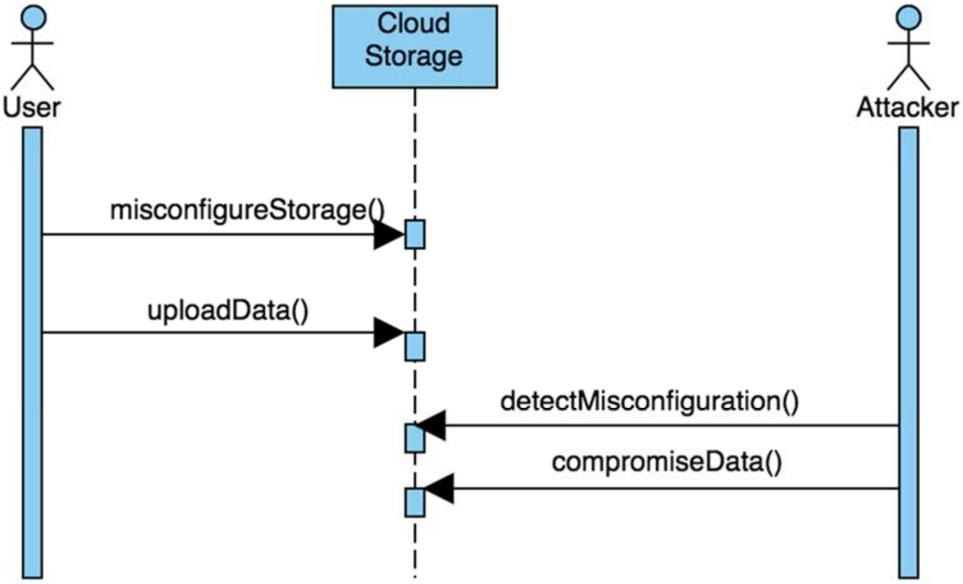

Figure 2. Sequence Diagram for the use case *Compromising cloud storage with security misconfigurations*

UC2: Compromising cloud storage due to exposure of access keys (Fig. 3)

Summary: The Attacker uses exposed access keys to compromise cloud storage

Actor: Attacker

Precondition: The Attacker must have access to the exposed access keys.

Description:

- The attacker finds exposed access keys.
- The attacker then misuses the access keys to either compromise the data, destroy the data or edit the data.

Postcondition:
The data is compromised.

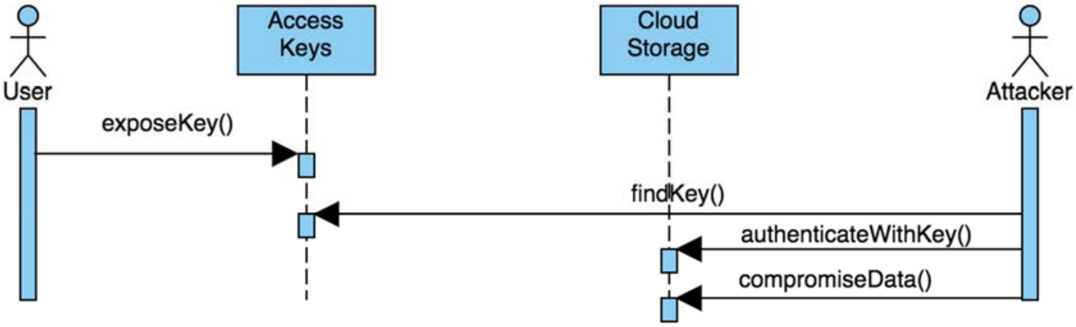

Figure 3. Sequence Diagram for the use case *Compromising cloud storage due to exposure of access keys*

3.5 Known uses (incidents)

Some incidents where these patterns (apparently) were used are:

- In 2014, Code Spaces, a code hosting service was forced to close shop after an unauthorized user was able to gain access to their AWS Console and deleted most of their data, backups, machine configurations and offsite backups (Arstechnica, June 2014)
- In Jan 2015, a developer woke up with a bill for $2375 after he accidentally published his Amazon S3 cloud storage keys to his Github account. A bot found the keys, spun up about 140 EC2 instances and started mining Bitcoin (The Register, Jan 6, 2015).
- In July 2017, it was discovered that Sweden leaked every car owners' details, including military and police personnel when it awarded IBM a contract to manager its databases and networks. The databases pushed to the IBM cloud covered every vehicle in the country including police and military registrations, plus details of individuals on witness protection programs. Details on members of the military were included in the databases. IBM workers outside Sweden then had access to this sensitive information without proper security clearance (The Register, July 23, 2017).
- In Sept 2017, Records of roughly four million Time Warner Cable customers in the US were exposed to the public internet after a contractor failed to properly secure an Amazon cloud database (The Register, Sept 5, 2017).
- In September 6th, 2017, a security researcher discovered three Amazon Web Services S3 cloud storage buckets configured to allow any AWS global authenticated user to browse and download the contents. These buckets named “centcom-backup,” “centcom-archive,” and “pacom-archive” were thought to belong to the US Central Command, based in Tampa, Fla. and responsible for US military operations from East Africa to Central Asia, including the Iraq and Afghan Wars. PACOM is the US Pacific Command, headquartered in Aiea, HI and covering East, South, and Southeast Asia, as well as Australia and Pacific Oceania. The data obtained from these buckets indicated that the organisation has been collecting social media posts from web users across the globe for eight or so years (Upguard, Sept 6, 2017).
- In Sept 2017, security researchers found a number of sensitive documents belonging to Verizon Wireless in a publicly accessible AWS S3 bucket. This data included usernames, passwords, router information and server information (The Register, Sept 22, 2017).
- In June 2018, researchers from Laceworks revealed that 22000 container orchestration and API management systems were publicly available on the internet. This means that attackers

could remotely access the infrastructure to install, remove or encrypt any application that the company was running in the cloud (Threatpost, June 2018).

3.6 Consequences

Some of the benefits of the misuse pattern are the following:

- An attacker can obtain data that he can either use or sell.
- An attacker might be able to edit or delete the data.
- The attacker might be able to extend the attack to other systems and obtain more confidential information.

Possible sources of failure for the attacker include:

- There might be additional security controls in place such as encryption that might prevent the attacker from getting access to the data
- The compromised system might be sufficiently firewalled to prevent the attacker from accessing other systems. However, in this case the data itself might still be compromised.

3.7 Countermeasures and Forensics

Leaky buckets can be stopped by the following countermeasures:

- Provide secure defaults to keep user data and infrastructure secure.
- Layer additional security controls by default such as default encryption of all cloud storage data.
- Require Multi-factor authentication for all cloud storage credentials.
- Derive meta data as files are uploaded and suggest security mechanisms based on the meta data.
- Only provide temporary access keys which can be easily revoked in case of compromise.
- Use security patterns against cloud storage. Security design patterns are descriptions or templates describing a general solution to a security problem that can be applied in many different situations. Rather than focus on the implementation of specific security mechanisms, security design patterns are meant to eliminate the accidental insertion of vulnerabilities into code or to mitigate the consequences of vulnerabilities.
- Design applications using appropriate security methodologies (Uzunov et al. 2012).

3.8 Related Patterns

- Misuse patterns for cloud computing: Malicious virtual machine creation is possible due to poor security configuration (K Hashizume et al. 2011).
- Two threat patterns that exploit "security misconfiguration" and "sensitive data exposure" vulnerabilities (Sulatycki and Fernandez, 2015). Those patterns have some common aspects with the ones presented here but are more general, they do not use leaky buckets.

4. Taking advantage of Malware Laden Buckets

4.1 Intent

An attacker may be able to upload malware to cloud buckets and distribute it to unsuspecting victims. Malware once downloaded to the victims machines can be used to destroy data or as ransomware.

4.2 Context

Cloud buckets can often be rapidly provisioned and accessed with minimal management effort or service provider interaction. However, this means that there is no governance of or oversight into the data that can be uploaded to cloud buckets. If the cloud provider does not provide anti virus scanning cloud buckets can be used by attackers to distribute malware.

4.3 Problem

Attacks can be performed by taking advantage of the following vulnerabilities within the cloud storage environment:

- A number of cloud providers do not provide anti-virus scanning of data uploaded to or downloaded from cloud buckets.
- Most cloud buckets provide links to allow users to download data
- The emergence of vulnerability markets (Böhme, R. 2005) provides an economic incentive for researchers to search for and to disclose information on vulnerabilities. These include vulnerability disclosures related to security misconfigurations.

4.4 Solution

An attacker can upload malware to cloud buckets and take advantage of lack of anti virus scanning to threaten other systems.

4.4.1 Structure

Figure 1 shows a class diagram for using cloud storage to distribute malware. An **Attacker** requests the **Provider** for access to **Cloud Storage**. The attacker then provisions the cloud storage for uploads, obtains **Malware** and uploads the malware to the cloud storage.

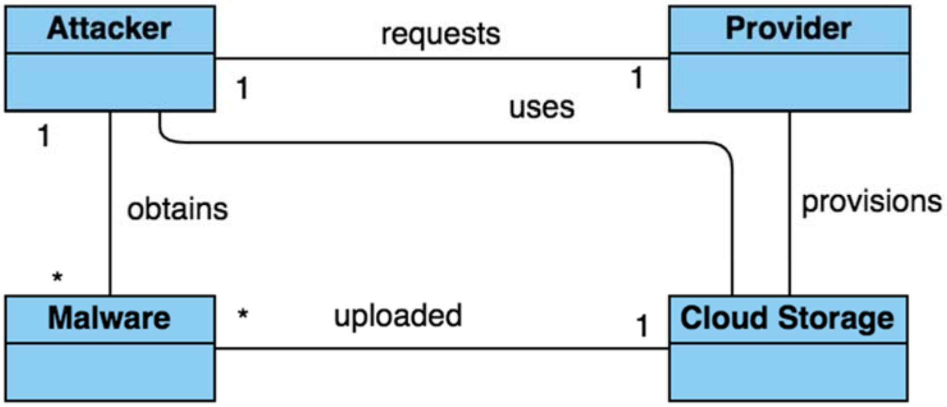

Figure 4. Class Diagram for Malware Laden Buckets Misuse Pattern

UC1: Using cloud storage used for delivering malware (Fig. 5)

Summary: The Attacker uses cloud storage to deliver malware

Actor: Attacker

Precondition: The Attacker must have access to the cloud storage. Files are not being scanned sufficiently enough to detect the malware

Description:

- The attacker uploads malware to cloud storage.
- The attacker then sends out links to the malware to the victims
- The victim clicks on the link, downloads the malware and the victims system becomes infected

Postcondition:
The victims system is compromised.

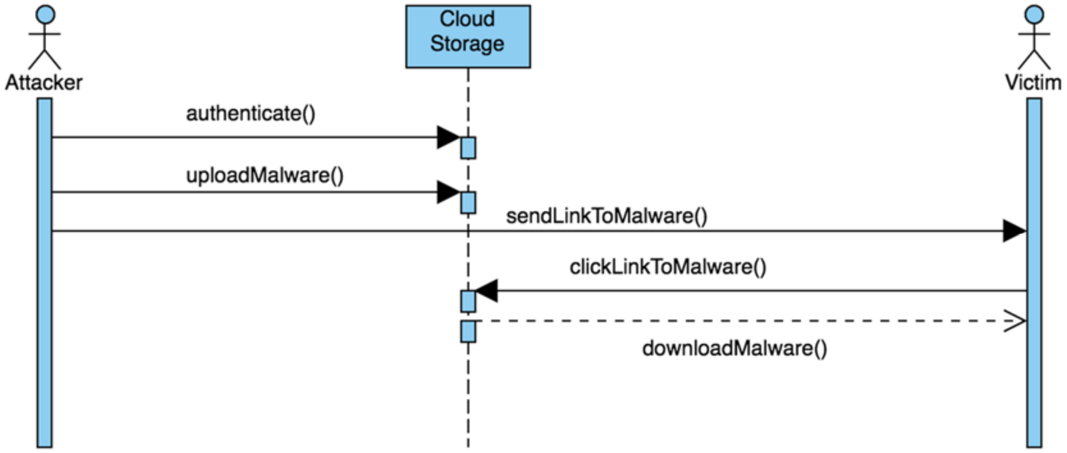

Figure 5. Sequence Diagram for the use case *Using cloud storage to deliver malware*

4.5 Known uses (incidents)

Some incidents where these patterns (apparently) were used are:

- In June 2016, it was found that innovative attackers were posing as job applicants and sending Dropbox links to their victims claiming that their resume were too large to email. Victims clinked on the links and downloaded ransomware called Petya that then proceeded to infect their systems by locking the screen and encrypting their files.[Player One, June 2016]

4.6 Consequences

Some of the benefits of the misuse pattern are the following:

- An attacker can get monetary advantage by using malware as ransomware.
- An attacker might be able to destroy data.
- Depending on the type of malware used, the attacker might be able to expand the attack to other systems.

Possible sources of failure for the attacker include:

- The cloud storage might have implemented malware scanning preventing malware from being uploaded.
- The victims systems might have malware scanning preventing the malware from infecting their system.

4.7 Countermeasures and Forensics

Malware laden buckets can be stopped by the following countermeasures:

- Require scanning of files uploaded to cloud storage and downloaded from cloud storage
- Derive meta data as files are uploaded and suggest security mechanisms based on the meta data.
- Use security patterns against cloud storage. Security design patterns are descriptions or templates describing a general solution to a security problem that can be applied in many different situations. Rather than focus on the implementation of specific security mechanisms, security design patterns are meant to eliminate the accidental insertion of vulnerabilities into code or to mitigate the consequences of vulnerabilities.
- Design applications using appropriate security methodologies (Uzunov et al. 2012).

4.8 Related Patterns

- Misuse patterns for cloud computing: Malicious virtual machine creation is possible due to poor security configuration (K Hashizume et al. 2011).

5. DISCUSSION AND CONCLUSIONS

Designers need to first understand possible threats before designing secure systems. However, identifying threats is not enough; we need to understand how a whole misuse is performed by taking advantage of them. Threat and Misuse patterns appear to be a good tool to understand how misuses are performed. It is possible to build a relatively complete catalog of threats and misuse patterns. Having such a catalog we can analyze a specific application and evaluate its degree of resistance to these misuses. The architecture (existing or under construction) must have a way to prevent or at least mitigate all the threats that apply to it. When potential customers use an application they must have assurance on what threat/misuses the application is able to prevent. Many providers do not want to show their security architectures; showing their list of the misuse patterns they can handle would give them a way to prove a degree of resistance to misuses without having to show their security details. System components which appear in more than one misuse pattern require special security handling.

We illustrated our ideas with two specific patterns. We are continuing to develop misuse patterns for cloud infrastructure in order to create a relatively complete catalog for it that can be used by developers. Finally, we intend to incorporate these patterns into the secure systems design methodology described in (Fernandez13).

REFERENCES

Arstechnica, https://arstechnica.com/information-technology/2014/06/aws-console-breach-leads-to-demise-of-service-with-proven-backup-plan/

R. Böhme, Vulnerability markets. What is the economic value of a zero-day exploit? Proceedings of 22nd Chaos Communication Congress, (Berlin, Germany, dec. 27-30, 2005).

F. Braz, E.B.Fernandez, and M. VanHilst, "Eliciting security requirements through misuse activities" Procs. of the 2nd Int. Workshop on Secure Systems Methodologies using Patterns (SPattern'07). 2008.

CVE, Common Vulnerabilities and Exposures, https://cve.mitre.org/

E.B. Fernandez, J.C. Pelaez, and M.M. Larrondo-Petrie, "Attack patterns: A new forensic and design tool", Procs. of the Third Annual IFIP WG 11.9 Int. Conf. on Digital Forensics, Orlando, FL, Jan. 29-31, 2007.
Chapter 24 in Advances in Digital Forensics III, P. Craiger and S. Shenoi (Eds.), Springer/IFIP, 2007, 345-357.

E.B. Fernandez, N. Yoshioka, and H. Washizaki, "Modeling misuse patterns", 4th Int. Workshop on Dependability Aspects of Data Warehousing and Mining Applications (DAWAM 2009), in conjunction with the 4th Int.Conf. on Availability, Reliability, and Security (ARES 2009). March 16-19, 2009, Fukuoka, Japan

E.B.Fernandez, “Security patterns in practice: Building secure architectures using software patterns”. Wiley Series on Software Design Patterns. 2013

K Hashizume, E. B. Fernandez, and N. Yoshioka, "Misuse patterns for cloud computing: Malicious virtual machine creation", Procs. of the Twenty-Third International Conference on Software Engineering and Knowledge Engineering (SEKE 2011), Miami Beach, USA, July 7-9, 2011

T.J. Mowbray, T.Hudson, eds. AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis. John Wiley & Sons, ltd. ISBN 978-0-471-19713-3, 1998.

OWASP, “The Ten Most Critical Web Application Security Risks.” 2010.
https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

Player One, http://www.player.one/petya-ransomware-locks-your-computer-screen-then-encrypts-files-how-remove-file-521931

The Register, http://www.theregister.co.uk/2015/01/06/dev_blunder_shows_github_crawling_with_keyslurping_bots/

The Register,
https://www.theregister.co.uk/2017/09/22/verizon_falls_for_the_old_unguarded_aws_s3_bucket_trick_exposes_internal_system/

The Register,
https://www.theregister.co.uk/2017/07/23/sweden_leaked_every_car_owners_details_last_year_then_tried_to_hush_it_up/

The Register, https://www.theregister.co.uk/2017/09/05/twc_loses_4m_customer_records/

Threatpost, https://threatpost.com/22k-open-vulnerable-containers-found-exposed-on-the-net/132898/

Rohini Sulatycki , Eduardo B. Fernandez, Two threat patterns that exploit "security misconfiguration" and "sensitive data exposure" vulnerabilities, Proceedings of the 20th European Conference on Pattern Languages of Programs, p.1-11, July 08-12, 2015, Kaufbeuren, Germany

Symantec,
https://www.symantec.com/connect/blogs/cloud-storage-apps-malware-delivery-platforms-mdp-dissecting-petya-ransomware-distribution-dro

Upguard, https://www.upguard.com/breaches/cloud-leak-centcom

A. V. Uzunov and E.B.Fernandez, “An Extensible Pattern-based Library and Taxonomy of Security Threats for Distributed Systems”- Special Issue on Security in Information Systems of the Journal of Computer Standards & Interfaces. 2013. http://dx.doi.org/10.1016/j.csi.2013.12.008

A. V. Uzunov, E.B.Fernandez, and K. Falkner, “Engineering Security into Distributed Systems: A Survey of Methodologies”, Journal of Universal Computer Science, Vol. 18, No. 20, 2012, pp. 2920-3006. http://www.jucs.org/jucs_18_20/engineering_security_into_distributed