International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

SMBLEEDING GHOST VULNERABILITY

Pratik Gupta[1], Tanvi Humane[2], Harshal Ismulwar[3], Krishna Kounder[4] , Asha Durafe[5], Ravindra Gotavade[6]
[1 2 3 4 5]*Department of Cyber Security, Mumbai University, India*
E-mail: Tanvi.humane16554@sakec.ac.in
E-mail: Krishna.kounder16745@sakec.ac.in
E-mail: Pratik.gupta16458@sakec.ac.in
E-mail: Harshal.ismulwar16157@sakec.ac.in
E-mail: Asha.durafe@sakec.ac.in
[6]E-mail: ravindragotavade007@gmail.com

---***---

Abstract - SMB protocol or server message block, one of Windows-based file sharing and network communication's cornerstones, has been subjected to concerns and examination of its security for a long time. However, as another episode in these concerns, a new spectre has arrived – SMBleeding Ghost. This abstract explains the details of the above-stated horror, discussing its starting points, effects, and ways of prevention. SMBleeding Ghost, as the name signifies, uses the SMB protocol stack vulnerabilities. Specifically, it occurs as a result of memory allocation weaknesses and the twisting of SMB packets developed to exploit these weaknesses. The latter is reserved for memory violation and, thus, may be used for remote code execution or trigger additional fancy issues including denial-of-service breaches and data leaks.

***Key Words*: The 5 keywords will be listed below**

1. INTRODUCTION

In the realm of cybersecurity, the landscape is constantly evolving, with new threats emerging and existing vulnerabilities taking on new forms. Among the myriad of potential dangers lurking in network protocols, the Server Message Block (SMB) protocol stands as both a cornerstone of file sharing and a perennial target for malicious actors. Over the years, vulnerabilities in SMB implementations have been exploited to devastating effect, leading to data breaches, system compromises, and widespread disruption. Amidst this backdrop of ongoing scrutiny, a new spectre has emerged - the SMBleeding Ghost vulnerability. Named for its elusive nature and clandestine behaviour, this vulnerability represents a significant threat to the security and integrity of Windows environments worldwide. Exploiting weaknesses in the SMB protocol stack, the SMBleeding Ghost vulnerability leverages intricate manipulation of SMB packets to trigger memory corruption, potentially leading to a range of exploits including remote code execution, denial of service, and information disclosure.

What sets the SMBleeding Ghost vulnerability apart from its predecessors is its ability to operate covertly within legitimate SMB traffic. Unlike overt attacks that may trigger alarms or raise suspicion, the SMBleeding Ghost flaw exploits subtle vulnerabilities in memory allocation mechanisms, evading traditional detection mechanisms and exacerbating the challenge of identifying and mitigating the threat. The consequences of the SMBleeding Ghost vulnerability extend beyond individual systems, posing significant risks to network security, data confidentiality, and organizational resilience. With the potential for widespread exploitation across diverse Windows environments, the urgency to address this vulnerability is paramount. In this context, proactive measures are essential. Timely patching, network segmentation, protocol hardening, and enhanced monitoring are among the strategies recommended to mitigate the risks posed by the SMBleeding Ghost vulnerability. Furthermore, collaboration between security researchers, vendors, and organizations is crucial for sharing insights and developing effective countermeasures against this spectral menace. As organizations navigate the complexities of modern cybersecurity, vigilance against emerging threats such as the SMBleeding Ghost vulnerability is indispensable. By understanding its intricacies and implementing comprehensive security measures, enterprises can fortify their defences and safeguard against the lurking dangers within the SMB protocol stack.

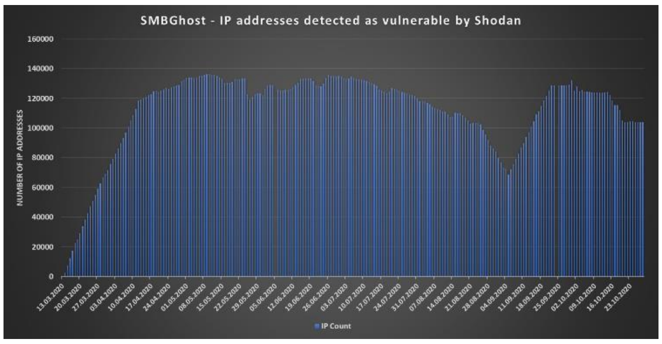


Figure 1Vulnerable IP address detected by Shodan Credits: Jan Kopriva

2. TECHNICAL BACKGROUND

2.1) SMB Protocol

Within the SMBv3 protocol, data compression plays a crucial role in optimizing network performance and reducing bandwidth utilization. This functionality is particularly relevant in scenarios where large amounts of data need to be transferred between client and server over the network. In the context of SMBleeding, understanding the compression mechanism within SMBv3, including the Srv2DecompressData function, is essential for comprehending the vulnerability's exploitation.

The compression mechanism in SMBv3 operates as follows:

1) Negotiation: Before data compression can be utilized, client and server negotiate the use of compression during the SMB session setup phase. This negotiation occurs as part of the SMB dialect negotiation process, where both client and server agree on the capabilities and features to be used during the session.
2) Compression Algorithms: SMBv3 supports various compression algorithms, including LZ77 (Lempel-Ziv 1977), LZNT1 (Lempel-Ziv NT), and others. These algorithms are used to compress data before transmission over the network. The choice of compression algorithm depends on factors such as performance requirements and compatibility between client and server.
3) Data Compression: When a client sends data to the server or vice versa, the data can be compressed using the agreed-upon compression algorithm. The compression process involves analysing the data stream for repetitive patterns and replacing them with shorter representations, thereby reducing the overall size of the data.
4) Srv2DecompressData Function: The Srv2DecompressData function is a critical component within the SMBv3 server-side implementation responsible for decompressing compressed data received from clients. This function is invoked when compressed data is received by the server, and it performs the reverse operation of the compression algorithm to decompress the data before further processing or storage.

Exploiting the SMBleeding vulnerability involves manipulating the compression mechanism within SMBv3 to trigger memory corruption and potentially execute arbitrary code on the server. By sending specially crafted compressed data packets to the server, attackers can exploit vulnerabilities in the Srv2DecompressData function, leading to memory corruption and potentially allowing for unauthorized access or control of the server. Understanding the intricacies of the compression mechanism in SMBv3, including functions like Srv2DecompressData, is crucial for both mitigating the SMBleeding vulnerability and developing effective countermeasures to protect against similar exploitation attempts in the future. It underscores the importance of thorough protocol analysis and robust security measures in networked environments to safeguard against potential threats and vulnerabilities.

2.2) Compression Mechanism

Within the SMBv3 protocol, data compression plays a crucial role in optimizing network performance and reducing bandwidth utilization. This functionality is particularly relevant in scenarios where large amounts of data need to be transferred between client and server over the network. In the context of SMBleeding, understanding the compression mechanism within SMBv3, including the Srv2DecompressData function, is essential for comprehending the vulnerability's exploitation.

The compression mechanism in SMBv3 operates as follows:

1. Negotiation: Before data compression can be utilized, client and server negotiate the use of compression during the SMB session setup phase. This negotiation occurs as part of the SMB dialect negotiation process, where both client and server agree on the capabilities and features to be used during the session.
2. Compression Algorithms: SMBv3 supports various compression algorithms, including LZ77 (Lempel-Ziv 1977), LZNT1 (Lempel-Ziv NT), and others. These algorithms are used to compress data before transmission over the network. The choice of compression algorithm depends on factors such as performance requirements and compatibility between client and server.
3. Data Compression: When a client sends data to the server or vice versa, the data can be compressed using the agreed-upon compression algorithm. The compression process involves analyzing the data stream for repetitive patterns and replacing them with shorter representations, thereby reducing the overall size of the data.
4. Srv2DecompressData Function: The Srv2DecompressData function is a critical component within the SMBv3 server-side implementation responsible for decompressing compressed data received from clients. This function is invoked when compressed data is received by the

server, and it performs the reverse operation of the compression algorithm to decompress the data before further processing or storage.

Exploiting the SMBleeding vulnerability involves manipulating the compression mechanism within SMBv3 to trigger memory corruption and potentially execute arbitrary code on the server. By sending specially crafted compressed data packets to the server, attackers can exploit vulnerabilities in the Srv2DecompressData function, leading to memory corruption and potentially allowing for unauthorized access or control of the server.

Understanding the intricacies of the compression mechanism in SMBv3, including functions like Srv2DecompressData, is crucial for both mitigating the SMBleeding vulnerability and developing effective countermeasures to protect against similar exploitation attempts in the future. It underscores the importance of thorough protocol analysis and robust security measures in networked environments to safeguard against potential threats and vulnerabilities.

3. Unveiling the Vulnerabilities

3.1) SMB Ghost (CVE-2020-0796)

The SMBGhost vulnerability, also known as CVE-2020-0796, represents a critical security flaw found in Microsoft's Server Message Block 3.1.1 (SMBv3) protocol implementation in certain versions of Windows. This vulnerability enables remote attackers to execute arbitrary code on vulnerable systems without requiring any user interaction. At its core, the vulnerability arises from a flaw in the way SMBv3 handles compression during the negotiation process between clients and servers. This negotiation, occurring during the session setup phase, involves determining the use of compression algorithms to optimize data transfer over the network. However, a flaw in this negotiation process allows attackers to send specially crafted packets, exploiting vulnerabilities within SMBv3. Specifically, attackers can send malformed compressed data packets to vulnerable SMBv3 servers, triggering a buffer overflow condition. This occurs when the server attempts to decompress the received data but fails to properly validate the size of the decompressed data, leading to memory corruption. Through this buffer overflow, attackers gain the ability to overwrite critical memory locations with malicious code of their choice. Subsequently, they can execute arbitrary code with elevated privileges on the vulnerable system, potentially leading to complete compromise of the system.

The flaw in data decompression within SMBv3 presents a pathway for local privilege escalation, wherein attackers exploit the vulnerability to execute arbitrary code with elevated privileges. Leveraging this initial foothold, attackers can escalate their privileges to gain further control over the system, enabling them to manipulate system configurations, access sensitive data, install persistent malware, or compromise other resources on the network. Given the severity of the SMBGhost vulnerability, prompt patching and mitigation measures are essential to safeguard vulnerable systems from potential exploitation.

3.2) SMBleed (CVE-2020-1206)

The SMBleed vulnerability, designated as CVE-2020-1206, is a critical security flaw found within Microsoft's implementation of the Server Message Block (SMB) protocol. It particularly affects versions of SMB implementing the SMBv3.1.1 compression feature, which enables data compression during the transfer of SMB packets over networks. During the negotiation phase between clients and servers, where compression settings are agreed upon, this vulnerability arises due to flawed handling of compressed data within SMBv3.1.1. Attackers exploit this flaw by sending specially crafted compressed data packets to vulnerable SMB servers. Exploiting SMBleed allows attackers to compel the server to read uninitialized kernel memory and return portions of this memory within the server's response to client requests. This facilitates remote retrieval of sensitive information residing in the server's memory, potentially including passwords, cryptographic keys, or other confidential data. The exploitation process entails sending malicious requests to the vulnerable SMB server, triggering the reading of uninitialized kernel memory. Attackers can then analyze the retrieved data to extract valuable information, which could aid in further compromising the server or other systems within the network.

The significance of SMBleed lies in its potential to remotely disclose sensitive information, thereby undermining the security integrity of affected systems. To address this vulnerability and mitigate associated risks, timely patching and implementation of mitigation measures are imperative. Additionally, deploying network monitoring and intrusion detection systems can enhance the ability to detect and thwart attempts to exploit SMBleed in real-time, bolstering overall security resilience against this threat.

3.3) Chaining SMBGhost & SMBleed for RCE

Combining the SMBGhost and SMBleed vulnerabilities for remote code execution (RCE) entails a sophisticated process for attackers. Initially, the attacker exploits SMBleed (CVE-2020-1206) by sending crafted packets to a vulnerable Windows system, triggering the flaw and enabling access to uninitialized kernel memory. This allows the extraction of sensitive data, such as memory addresses or pointers to critical kernel functions or structures. Subsequently, the attacker analyzes this leaked information to identify valuable targets for exploitation within the system's memory space. Armed with this data, the attacker crafts a malicious payload tailored to exploit the SMBGhost vulnerability (CVE-2020-0796). By leveraging the SMBGhost vulnerability, which involves a buffer overflow in the SMBv3 protocol implementation, the attacker can execute arbitrary code on the target system with kernel-level privileges. This achieves remote code execution, granting the attacker control over the compromised system. With RCE achieved, the attacker can execute various malicious activities, including deploying additional payloads, establishing persistence, exfiltrating sensitive data, or carrying out other objectives aligned with their malicious intent. Thus, the seamless integration of these vulnerabilities underscores the critical importance of promptly patching and securing systems to mitigate the risk of such exploits.

4. Working

The SMBleeding Ghost implementation is characterized by its technical simplicity. One can readily access the payload via GitHub, attributed to the username chompie1337. The testing phase of this project is exclusively carried out within a controlled environment. The hardware and software configurations utilized during the implementation phase are detailed below.

1) Hardware.

Xiaomi notebook Ultra.
Processor- 11th gen Intel (R) Core^{TM} i7-11370H @ 3.30GHz
Ram- 16.0 GB DDR4
GPU- Intel(R) Iris(R) Xe Graphics 8 Gb

2) Software.
   a) Base OS: Windows 11 Home 23H2
   b) Oracle VM VirtualBox
   c) Attacking OS – Parrot OS Sec. edition
   d) Target OS- Windows 10 professional 19H1
3) Tools used
   a) Nmap
   b) Metasploit
4) Private Internet Connection

Router: Nokia 2.4 hz band

Additionally, network configuration played a pivotal role in the execution. The bridged network adapter treated each of the three operating systems as distinct entities, reinforcing support for the SMB exploitation thesis. The operational mechanics of the SMBleeding Ghost vulnerability are delineated as follows:

Step 1- Configuring Virtual machine with Widows 10 & Parrot OS

Oracle's VM VirtualBox is a preferred third-party solution due to its open-source nature and the provision of comprehensive manual customization options. Initially, we installed Parrot OS. It's imperative to ensure that the network adapter is configured for bridged connection, as this assigns distinct IP addresses to each operating system, facilitating their independent operation.

Subsequently, we virtualized Windows 10 version 19H3. Similar steps were taken to configure the bridged network adapter for this virtual machine, ensuring seamless connectivity and individualized functionality.

Step 2- Configuration of SMBleeding ghost vulnerability

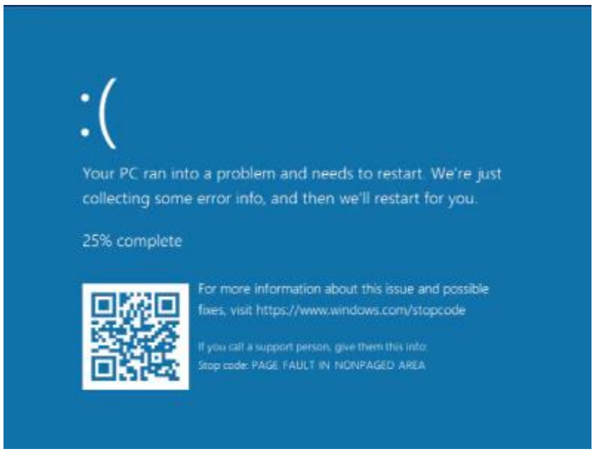
After successfully configuring the operating systems, the target machine, namely Windows 10 19H1, undergoes SMB vulnerability scanning. Upon obtaining positive results from the scan, an offset script is executed to acquire the machine's offset, which is then incorporated into the payload file.

Step 3- Establishing Reverse TCP connection

A reverse TCP connection is established between the target machine and the attacker machine utilizing the Nmap tool. This connection serves to maintain access even after the target machine has been exploited.

Step 4- Exploiting Machine & maintaining Shell Access

The payload persistently attempts to acquire shell access by targeting the Server Message Block (SMB) protocol of the operating system. Nmap maintains reverse shell access until the connection is terminated. However, upon termination of the connection, it results in the crashing of the Windows 10 operating system.



:(

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

25% complete

For more information about this issue and possible fixes, visit https://www.windows.com/stopcode

If you call a support person, give them this info:
Stop code: PAGE FAULT IN NONPAGED AREA

Figure 2Blue screen of Death in Windows 10

5. Impact & Mitigation

5.1) Potential Consequences of SMBleeding Ghost Exploits

A successful exploitation of the SMBleeding vulnerability, which combines the weaknesses of SMBGhost (CVE-2020-0796) and SMBleed (CVE-2020-1206), can unleash a cascade of devastating consequences, posing grave risks to both individual users and organizations alike. Foremost among these is the looming threat of data breaches, where attackers can illicitly access sensitive information stored on compromised systems. This could encompass a wide array of data ranging from personally identifiable information (PII) to financial records and proprietary intellectual property, potentially leading to identity theft, fraud, or other malicious activities. Furthermore, the exploit's ability to achieve remote code execution with kernel-level privileges enables attackers to wrest complete control over the compromised system. This facilitates a range of nefarious actions including the execution of arbitrary commands, installation of malware, creation of backdoors, and alteration of system configurations. Consequently, such a system takeover could result in severe disruptions to services, compromise data integrity, and undermine critical infrastructure components. Additionally, the exploit's utility as an entry point for ransomware attacks exacerbates the risk landscape, with potential consequences including substantial financial losses, operational downtime, and reputational harm for affected organizations. Furthermore, the vulnerability's potential to propagate worms across interconnected networks amplifies the scope and impact of the attack, while regulatory compliance violations further compound the repercussions for non-compliant organizations. To mitigate these multifaceted risks, proactive measures such as promptly applying security patches, implementing robust cybersecurity protocols, and establishing proactive threat detection and response strategies are imperative.

5.2) Patching and Security Measures

Applying security patches to address the SMBleeding vulnerability is essential for mitigating the risks associated with this exploit. These patches, issued by software vendors, close identified security gaps and vulnerabilities, including those exploited by SMBleeding. Failing to apply these patches promptly leaves systems vulnerable to exploitation, increasing the likelihood of successful attacks such as data breaches and system compromises. In addition to patching, implementing supplementary security measures can further bolster defenses against SMBleeding:

Firstly, network segmentation involves dividing a network into smaller, isolated segments to limit the impact of security breaches and unauthorized access. By segregating sensitive systems and data, organizations can contain the spread of attacks and minimize potential damage in case of compromise. Additionally, configuring firewalls to restrict access to vulnerable services like SMB can prevent unauthorized exploitation. Application-aware firewalls can inspect SMB traffic for signs of malicious activity, blocking harmful connections before they reach vulnerable systems.

Moreover, deploying Intrusion Detection and Prevention Systems (IDPS) enables real-time monitoring of network traffic for suspicious activity, such as attempts to exploit SMBleeding. By detecting and blocking malicious traffic, IDPS solutions provide organizations with enhanced visibility and protection against cyber threats. Furthermore, educating users about security best practices empowers them to recognize and report potential threats, thereby strengthening the organization's overall security posture.

6. Conclusion

The emergence of the SMBleeding vulnerability, a convergence of SMBGhost (CVE-2020-0796) and SMBleed (CVE-2020-1206), poses a grave threat to Windows systems worldwide. This exploit chain begins innocuously enough with SMBleed, a vulnerability allowing attackers to glean sensitive kernel memory information through ingeniously crafted SMB packets. Exploiting SMBleed effectively opens a backdoor for threat actors to clandestinely access crucial system data, leveraging this initial breach to orchestrate more devastating attacks. Once armed with the leaked kernel memory information, attackers pivot towards exploiting the SMBGhost vulnerability. SMBGhost, notorious for its buffer overflow flaw within the SMBv3 protocol implementation, presents a golden opportunity for attackers to escalate their intrusion. By exploiting SMBGhost, attackers can execute malicious code remotely with elevated kernel-level privileges, effectively seizing control of the vulnerable system. The potential consequences of such a compromise are dire, ranging from unauthorized access to sensitive information, data exfiltration, and even complete system compromise.

To underscore the gravity of the situation, the ramifications of a successful SMBleeding attack extend far beyond immediate data breaches. Organizations and users must recognize the pivotal importance of promptly applying security patches provided by vendors. Timely patching not only mitigates the specific vulnerabilities associated with SMBleeding but also serves as a crucial line of defence against a myriad of other potential threats lurking in the digital ecosystem. Furthermore, maintaining robust security practices is paramount in the ongoing battle against cyber threats. Implementing rigorous network segmentation, enforcing the principle of least privilege, and conducting regular security audits are indispensable measures for fortifying defences. By fostering a culture of proactive security awareness and diligence, organizations can significantly bolster their resilience against the ever-evolving threat landscape. In a world where cyber adversaries constantly seek to exploit vulnerabilities, staying updated with security patches and adhering to best security practices are indispensable pillars of cyber defence.

ACKNOWLEDGEMENT

We have great pleasure in presenting this paper on **"SMBleeding Ghost Vulnerability"**. We take this opportunity to express our sincere thanks to our Guide, **Dr. Asha Durafe** & **Mr. Ravindra Gotavade**, the faculty in Department of Cyber Security in Shah & Anchor Kutchhi Engineering College for guiding us and suggesting regarding the link of work. We would like to express our gratitude towards their constant encouragement, support and guidance throughout the progress.

Also, we would thank our Principal **Dr. Bhavesh Patel** and **Dr. Nilakshi Jain**, Head of Cyber Security Department, for their help & guidance of this project. We are also thankful to all faculty member of our department for their help and guidance during completion of our paper

REFERENCES

[1] National Institute of Standards and Technology (NIST). "NISTSpecial Publication 800-115: Technical Guide to InformationSecurity Testing and Assessment." NIST, 2008. (Offers guidelinesand best practices for information security testing and assessment,including vulnerability scanning.)
[2] Microsoft. "Microsoft Security VulnerabilityResearchDefense."[Website].Available:https://msrc-blog.microsoft.com/. (Provides insights into Microsoft's approach tovulnerability research and defense, including security advisoriesand best practices.)
[3] Kalyan Kumar Dasari & Dr, K.Venkatesh Sharma, ‚Mobile AgentApplications in Intrusion Detection System (IDS)'-JASC, Volume4, Issue 5, October/2017, ISSN NO:1076-5131, Pages: 97-103.
[4] Kalyan Kumar Dasari& Dr, K.Venkatesh Sharma, ‚Analyzing theRole of Mobile Agent in Intrusion Detection System'-JASRAE,Vol. XV, Issue No. 1, April-2018, ISSN 2230-7540, Pages: 566-573.
[5] Kalyan Kumar Dasari& Dr, K.Venkatesh Sharma, ‚A Study onNetwork Security through a Mobile Agent Based IntrusionDetection Framework'-JASRAE, Vol. XI, Issue No. 22, July-2016,ISSN 2230-7540, Pages: 209-214
[6] K. K. Kumar, S. G. B. Kumar, S. G. R. Rao and S. S. J. Sydulu, "Safeand high secured ranked keyword searchover an outsourcedcloud data," 2017 International Conference on InventiveComputing and Informatics (ICICI),

Coimbatore, India, 2017, pp.20-25, doi: 10.1109/ICICI.2017.8365348.

[7] K. K. .Kommineni and A. . Prasad, ‚A Review on Privacy andSecurity Improvement Mechanisms in MANETs‛, Int JIntellSystApplEng, vol. 12, no. 2, pp. 90–99, Dec. 2023.

[8] Kalyan Kumar Dasari&amp; M.Prabhakar ‚ProfessionallyResolve the Password Security knowledge in the Contexts ofTechnology‛-IJCCIT, Vol. 3, Issue. 1, April' 2015;ISSN: 2345 – 9808(2015).

[9] V.Mounika D. Kalyan Kumar ‚Background Subtraction by UsingDE Color Algorithm‛ -IJATCSE, ISSN 2278-3091 Vol: 3, No: 1,Pages: 273-277(2014).

[10] Vellela, S.S., Balamanigandan, R. Optimized clustering routingframework to maintain the optimal energy status in the wsnmobile cloud environment. Multimed Tools Appl (2023).https://doi.org/10.1007/s11042-023- 15926-5

[11] Vellela, S. S., Reddy, B. V., Chaitanya, K. K., &Rao, M. V. (2023,January). An Integrated Approach to Improve E-HealthcareSystem using Dynamic Cloud Computing Platform. In 2023 5thInternational Conference on Smart Systems and InventiveTechnology (ICSSIT) (pp. 776-782). IEEE.

[12] K. N. Rao, B. R. Gandhi, M. V. Rao, S. Javvadi, S. S. Vellela and S.KhaderBasha, "Prediction and Classification of Alzheimer'sDisease using Machine Learning Techniques in 3D MR Images,"2023 International Conference on Sustainable Computing andSmart Systems (ICSCSS), Coimbatore, India, 2023, pp. 85-90, doi:10.1109/ICSCSS57650.2023.10169550.

[13] VenkateswaraRao, M., Vellela, S., Reddy, V., Vullam, N., Sk, K. B.,&Roja, D. (2023, March). Credit Investigation and ComprehensiveRisk Management System based Big Data Analytics inCommercial Banking. In 2023 9th International Conference onAdvanced Computing and Communication Systems (ICACCS)(Vol. 1, pp. 2387-2391). IEEE

[14] S Phani Praveen, RajeswariNakka, AnuradhaChokka,VenkataNagarajuThatha, SaiSrinivasVellela and UddagiriSirisha,‚A Novel Classification Approach for Grape Leaf DiseaseDetection Based on Different Attention Deep LearningTechniques‛ International Journal of Advanced ComputerScience and Applications(IJACSA), 14(6), 2023.http://dx.doi.org/10.14569/IJACSA.2023.01406128

[15] Vellela, S. S., &Balamanigandan, R. (2022, December). Design ofHybrid Authentication Protocol for High Secure Applications inCloud Environments. In 2022 International Conference onAutomation, Computing and Renewable Systems (ICACRS) (pp.408-414). IEEE

[16] Wikipedia : https://en.wikipedia.org/wiki/SMBGhost

[17] AWAKE the NDR security devision of ARISTA: https://www.arista.com/en/solutions/security

[18] Macafee.com: https://www.mcafee.com/blogs/other-blogs/mcafeelabs/smbghost-analysis-of-cve-2020-0796/

[19] Aleksa Tamburkovski (YouTube channel): https://www.youtube.com/watch?v=Ltt2Cz_Dy9c&t=694