Team 7.3

Main Website: vtop.vit.ac.in

1)Vulnerability Name: **SL Medium Strength Cipher Suites Supported (SWEET32)**

CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

OWASP Category: Broken access control

Description: The remote host supports the use of SSL ciphers that offer medium strength encryption. Nessus regards medium strength as any encryption that uses key lengths at least 64 bits and less than 112 bits, or else that uses the 3DES encryption suite.

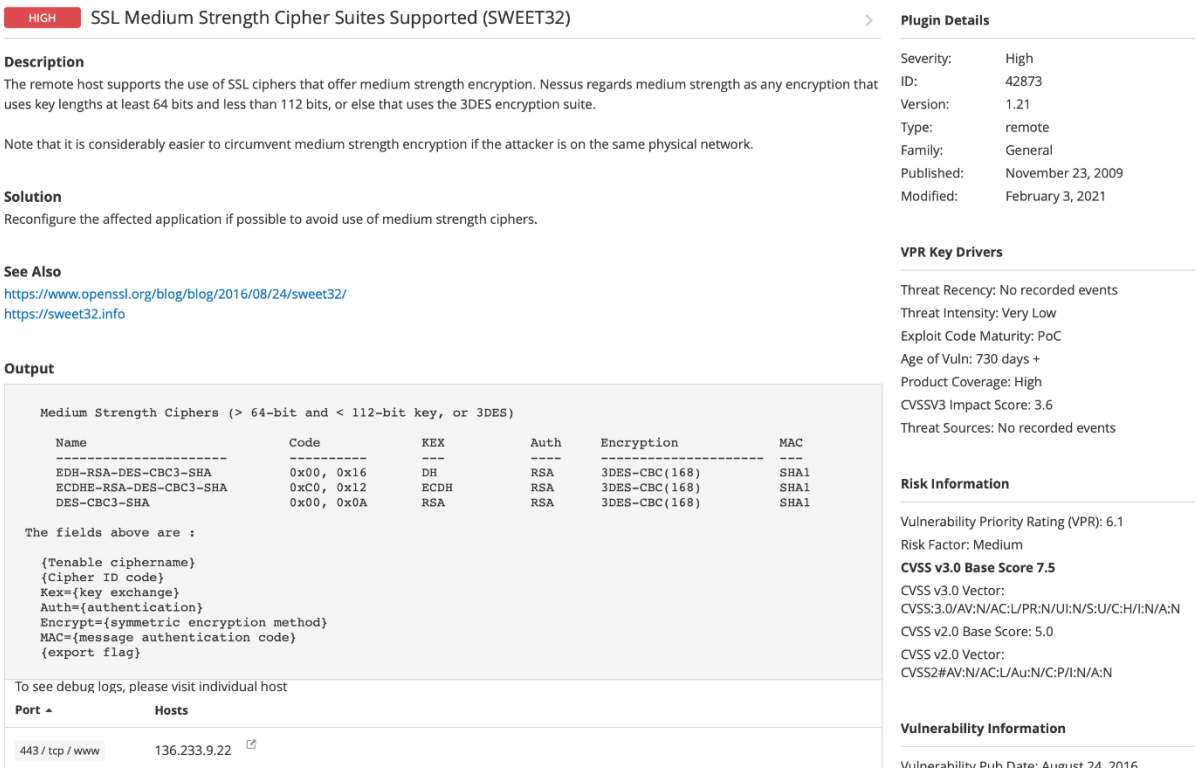
Note that it is considerably easier to circumvent medium strength encryption if the attacker is on the same physical network.

Business Impact: The vulnerability known as SWEET32 (Short for Sweet 32 Birthday Attack) relates to the continued use of legacy cryptographic cipher suites with 64-bit block sizes in TLS and SSL protocols. These suites are susceptible to birthday attacks, which exploit the probability of two blocks colliding after a certain number of transactions. If an organization's systems are vulnerable to SWEET32, it means they're potentially at risk of having encrypted communications intercepted or manipulated. This could lead to unauthorized access, data theft, or even the injection of malicious code. As a result, businesses face a significant risk to the confidentiality and integrity of their sensitive information. Remediation involves updating systems to support more secure cipher suites with larger block sizes, which would effectively mitigate this vulnerability and enhance the overall security posture. Failure to address SWEET32 could leave an organization exposed to increasingly sophisticated attacks and legal consequences due to potential data breaches.

Vulnerability Path : The Sweet32 vulnerability when detected with a vulnerability scanner will report it as a CVSS 7.5.

Vulnerability Parameter:

Steps to Reproduce:



HIGH SSL Medium Strength Cipher Suites Supported (SWEET32)

Description

The remote host supports the use of SSL ciphers that offer medium strength encryption. Nessus regards medium strength as any encryption that uses key lengths at least 64 bits and less than 112 bits, or else that uses the 3DES encryption suite.

Note that it is considerably easier to circumvent medium strength encryption if the attacker is on the same physical network.

Solution

Reconfigure the affected application if possible to avoid use of medium strength ciphers.

See Also

https://www.openssl.org/blog/blog/2016/08/24/sweet32/
https://sweet32.info

Output

```
Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

  Name                    Code          KEX       Auth    Encryption          MAC
  ----------------------  ----------    ---       ----    ------------------  ---
  EDH-RSA-DES-CBC3-SHA    0x00, 0x16    DH        RSA     3DES-CBC(168)       SHA1
  ECDHE-RSA-DES-CBC3-SHA  0xC0, 0x12    ECDH      RSA     3DES-CBC(168)       SHA1
  DES-CBC3-SHA            0x00, 0x0A    RSA       RSA     3DES-CBC(168)       SHA1

The fields above are :

  {Tenable ciphername}
  {Cipher ID code}
  Kex={key exchange}
  Auth={authentication}
  Encrypt={symmetric encryption method}
  MAC={message authentication code}
  {export flag}
```

To see debug logs, please visit individual host

Port	Hosts
443 / tcp / www	136.233.9.22

Plugin Details

Severity:	High
ID:	42873
Version:	1.21
Type:	remote
Family:	General
Published:	November 23, 2009
Modified:	February 3, 2021

VPR Key Drivers

Threat Recency: No recorded events
Threat Intensity: Very Low
Exploit Code Maturity: PoC
Age of Vuln: 730 days +
Product Coverage: High
CVSSV3 Impact Score: 3.6
Threat Sources: No recorded events

Risk Information

Vulnerability Priority Rating (VPR): 6.1
Risk Factor: Medium
CVSS v3.0 Base Score 7.5
CVSS v3.0 Vector:
CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
CVSS v2.0 Base Score: 5.0
CVSS v2.0 Vector:
CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N

Vulnerability Information

Vulnerability Pub Date: August 24, 2016

Recommendation: organizations should take several proactive steps. Firstly, it is crucial to update and configure cryptographic libraries, web servers, and any network devices to disable support for cipher suites with 64-bit block sizes. This can be achieved by ensuring that only modern and secure cipher suites are enabled in the TLS/SSL configurations. Additionally, organizations should regularly monitor and assess their systems for any outdated or vulnerable protocols and promptly apply security patches and updates. Implementing strong, unique encryption keys and employing forward secrecy protocols can further bolster security. It's also recommended to utilize intrusion detection systems and perform regular security audits to identify and address any potential vulnerabilities promptly. Lastly, organizations should stay informed about emerging security threats and best practices in cryptographic protocols to proactively adapt their security measures. By taking these measures, businesses can significantly reduce the risk of falling victim to SWEET32 attacks and enhance the overall security of their digital infrastructure.

2)Vulnerability Name: **Cookie SERVERID created without secue flag**

CWE-79: Improper Neutralization of Input During Web Page Generation

OWASP Category: A07-2017: Cross-site Scripting

Description: The vulnerability allows an attacker to inject malicious code into a WordPress website, which can then be executed by visitors to the website.

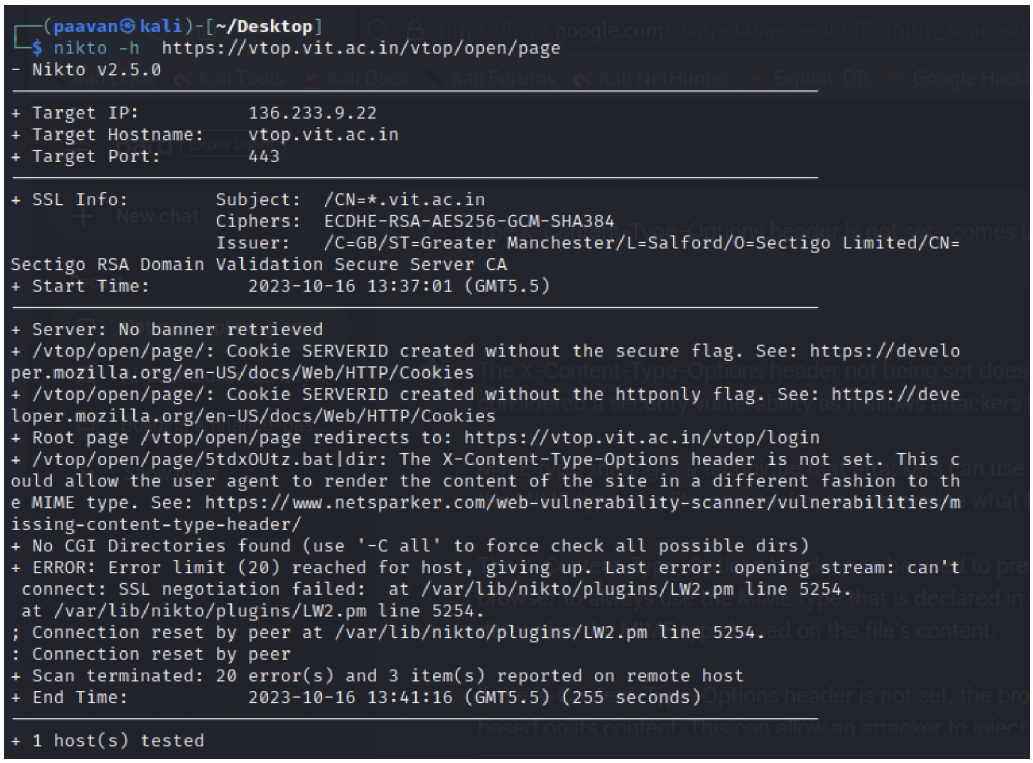
To exploit this vulnerability, an attacker would simply need to trick a user into clicking on a malicious link or opening a malicious attachment. Once the user clicks on the link or opens the attachment, the attacker's code would be executed, giving the attacker control over the user's account and potentially the entire WordPress website.

Business Impact: The business consequences of XSS can be severe, ranging from reputational damage due to compromised customer trust, to legal repercussions in cases of data breaches. Moreover, it can lead to financial losses through fraud or the cost of remediation efforts. Beyond the immediate impacts, long-term harm may occur if customers abandon the platform due to security concerns. To mitigate CWE-79, businesses must implement rigorous input validation and output encoding practices, as well as stay vigilant for emerging threats in web application security. Failure to address XSS vulnerabilities could result in significant harm to both the business's reputation and its bottom line.

Vulnerability Path : https://securityheaders.com/q=http%3A%2F%2Fvtop.vit.ac.in%2F&followRedirects=on

Vulnerability Parameter:

Steps to Reproduce:



Recommendation:

1. **Update and Patch:** Ensure that Drupal and all related modules and components are up to date with the latest security patches to mitigate known vulnerabilities.
2. **Input Validation:** Implement strong input validation and output encoding to sanitize user inputs and prevent the injection of malicious scripts.
3. **Content Security Policy (CSP):** Implement CSP headers to restrict the sources from which content can be loaded, reducing the risk of XSS.
4. **Secure Coding Practices:** Train developers in secure coding practices, emphasizing the importance of validating and escaping user inputs.
5. **Security Scanning:** Conduct regular security scans and penetration tests to identify and address XSS vulnerabilities.

3)**Vulnerability Name:** **X-Content-Type Options Header Missing**

CWE-918: Server-Side Request Forgery (SSRF)

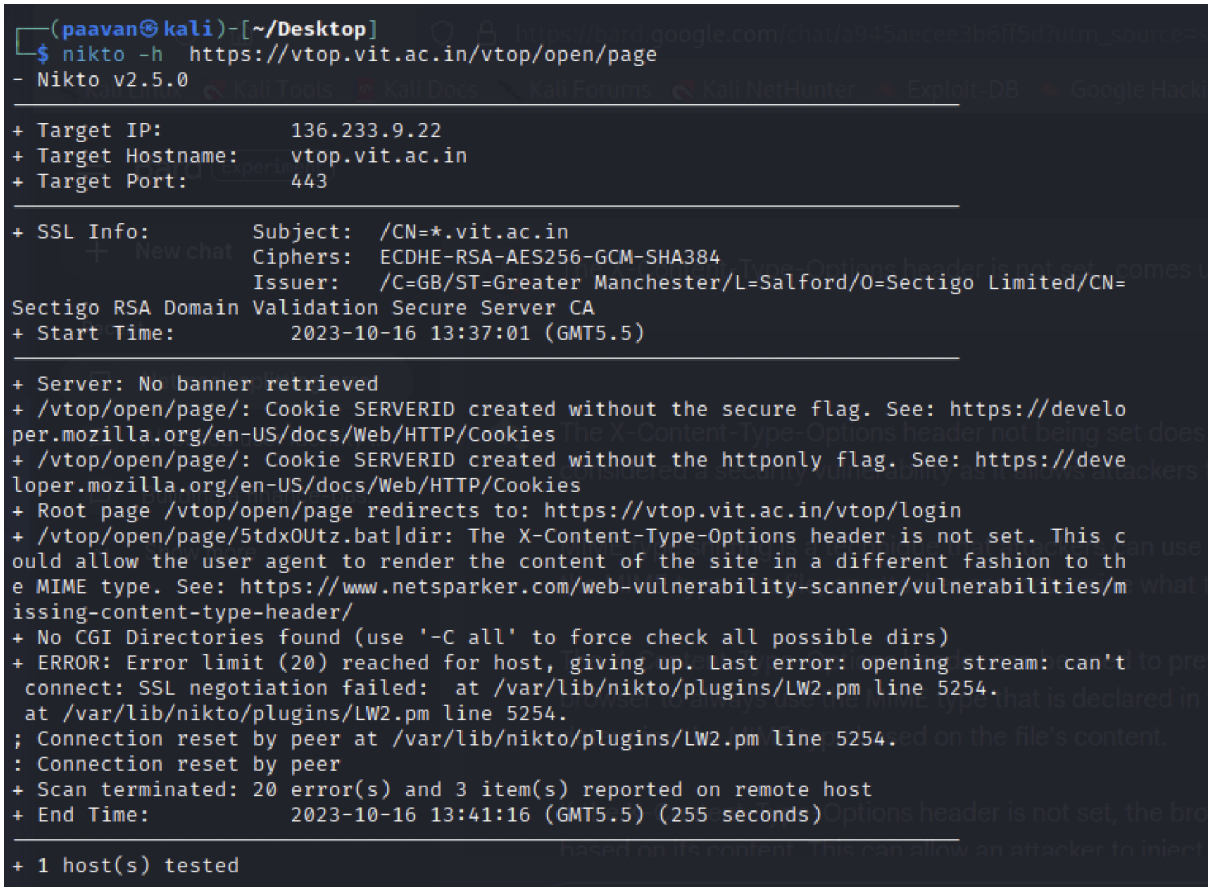
OWASP Category:A10:2021 – Server-Side Request Forgery (SSRF)

Description: The absence of the "X-Content-Type-Options" header poses a notable security risk for web applications. This header is a crucial security feature that instructs the browser to interpret files with a MIME type as declared in the Content-Type header, rather than trying to guess the type. When this header is missing, the browser may attempt to infer the content type, leaving room for potential misinterpretation and security vulnerabilities like MIME-sniffing attacks. This could enable attackers to disguise malicious content, such as scripts or executable files, as benign content, leading to Cross-Site Scripting (XSS) or other types of attacks. Implementing the "X-Content-Type-Options" header with the value "nosniff" is a fundamental security measure that helps prevent these risks. It ensures that browsers strictly adhere to the declared content type, enhancing the overall security posture of the web application and safeguarding against potential exploitation stemming from content misinterpretation.

Business Impact: The header missing impact Data Exposure(Attackers might force the server to render sensitive data as HTML, leading to data

exposure),Security Risks(Content type manipulation can increase security risks, potentially leading to breaches, data leaks, and unauthorized access), Brand and Reputation Damage(Successful attacks that exploit SSRF vulnerabilities can harm the organization's reputation and trustworthiness)

Vulnerability Path :
https://securityheaders.com/?q=http%3A%2F%2Fvtop.vit.ac.in%2F&followRedirects=on
Vulnerability Parameter:
Steps to Reproduce:



Recommendation:

- Implement "X-Content-Type-Options" Header: Configure the "X-Content-Type-Options" header with the value "nosniff" in your web server or application to prevent content type sniffing.
- Secure SSRF: To directly mitigate SSRF vulnerabilities, ensure proper input validation, access controls, and allow-list-based URL validation to prevent attackers from making unauthorized requests to internal resources.

- Security Headers: Implement other security headers like "X-Frame-Options" and "Content-Security-Policy" to further enhance security and mitigate various types of web-based attacks.

While the "X-Content-Type-Options" header may not be a direct countermeasure to SSRF, it is a valuable security control that, when configured correctly, can help protect against content type-based attacks that may be exploited in SSRF scenarios.

4)**Vulnerability Name:** **Denial of Service**

CWE -400: Uncontrolled Resource Consumption

OWASP Category:

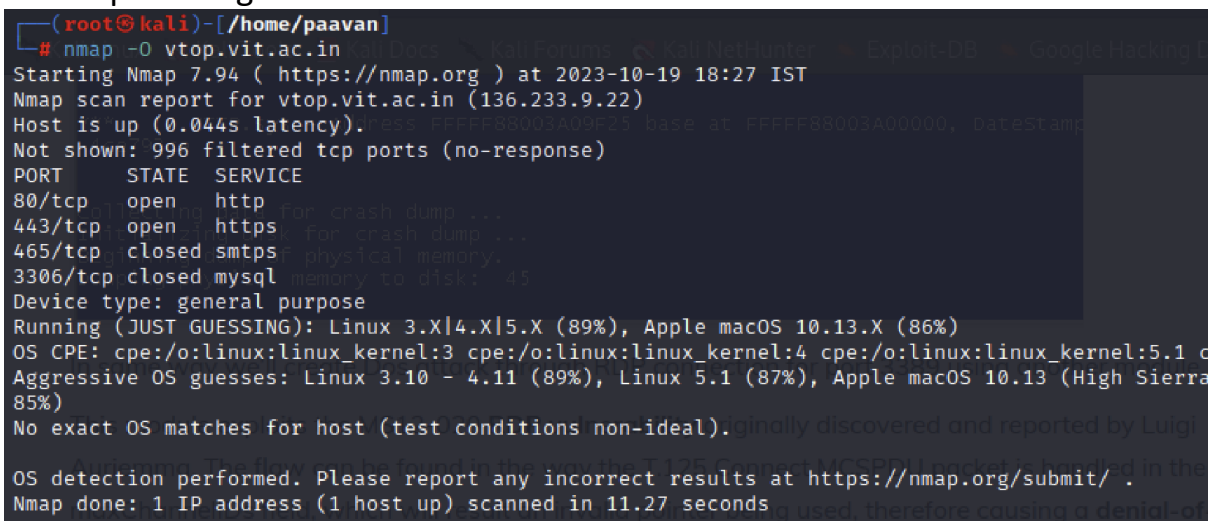
Description: The Denial of Service (DoS) attack is focused on making a resource (site, application, server) unavailable for the purpose it was designed. There are many ways to make a service unavailable for legitimate users by manipulating network packets, programming, logical, or resources handling vulnerabilities, among others. If a service receives a very large number of requests, it may cease to be available to legitimate users. In the same way, a service may stop if a programming vulnerability is exploited, or the way the service handles resources it uses.

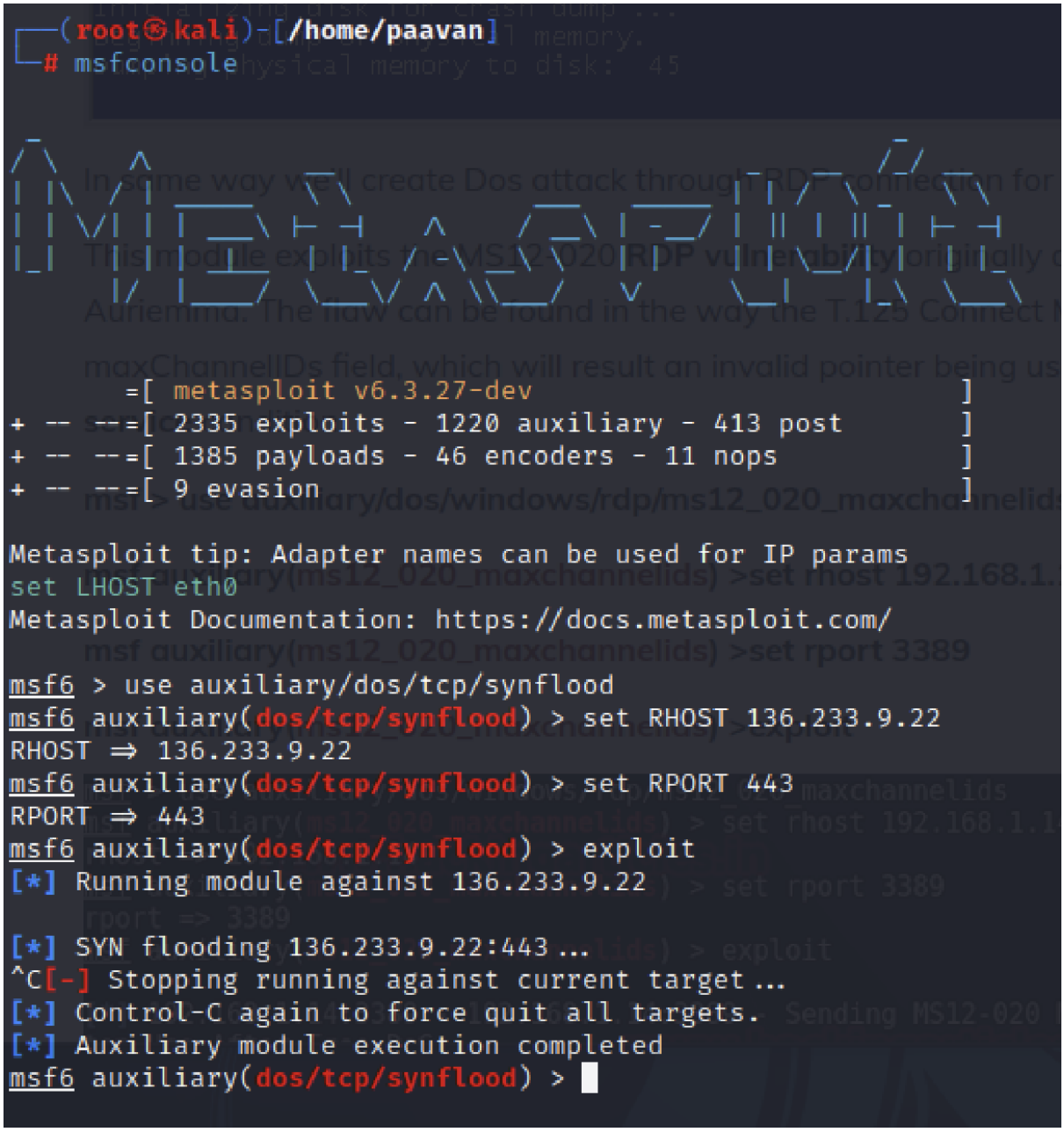
Sometimes the attacker can inject and execute arbitrary code while performing a DoS attack in order to access critical information or execute commands on the server. Denial-of-service attacks significantly degrade the service quality experienced by legitimate users. These attacks introduce large response delays, excessive losses, and service interruptions, resulting in direct impact on availability.

Business Impact: If successfully exploited, it involves overwhelming a system or application with excessive demands on its resources. This can lead to a range of disruptive consequences. The targeted system may experience significant slowdowns, or even complete unresponsiveness, making it difficult or impossible for legitimate users to access. Critical resources like CPU, memory, and network bandwidth can become depleted, causing further degradation in performance. In severe cases, the exploited vulnerability may

lead to system crashes or failures, potentially resulting in data loss. This disruption not only impacts business productivity and financial stability but also risks reputational damage, eroding customer trust and confidence. Additionally, organizations may face regulatory compliance issues if the uncontrolled resource consumption leads to violations of service level agreements or industry standards. Mitigating this vulnerability requires proactive resource management practices and a well-prepared incident response plan to minimize the potential impacts on business operations and customer satisfaction.

Vulnerability Path:
Vulnerability Parameter:
Steps to Reproduce: exploit open ports 443 to perform SQL injections, cross-site request forgeries and DDoS attacks.





Recommendation: employ a robust Web Application Firewall (WAF) capable of filtering and mitigating suspicious traffic targeting port 80. Configure it to block excessive requests or patterns indicative of an attack. Utilize load balancing techniques to distribute traffic across multiple servers, ensuring no single server becomes overwhelmed. Additionally, implement rate limiting and throttling mechanisms to restrict the number of requests a single IP address can make within a defined time frame. Regularly monitor traffic patterns and set up alerts to detect unusual spikes in activity, enabling rapid response to potential attacks. Finally, keep software and systems up-to-date with the latest security patches to address known vulnerabilities that attackers may exploit.

5)**Vulnerability Name:** **OS Identification**

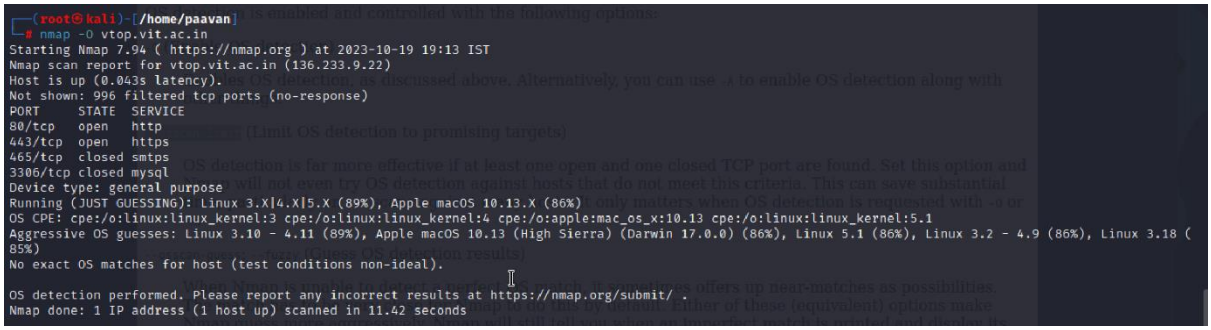
CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

OWASP Category: A02:2021-Cryptographic Failures

Description: Using a combination of remote probes (e.g., TCP/IP, SMB, HTTP, NTP, SNMP, etc.), it is possible to guess the name of the remote operating system in use. It is also possible sometimes to guess the version of the operating system.

Business Impact: If sensitive information is improperly handled or inadequately protected, it can lead to financial losses, regulatory penalties, and damage to reputation. Customers may lose trust in the organization's ability to safeguard their data, potentially resulting in a loss of business and market share. Legal consequences and compliance issues may arise, leading to additional costs. Moreover, intellectual property theft or exposure can compromise a company's competitive edge. Remediation efforts, including implementing stronger security measures and compensating affected parties, can also entail substantial expenses. Overall, CWE-200 highlights the critical importance of robust data protection measures to safeguard an organization's financial health, reputation, and long-term viability.

Vulnerability Path:
Vulnerability Parameter:
Steps to Reproduce: use nmap -O TARGET to enable os detection



Recommendation: implementing robust encryption protocols for data both in transit and at rest, coupled with stringent access controls and strong authentication measures. Regular vulnerability assessments and penetration testing should be conducted to identify and rectify potential weak points. Educating employees through comprehensive security awareness training is vital, emphasizing proper data handling practices and recognition of potential threats. Employing data masking techniques for non-production environments

and utilizing advanced Data Loss Prevention (DLP) solutions can further fortify defenses. Additionally, maintaining compliance with relevant data protection laws and regulations, as well as establishing a well-defined incident response plan, are crucial components of a comprehensive strategy to safeguard against sensitive information exposure.