ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

A Review on Malware Injection in Cloud Computing

Rohit K Rao[1], Vasudha[2], Shraddha Bhat[3]

Under Graduate Student, Dept. of CS&E, St Joseph Engineering College, Vamanjoor, Mangaluru, India[1,2,3]

ABSTRACT: In this modern technological world everything becomes very vast and more reliable .We save data on a remote system and this can be accessed through other systems, but this is possible only with the help of internet. The rapid improvement of the capacity of online connectivity gave birth to cloud computing. Data and processes could be processed online without the need of any local software or client. As long as the user understands the process and has the right security credentials, he could access the system and make the necessary changes, but there are major challenges and security issues in cloud computing that makes accessing difficult. In this paper we are giving an introduction to cloud computing and discussing how to overcome these security issues that are being detected in cloud computing applications.

KEYWORDS: Cloud Computing, Network Security, Malware Injection, Cloud Penetration Testing.

I. INTRODUCTION

In the past decades, the world of computation has experienced some dramatic changes from standalone application to client-server architecture and from distributed to service oriented architecture. All of these transformations aimed to make the software easier to use and improve business process execution efficiency. Cloud computing, an emerging IT delivery model, is the next generation of networking computing which can deliver both software and hardware as on-demand resources and services over the internet with lower IT costs and complexities. Many companies such as Amazon, IBM, Google, Oracle, Microsoft, Salesforce and HP are rushing to provide cloud solutions in various ways. From different perspectives, there are more than a dozen definitions for cloud computing in academia. But the following features of cloud computing defined among them are common:

- Cloud computing is a computing platform to enable resource sharing in terms of scalable infrastructures, middleware and application development platforms and applications.
- New computer technologies, such as service oriented architecture, virtualization, high power enterprise servers and high band width, support to realize cloud computing platforms.

A.Types of cloud computing services

IT people talk about three different kinds of cloud computing, where different services are being provided for you.

1. Infrastructure as a Service (IaaS)

IaaSmeans you're buying access to raw computing hardware over the Net, such as servers or storage. Since you buy what you need and pay-as-you-go, this is often referred to as utility computing. Ordinary web hosting is a simple example of IaaS, you pay a monthly subscription or a per-megabyte/gigabyte fee to have a hosting company serves up files for your website from their servers.

2. Software as a Service (SaaS)

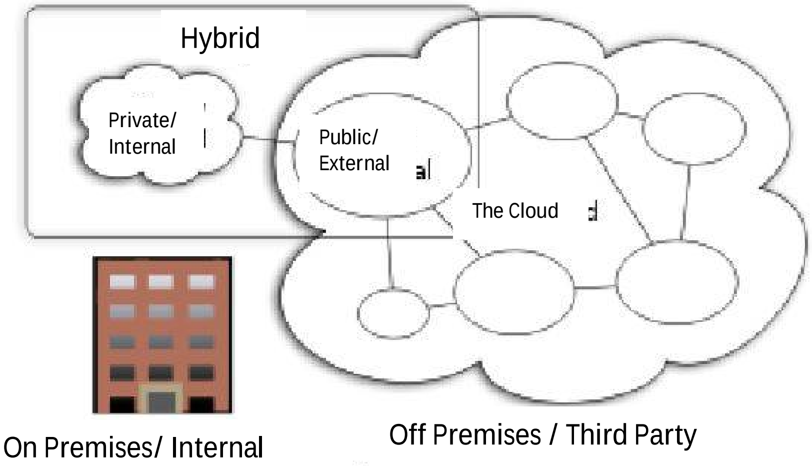
Saas means you use a complete application running on someone else's system. Web-based email and Google Documents are perhaps the best-known examples. Zoho is another well-known SaaS provider offering a variety of office applications online.

3. Platform as a Service (PaaS)
PaaSmeans you develop applications using Web-based tools so they run on systems software and hardware provided by another company. Force.com (from salesforce.com) and the Google App Engine are examples of PaaS.

B. Type of clouds:
1. Public cloud:
A public cloud is one based on the standard cloud computing model, in which a service provider makes resources, like applications and storage, available to the general public over the Internet. Cloud services may be free or offered on a pay-per-usage model. The main benefits of using a public cloud service are:
1. Easy and inexpensive set-up because hardware, application and bandwidth costs are covered by the provider.
2. Scalability to meet needs.
3. No wasted resources

2. Private cloud:
Private cloud (also called internal cloud or corporate cloud) is a marketing term for a proprietary computing architecture that provides hosted services to a limited number of people behind a firewall.

3. Hybrid cloud:
A hybrid cloud environment consisting of multiple internal and/or external providers will be typical for most enterprises. It is a cloud computing environment in which an organization provides and manages some resources in-house and has others provided externally. For example, an organization might use a public cloud service, such as Amazon's Elastic Compute Cloud (EC2) for general computing but store customer data within its own data center.



Cloud Computing Types

II. LITERATURE REVIEW

The migration of networks and servers to cloud also migrates the security threats. This section gives the various injection attacks that are possible on plain network which are also applicable to the cloud environment. Cross Site Scripting is one of the attacks where malicious scripts are injected into a trusted web application through a client site web browser. Reference [18] describes the browser features that help in inducing scriptless attacks and how it helps to access information and establish side channels. It also describes the scriptless attacks against web applications and retrieves sensitive information by abusing legitimate browser concepts. The next type of attack is the command injection attack wherein unwanted system commands are injected and executed through vulnerable applications. Reference [19] describes the occurrence of command injection works, different platforms where the command injection attacks work, their consequences and the mitigation technique. The last type of attack is Sqlia,they are introduced into web browsers through input field, cookies or servers side parameters. The aim of this attack is to retrieve or manipulate sensitive information. Reference [20] describes different Sqlia attacks and methods to protect thee system against the Sqlia attack. Reference [21] describes fully automated technique for detecting, preventing, and reporting SQL Injection Attacks. Reference [11] describes all three types of malware injection attack, surfaces where these attacks can be injected and steps for generation of these attacks to the system.

III. NETWORK SECURITY

In a SaaS deployment model, sensitive data is obtained from the enterprises, processed by the SaaS application and stored at the SaaS vendor end. All dataflow over the network needs to be secured in order to prevent leakage of sensitive information. This involves the use of strong network traffic encryption technique such as Secured Socket Layer (SSL), and Transport Layer Security (TSL) for security.

A. Different network security issues:

1. Data Breaches:

A data breach is a confirmed incident in which sensitive, confidential or otherwise protected data has been accessed and/or disclosed in an unauthorized fashion. Data breaches may involve personal health information (PHI), personally identifiable information (PII), trade secrets or intellectual property.

2. Hijack of accounts:

Cloud account hijacking is a process in which an individual or organization’s cloud account is stolen or hijacked by an attacker. Cloud account hijacking is a common tactic in identity theft schemes in which the attacker uses the stolen account information to conduct malicious or unauthorized activity. When cloud account hijacking occurs, an attacker typically uses a compromised email account or other credentials to impersonate the account owner.

3. Insider threat:

Regardless of the technical and operational countermeasures deployed in an infrastructure, defending against accidental or malicious human actions is hard to do. The insider threat affects virtually every infrastructure. Given the functional context of cloud computing, a malicious insider with access to cloud resources can cause significantly more damage to the organization.

4. Malware Injection:

In Cloud Malware Injection Attack an attacker tries to inject malicious service or virtual machine into the cloud. In this type of attack attacker creates its own malicious service implementation module (SaaS or PaaS) or virtual machine instance (IaaS), and try to add it to the Cloud system. Then, the attacker has to behave so as to make it a valid service to the Cloud system that it is some new service implementation instance among the valid instances. If the attacker succeeds in this, the Cloud automatically redirects the requests of valid user to the malicious service implementation, and the attacker code starts to execute. The main scenario behind the Cloud Malware Injection attack is that an attacker transfers a malicious service instance into cloud so that it can achieve access to the service requests of the victim’s service.

5. Abuse of cloud services:

The expansion of cloud-based services has made it possible for both small and enterprise-level organizations to host vast amounts of data easily. However, the cloud’s unprecedented storage capacity has also allowed both hackers and authorized users to easily host and spread malware, illegal software, and other digital properties. In some cases this practice affects both the cloud service provider and its client. For example, privileged users can directly or indirectly increase the security risks and as a result infringe upon the terms of use provided by the service provider.

6. Insecure APIs:
Application Programming Interfaces (API) gives users the opportunity to customize their cloud experience. However, APIs can be a threat to cloud security because of their very nature. Not only do they give companies the ability to customize features of their cloud services to fit business needs, but they also authenticate, provide access, and effect encryption. As the infrastructure of APIs grows to provide better service, the security risks increases along with it. APIs give programmers the tools to build their programs to integrate their applications with other job-critical software. A popular and simple example of an API is YouTube, where developers have the ability to integrate YouTube videos into their sites or applications. The vulnerability of an API lies in the communication that takes place between applications. While this can help programmers and businesses, they also leave exploitable security risks.

7. Denial of service attacks:
Unlike other kind of cyberattacks, which are typically launched to establish a long-term foothold and hijack sensitive information, denial of service assaults do not attempt to breach your security perimeter. Rather, they attempt to make your website and servers unavailable to legitimate users. In some cases, however, DoS is also used as a smokescreen for other malicious activities, and to take down security appliances such as web application firewalls.

8. Insufficient Due Diligence:
Most of the issues we've looked at here are technical in nature, however this particular security gap occurs when an organization does not have a clear plan for its goals, resources, and policies for the cloud. In other words, it's the people factor.Additionally, insufficient due diligence can pose a security risk when an organization migrates to the cloud quickly without properly anticipating that the services will not match customer's expectation.This is especially important to companies whose data falls under regulatory laws like PII, PCI, PHI, and FERPA or those that handle financial data for customers.

9. Shared Vulnerabilities:
Cloud security is a shared responsibility between the provider and the client.This partnership between client and provider requires the client to take preventative actions to protect their data. While major providers like Box, Dropbox, Microsoft, and Google do have standardized procedures to secure their side, fine grain control is up to you, the client.As Skyfence points out in its article "Office 365 Security & Share Responsibility," this leaves key security protocols – such as the protection of user passwords, access restrictions to both files and devices, and multi-factor authentication – firmly in your hands.The bottom line is that clients and providers have shared responsibilities, and omitting yours can result in your data being compromised.

10. Data Loss:
Data loss is any process or event that results in data being corrupted, deleted and/or made unreadable by a user and/or software or application. It occurs when one or more data elements can no longer be utilized by the data owner or requesting application. Data loss is also known as data leakage.

IV. MALWARE INJECTION

Malware injection is anattack that aims at injecting a malicious service implementation or virtual machine into the Cloud system. Such kind of Cloud malware could serve any particular purpose the adversary is interested in, ranging from eavesdropping via subtle data modifications to full functionality changes or blockings. This attack requires the adversary to create its own malicious service implementation module (SaaS or PaaS) or virtual machine instance (IaaS), and add it to the Cloud system. Then, the adversary has to trick the Cloud system so that it treats the new service implementation instance as one of the valid instances for the particular service attacked by the adversary. If this succeeds, the Cloud system automatically redirects valid user requests to the malicious service implementation, and the adversary's code is executed. A promising countermeasure approach to this threat consists in the Cloud system performing a service instance integrity check prior to using a service instance for incoming requests. This can e.g. be done by storing a hash value on the original service instance's image file and comparing this value with the hash values

DOI: 10.15680/IJIRCCE.2018.0604069

of all new service instance images. Thus, an attacker would be required to trick that hash value comparison in order to inject his malicious instances into the Cloud system. The main idea of the Cloud Malware Injection attack is that an attacker uploads a manipulated copy of a victim's service instance so that some service requests to the victim service are processed within that malicious instance. In order to achieve this, the attacker has to gain control over the victim's data in the cloud system (e.g. using one of the attacks described above). In terms of classification, this attack is the major representative of exploiting the service-to-cloud attack surface. The attacker controlling the cloud—exploits its privileged access capabilities to the service instances in order to attack that service instance's security domains. Different types of Malware Injection are

1. SQL Injection Attack:
Sqlia targets the database underlying an application through a user input field. A destructive SQL command is given as a part of the input field which when substituted into the SQL query makes it a valid one but performs a unexpected harmful action.

2. Cross site scripting attack (XSS):
XSS deals with injecting code into data context of HTML based documents at client and gaining access to sensitive information from server. It allows an attacker to execute scripts in victims' web browser. OWASP classifies XSS attacks as stored and reflected. According to the WHID (2011), 12.58% of the overall attacks on the web are associated with XSS. The variety of attacks based on XSS is almost limitless.

3. Command Injection attack:
Command injection is a type of code injection where the commands are injected in identified vulnerable applications. It allows such inputs to get executed on shell or in the respective runtime environment. The injected commands like ls, ps, cat etc. get executed in the runtime environment with the same privileges that a targeted application possess. One of the major consequences of the above attack is increased waiting time for the other users who makes use of applications running on the same VM in which vulnerable application runs.

IV. TEST FOR MALWARE INJECTION

Cloud Computing Penetration Testing is a method of actively checking and examining the Cloud system by simulating the attack from the malicious code.Cloud computing is the shared responsibility of Cloud provider and client who earn the service from the provider.
Due to impact of the infrastructure, Penetration testing not allowed in SaaSEnvironment.Cloud Penetration Testing allowed in PaaS, IaaS with some Required coordination.Regular Security monitoring should be implemented to monitoring the presence of threats, Risks, and Vulnerabilities.SLA contract will decide what kind oftesting should be allowed and how often it can be done.

A.Important Recommendation for Cloud Penetration Testing:

1. Authenticate users with Username and Password.
2. Secure the coding policy by giving attention Towards Services Providers Policy
3. Strong Password Policy must be advised.
4. Change Regularly by Organization such as user account name, a password assigned by the cloud Providers.
5. Protect information which is uncovered during the Penetration Testing.
6. Password Encryption Advisable.
7. Use centralized Authentication or single sign-on for SaaS Applications.
8. Ensure the Security Protocols are up to date and Flexible.

DOI: 10.15680/IJIRCCE.2018.0604069

B. Important Tools for Cloud Penetration Testing:

1. SOASTA CloudTest:

This suite can enable four types of testing on a single web platform: mobile functional and performance testing and web-based functional and performance testing.

2. LoadStorm:

LoadStorm is a load-testing tool for web and mobile applications and is easy to use and cost-effective.

3. BlazeMeter:

BlazeMeter is used for end-to-end performance and load testing of mobile apps, websites, and APIs.

4. Nexpose:

Nexpose is a widely used vulnerability scanner that can detect vulnerabilities, misconfiguration, and missing patches in a range of devices, firewalls, virtualized systems, cloud infrastructure.

5. AppThwack:

AppThwack is a cloud-based simulator for testing Android, iOS, and web apps on actual devices. It is compatible with popular automation platforms like Robotium, Calabash, UI Automation, and several others.

V. MALWARE-INJECTION ATTACK SOLUTION

Usually when a customer opens an account in the cloud, the provider creates an image of the customer's VM in the image repository system of the cloud. The applications that the customer will run are considered with high efficiency and integrity. Developers propose to consider the integrity in the hardware level, because it is very difficult for an attacker to intrude in the IaaS level. Developers utilize the File Allocation Table (FAT) system architecture, since its straightforward technique is supported by virtually all existing operating systems. From the FAT table developers can know about the code or application that a customer is going to run. Developers can check with the previous instances that had been already executed from the customer's machine to determine the validity and integrity of the new instance. For this purpose, developers need to deploy a Hypervisor in the provider's end. This Hypervisor will be considered the most secured and sophisticated part of the cloud system whose security cannot be breached by any means. The Hypervisor is responsible for scheduling all the instances, but before scheduling it will check the integrity of the instance from the FAT table of the customer's VM.

Another approach is to store the OS type of the customer in the first phase when a customer opens an account. As the cloud is totally OS platform independent, before launching an instance in the cloud, cross checking can be done with the OS type from which the instance was requested from with the account holder's OS type.

A. Solution against Sql Injection attack

1. Employ comprehensive data sanitization. Websites must filter all user input. Ideally, user data should be filtered for context. For example, email addresses should be filtered to allow only the characters allowed in an e-mail address, phone numbers should be filtered to allow only the characters allowed in a phone number, and so on.

2. Use a web application firewall. A popular example is the free, open source module ModSecurity which is available for Apache, Microsoft IIS, and nginx web servers. ModSecurity provides a sophisticated and ever-evolving set of rules to filter potentially dangerous web requests. Its SQL injection defenses can catch most attempts to sneak SQL through web channels.

3. Limit database privileges by context. Create multiple database user accounts with the minimum levels of privilege for their usage environment. For example, the code behind a login page should query the database using an account limited only to the relevent credentials table. This way, a breach through this channel cannot be leveraged to compromise the entire database.

4. Avoid constructing SQL queries with user input. Even data sanitization routines can be flawed. Ideally, using SQL variable binding with prepared statements or stored procedures is much safer than constructing full queries.

5. Eliminate unnecessary database capabilities, especially those that escalate database privileges and those that spawn command shells.

6. Regularly apply software patches. Because SQL injection vulnerabilities are regularly identified in commercial software, it is important to stay up to date on patching.

7. Suppress error messages. These messages are an important reconnaissance tool for attackers, so keep them local if possible. If external messages are necessary, keep them generic.
8. Continuously monitor SQL statements from database-connected applications. This will help identify rogue SQL statements and vulnerabilities. Monitoring tools that utilize machine learning and/or behavioral analysis can be especially useful.

B.Prevention against Cross Site Scripting attack

1. PHP AntiXSS
This is a nice PHP library that can help developers add an extra layer of protection from cross-site scripting vulnerabilities. It automatically detects the encoding of the data that must be filtered. Using of the library is easy.

2. xss_clean.php filter
This is a strong XSS filter that cleans various URF encodings and nested exploits. The developer built the function after analyzing the various sources.

3. HTML Purifier
This is a standard HTML filtering library written in PHP. It removes all malicious code from the input and protects the website from XSS attack. It is also available as a plug-in for most PHP frameworks.
4. xssprotect
xssprotect is another nice library that gives developers a way to clean XSS attack vectors. This Library works by creating the HTML tag tree of the webpage. Then it parses the page and matches all tags. After that, it calls the filter interface to filter improper HTML attributes and XSS attacks. This library is written in Java.

5. XSS HTML Filter
This is another XSS filter for Java. It is a simple single-class utility that can be used to properly sanitize user input against cross-site scripting and malicious HTML code injection.

C. Prevention for command injection attack:
If your application calls out to the operating system, you need to be sure command strings are securely constructed, or else you risk having malicious instructions injected by an attacker. This section outlines a few approaches to protecting yourself.
1. Try to Avoid Command Line Calls Altogether
Modern programming languages have interfaces that permit you to read files, send emails, and perform other operation system functions. Use APIs wherever possible –only use shell commands where absolutely necessary. This will reduce the number of attack vectors in your application, and will also simplify your codebase.
2. Escape Inputs Correctly
Injection vulnerabilities occur when untrusted input is not sanitized correctly. If you use shell commands, be sure to scrub input values for potentially malicious characters:

“;”, “&”, “|”, “`”

Even better, restrict input by testing it against a regular expression of known safe characters. (For example, alphanumeric characters.)
3. Restrict the Permitted Commands
Try to construct all or most of your shell commands using string literals, rather than user input. Where user input is required, try to whitelist permitted values, or enumerate them in a conditional statement.
4. Perform Thorough Code Reviews
Check system calls for vulnerabilities as a part of your code review process. Vulnerabilities often creep in over time – make sure your team knows what to look for.

5. Run with Restricted Permissions

It is a good practice to run your server processes with only the permissions that they require to function – the principle of least privilege. This can help limit the impact of command injection vulnerabilities as a second line of defense.

Make sure each web server process can only access the directories that it needs, and narrow down the directories in which they write or execute files. Consider running the process in a chroot jail if you are running on Unix. This will limit the ability of maliciously injected code to “climb out” of a directory.

VI. CONCLUSION AND FUTURE WORK

Cloud computing is revolutionizing how information technology resources and services are used and managed, but the revolution always comes with new problems. We have depicted some crucial and well known security attacks and studied about Malware Injection. We have seen different types of attacks of malware injection and also different solutions for these attacks.

In the future, we will extend our research by providing implementations and producing results to justify our concepts of security against Malware Injection. The concepts we have discussed here will help to build a strong architecture for security with respect to Malware Injection. This kind of structured security will also be able to improve customer satisfaction to a great extent and will attract more investors in this cloud computation concept for industrial as well as future research farms. Lastly, we propose to build strong theoretical concepts for security in order to build a more generalized architecture to prevent different kinds of attacks.

REFERENCES

[1] Network security threats in cloud-https://www.incapsula.com/blog/top-10-cloud-security-concerns.html
[2] Testing for Malware Injection-https://gbhackers.com/cloud-computing-penetration-testing-checklist-important-considerations/
[3] Prevention for XSS-http://resources.infosecinstitute.com/how-to-prevent-cross-site-scripting-attacks/
[4] Prevention for Sqlia-https://www.esecurityplanet.com/hackers/how-to-prevent-sql-injection-attacks.html
[5] Prevention for Command execution-https://www.hacksplaining.com/prevention/command-execution
[6] Description for Data Breach-https://searchsecurity.techtarget.com/definition/data-breach
[7] Description for Account Hijacking-https://digitalguardian.com/blog/what-cloud-account-hijacking
[8] Description for Data loss-https://www.techopedia.com/definition/29863/data-loss
[9]PriyankaChouhan andRajendra Singh “Security Attacks on Cloud Computing With Possible Solution” , International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 1, January 2016.
[10]Hanqian Wu, Yi Ding, Chuck Winer and Li Yao, “Network Security for Virtual Machine in Cloud Computing”, Computer Sciences and Convergence Information Technology (ICCIT), December 2010.
[11]TulasiRam N, Anusha K and Mary SairaBhanu S, “An Analysis of Malware Injection Attacks and Their Impact on Cloud”, Engineering Sciences International Research Journal, 2330 – 4338, Volume 1, Issue 1, February-2013.
[12] S. Subhashini and V. Kavitha, “A survey on security issue in service models of cloud computing”, Journal of Network and Computer Applications, July-2010.
[13]Ajey Singh and Dr. ManeeshShrivastava, “Overview of Attacks on Cloud Computing”, International Journal of Engineering and Innovative Technology (IJEIT) ,Volume 1, Issue 4, April 2012.
[14] Kandias M., Virvilis N., Gritzalis D. (2013) The Insider Threat in Cloud Computing. In: Bologna S., Hämmerli B., Gritzalis D., Wolthusen S. (eds) Critical Information Infrastructure Security. CRITIS 2011. Lecture Notes in Computer Science, vol 6983. Springer, Berlin, Heidelberg
[15]KrešimirPopović andŽeljkoHocenski, “Cloud computing security issues and challenges”, MIPRO, 2010 Proceedings of the 33rd International Convention, June 2010.
[16] KaziZunnurhain and Susan V. Vrbsky, “Security Attacks and Solutions in Clouds”, 2010.
[17] Naveen Sharma, Dimple Malik and Mahesh Kr. Saini, “Overcoming Network Security Issues in Cloud Computing and its Applications”, International Journal of Computer Applications, 2012
[18] Mario Heiderich, Marcus Niemietz, Felix Schuster, Thorsten Holz, JörgSchwenk ,"Scriptless Attacks Stealing the Pie Without Touching the Sill” CCS’12,North Carolina, USA.
[19] CWE: Individual Dictionary Definition. “Improper Neutralization of Special Elements used in a command” https:// cwe.mitre.org/data/definitions/77.html
[20]Oracle Developers,"An Introduction to SQL Injection Attacks for Oracle Developers" March 2007.
[21] William G.J. Halfond and Alessandro Orso," AMNESIA: Analysis and Mo- nitoring for Neutralizing SQL Injecti- on Attacks", ASE’05, USA.ACM.

BIOGRAPHY

Rohit K Rao is currently studying his Bachelor of Engineering in Computer Science and Engineering at St. Joseph Engineering College, Vamanjoor, Mangaluru, India. His field of interest is Cloud Computing and Cryptography.

Vasudha is currently studying her Bachelor of Engineering in Computer Science and Engineering at St. Joseph Engineering College, Vamanjoor, Mangaluru, India. Her field of interest is Cloud Computing and Cyber security.

Shraddha Bhat is currently studying her Bachelor of Engineering in Computer Science and Engineering at St. Joseph Engineering College, Vamanjoor, Mangaluru, India. Her field of interest is Cloud Computing and Internet of things.

DOI: 10.15680/IJIRCCE.2018.0604069