SLIIT

Discover Your Future

Systems and Network Programming - IE2012

YEAR 2 , SEMESTER 1

CVE RESEARCH REPORT 4

IT22315496

Chamod Anuradha

1) Contents

Abstract

In today's networked digital environment, remote code execution (RCE) is a serious security risk. This abstract gives a general overview of RCE, emphasizing its definition, possible hazards, underlying vulnerabilities, and practical mitigation techniques.

RCE is the term used to describe the malevolent capacity to run instructions or arbitrary code on a remote system or application. Data integrity, confidentiality, and system operation are seriously threatened by attackers who take advantage of weaknesses in target systems to get unauthorized access and control. Injection attacks, in which adversaries alter user inputs or data channels to insert malicious code payloads, are often used in RCE.

The significance of proactive security measures is emphasized in this abstract. In order to mitigate RCE attacks, it addresses the vital roles that vulnerability assessment, secure coding techniques, and prompt patch management play. Strict access restrictions, intrusion detection systems, and web application firewalls are emphasized as essential defensive techniques.

Organizations and people alike must be aware of the dangers associated with RCE attacks sincesuccessful assaults may lead to data breaches, system compromises, and legal repercussions forthe attackers. Maintaining vigilance and putting strong security measures in place are critical to preventing Remote Code Execution attacks as technology advances.

Introduction

Cybersecurity is a constantly changing concern in an age characterized by the fast growth of technology and the pervasive integration of digital systems into every part of our lives. The continuous expansion of linked networks and the widespread use of software applications expose people and businesses to a growing variety of risks. It is critical to recognize and comprehend software and hardware system vulnerabilities in order to remedy these issues.

The vital area of Common Vulnerabilities and Exposures (CVE) research is explored in this study. A widely accepted and defined method for classifying and monitoring vulnerabilities in hardware and software components is called CVE. The fundamental building blocks for comprehending, recording, and reducing cybersecurity risks are contained in CVE entries.

Vulnerabilities are found at an alarming pace as the digital world keeps growing. Organizations, governmental bodies, and people are all at significant risk from cyberattacks, data breaches, and their aftermath. The cybersecurity community relies heavily on CVE, a comprehensive catalog of known vulnerabilities. Through the provision of a methodical and widely accepted approach to detecting and characterizing vulnerabilities, CVE enables professionals, researchers, and stakeholders to communicate, work together, and develop efficient mitigation techniques.

CVE Research Topic

- Remote Code Execution

Introduction

An attack known as remote code execution (RCE) happens when an attacker may run arbitrary code or instructions on a target system or application from a distance. assaults of this kind have the potential to be very dangerous since they provide the attacker with the ability to take over asystem, steal confidential information, alter the way the system behaves, and even use it as a springboard for further assaults.

TryHackMe box link - tryhackme.com/jr/cveresearchproject

An overview of the main ideas behind remote code execution is provided

Remote Attack Vector:

- Since remote code execution (RCE) attacks are usually conducted remotely, the attacker does not need physical access to the target machine. Attackers often take advantage of flaws in services or software that are available across a network, including servers, networked devices, and online applications.

Vulnerabilities:

- RCE attacks often rely on the presence of system or software vulnerabilities. RCE is often caused by vulnerabilities like as buffer overflows, input validation mistakes, and improper deserialization.

Injection Attacks:

- Injection attacks, such SQL injection and command injection, are a popular means of achieving RCE. In order to carry out these attacks, malicious code or instructions are injected into user inputs or other data channels that the target system processes.

Exploitation:

- An attacker must locate and take advantage of a vulnerability in the target system in order to launch an RCE attack. This might include transmitting carefully constructed payloads or inputs that cause the vulnerability and let the attacker run their code.

Payloads:

- Payloads, which are bits of code or instructions intended to take advantage of a vulnerability and run arbitrary code on the target system, are often used by attackers in RCE attacks. These payloads have been thoughtfully designed to exploit a particular vulnerability.

Consequences:

- An attacker may be able to carry out any action authorized by the compromised system or application after they have obtained RCE. This covers data theft, file alteration or deletion, malware or backdoor installation, and system takeover.

Mitigation:

- Organizations and developers should adhere to security best practices, such as input validation, code reviews, and the timely patching of known vulnerabilities, in order to avoid RCE attacks. RCE attempts may also be found and stopped with the use of intrusion detection systems and web application firewalls.

Common Targets:

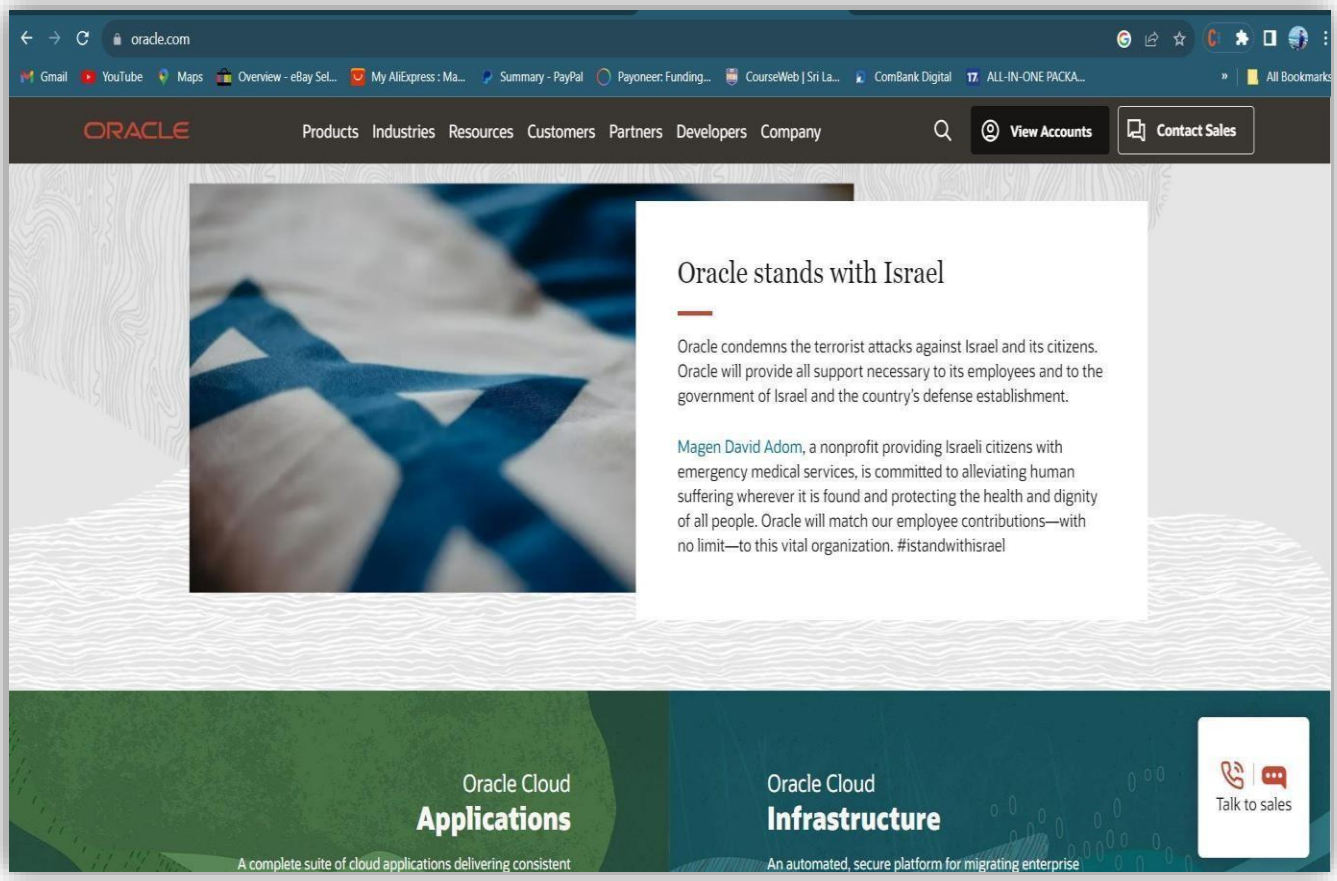
- Common targets of RCE attacks include servers, online apps, Internet of Things devices, and any networked system. Online apps are especially vulnerable because of the complexity of web technology and the variety of user inputs they might process.

Legal Implications:

- RCE assaults are prohibited and may have serious legal repercussions for those who carry them out. Prison time and fines are possible punishments, depending on the jurisdiction and degree of harm.

Research Findings

CVE-2020- 14883



A security vulnerability that was identified in 2020 has been awarded the Common Vulnerabilities and Exposures (CVE) identification CVE-2020-14883. Please be aware, nevertheless, that I do not have access to real-time data, and that my understanding is based on data that was accessible as of September 2021. I am thus able to provide details on this vulnerability as of my most recent update.

Popular Java-based application server Oracle WebLogic Server is linked to CVE-2020-14883. The Console component of the server is vulnerable. This vulnerability specifically relates to a deserialization problem in the Console component, which can provide an attacker remote code execution capabilities without the need for authentication.

Impact of the Vulnerabilities:

CVE-2020-14883 may have a major and perhaps dangerous effect:

1. Remote Code Execution: Via this vulnerability, attackers may remotely run any code on the target server. This implies that they could be able to take over the server, steal confidential information, or carry out other nefarious deeds.

2. Unauthorized Access: The flaw could enable unauthorized users to access the Oracle WebLogic Server Console, which might result in further data leaks or exploitation.

3. System Compromise: This vulnerability has the potential to completely compromise the affected system if it is properly exploited.

Summary of Vulnerabilities:

In conclusion, deserialization problems have resulted in a remote code execution vulnerability (CVE-2020-14883) in the Oracle WebLogic Server Console component. This vulnerability maybe used by attackers to run arbitrary code on the target system without requiring authentication, granting them access without authorization and possibly compromising the system.

Potential mitigations:

In order to reduce the risks related to CVE-2020-14883, take into account the following actions:

1. Patch or Update: To fix this issue, Oracle most likely published a security patch or update. Make that the most recent security fixes are applied to your Oracle WebLogic Server installation.

2. Network Segmentation: To restrict access to the Oracle WebLogic Server Console, use network segmentation. Permit only those who are required and trustworthy to access it.

3. Firewall Rules: Use firewall rules to limit traffic that enters and leaves the server, particularly to ports connected to the Console component.

4. Access restrictions : Set up robust authentication procedures and access restrictions to get access to the Console. Create strong, one-of-a-kind passwords and, if available, take into accounttwo-factor authentication (2FA).

5. Security Best Practices : When setting up and maintaining your Oracle WebLogic Server installation, adhere to security best practices. Oracle offers security recommendations and documentation for its products.

6. Security Monitoring: To identify and stop any unusual activity or attempts to take advantage of this vulnerability, put strong security monitoring and intrusion detection systems in place.

CVE Exploit

Initially, you may get around the console component's authentication by sending a straightforward HTTP GET request to a double-encrypted endpoint, which hosts the Console Portal page and triggers the CVE-2020-14882 vulnerability.

```
1  curl -k -X "GET" https://<HOST>/console/css/%252e%252e%252fconsole.portal
```

This HTTP GET request will validate that your target server is vulnerable by bypassing authentication and rerouting you to the Administrator Console page.

After that, you will be able to setup, track, and manage the apps through your connection to the target domain.

You may submit commands through an MVEL expression under the handle of the com.tangosol.coherence.mvel2.sh.Shell Session class by taking advantage of CVE-2020-14883.

A malevolent, unauthenticated attacker may be able to utilize this to execute arbitrary codeon the server via remote code execution.

Curl -k -X "GET" https://<HOST>/console/css/%252e%252e%252fconsole.portal?_nfpb=true &_pageLabel=& handle=com.tangosol.coherence.mvel2.sh.ShellSession("java.lang.Runtime. getRuntime().ex ec('touch%20/tmp/pentest00ls');")

Conclusion

In summary, the Common Vulnerabilities and Exposures (CVE) system is a critical tool used by the cybersecurity community to manage the constantly evolving world of digital threats. CVE is a fundamental tool that helps academics, businesses, and cybersecurity professionals prioritize vulnerabilities, communicate clearly, and create plans to protect important systems and data. In this age of unrelenting technological development, the importance of CVE research is immeasurable. By keeping up a thorough awareness of vulnerabilities, their effects, and mitigation techniques, we may better guard our globalized society from the always changing threats posed by cybersecurity.


THANK YOU!