

Assignment-1

Name: Ayush Aryan

Reg. No.: 21BCT0316

1. Kevin Mitnick: White Hat Hacker

Kevin Mitnick, once a notorious black hat hacker, transformed himself into a white hat consultant. After his conviction for hacking activities, Mitnick shifted his focus to cybersecurity. He now runs a consulting firm, using his expertise to help organizations bolster their defenses. His journey underscores the possibility of positive change within the hacking community and highlights the value of his insights in promoting digital security.

2. Anonymous: Gray Hat Hacktivists

Anonymous, a loosely affiliated collective of hacktivists, blurs the lines between white and

black hat hacking. Operating under the banner of anonymity, they engage in various online operations to promote social and political causes. Their activities range from advocating for free speech to exposing corruption. While their actions often challenge conventional norms, Anonymous' motives and methods can be both controversial and impactful, reminding us of the complex role hacktivism plays in the digital age.

3. Adrian Lamo: Gray Hat Hacker

Adrian Lamo, known as the "Homeless Hacker," existed in the gray area between white and black hat hacking. His actions were often ethically driven but raised ethical dilemmas themselves. Lamo gained notoriety for reporting Chelsea Manning's leak of classified documents, sparking debates about the responsibilities of hackers. His complex legacy serves as a reminder of the moral

complexities inherent in hacking and its potential consequences.

4. Albert Gonzalez: Black Hat Hacker

Albert Gonzalez is recognized as a black hat hacker responsible for orchestrating major credit card and data breaches. He engaged in cybercriminal activities that led to substantial financial losses for numerous companies.

Gonzalez's actions earned him a reputation as a proficient and malicious hacker. Eventually, he was apprehended, convicted, and sentenced for his cybercrimes.

5. Matthew Bevan and Richard Pryce: Gray Hat Hackers

Matthew Bevan and Richard Pryce, often referred to as "Kuji" and "Datastream Cowboy," respectively, gained recognition as gray hat

hackers. They were known for their involvement in hacking incidents during the 1990s. While some of their activities leaned towards black hat hacking, their motives appeared to be more curiosity-driven than malicious. Their exploits brought attention to security vulnerabilities in government and military systems, sparking discussions about the ethics and consequences of hacking. Bevan and Pryce's legacy highlights the complexities of hacking motivations and the impacts of their actions on cybersecurity awareness.

6. Jeanson James Ancheta: Black Hat Hacker

Jeanson James Ancheta earned notoriety as a black hat hacker for his involvement in creating and spreading malicious botnets. Ancheta's activities included infecting a significant number of computers and using them to carry out large-

scale attacks, such as distributed denial-of-service (DDoS) attacks and the distribution of adware. His actions demonstrated a clear intent to exploit and compromise computer systems for personal gain. Ancheta's arrest and subsequent conviction served as a reminder of the criminal nature and potential consequences of engaging in malicious hacking activities.

7. Michael Calce: Black Hat Hacker turned White Hat

Michael Calce, also known by his online moniker "Mafiaboy," initially gained infamy as a black hat hacker for launching high-profile DDoS attacks in the early 2000s. His attacks disrupted major websites, including Yahoo!, eBay, and Amazon. However, Calce's story took a turn after his arrest, as he transformed into a white hat hacker. Recognizing the negative impact of his actions, he

began working with cybersecurity experts and law enforcement agencies to help improve online security. Calce's journey from a disruptive black hat hacker to a reformed white hat consultant highlights the potential for redemption and positive contributions within the hacking community.

8. Kevin Poulsen: Black Hat Turned White Hat

Kevin Poulsen, once known as the "Dark Dante," began his hacking journey as a black hat hacker in the 1980s. His activities included taking over phone lines to win radio contests and hacking into various computer systems. However, after his legal troubles and subsequent arrest, Poulsen underwent a significant transformation. He shifted from the realm of malicious hacking to become a respected journalist and commentator on cybersecurity issues. Poulsen's unique

perspective, drawn from his hacking background, has contributed to a deeper understanding of cybersecurity challenges and the importance of ethical behavior in the digital landscape.

9. Jonathan James: Black Hat Hacker

Jonathan James gained attention as a black hat hacker in the early 2000s. He became one of the youngest individuals to be convicted for cybercrimes at the age of 16. James' most notable exploit was infiltrating systems at NASA and the U.S. Department of Defense, where he accessed sensitive information. His actions highlighted vulnerabilities in high-profile organizations' security measures. Tragically, James' life took a dark turn after his arrest, and he passed away in 2008. His story serves as a reminder of the legal and personal consequences associated with hacking activities.

10. ASTRA: Black Hat Hacker

ASTRA is the only one among the famous hackers in this list who has yet to be publicly identified. We only know that he is a Greek mathematician responsible for hacking into the Dassault Group's server. He stole weapons technology software and other relevant data, which he sold to at least 250 people across the globe. Ironically, the alias ASTRA is Sanskrit for "weapon."

Assignment-2

1. Port 20:

- Cross-site scripting
- Directory traversal attacks

2. Port 21:

- Cross-site scripting
- Directory traversal attacks

3. Port 22: Can be exploited using

- leaked SSH keys
- brute-forcing credentials.

4. Port 23:

- credential brute-forcing
- spoofing
- credential sniffing.

5. Port 25:

- Spoofing
- Spamming.

6. Port 53:

- DDoS attacks

7. Port 69:

- Remote attackers can download server files without authorization.

8. Port 80:

- SQL injections
- Cross-site request forgeries
- DDoS attacks.

9. Port 110:

- Re-usable cleartext password
- No auditing of connections

10. Port 123:

- MitM attacks

11. Port 143:

- Non-encrypted passwords

12. Port 443:

- Man-in-the-middle (MITM) attacks
- SSL/TLS vulnerabilities
- Malware infections

Assignment-3

1. CWE-284: Improper Access Control:

A01:2021-Broken Access Control

Description: The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

Business Impact: This weakness can lead to data breaches, financial losses, regulatory non-compliance, reputation damage, and intellectual property theft. Operational disruptions, legal consequences, supply chain risks, and customer trust erosion are also potential outcomes. To mitigate these impacts, businesses should implement robust access control measures, including authentication, authorization, and regular

security assessments, while maintaining vigilance through employee training and vulnerability testing.

2. CWE-324: Use of a Key Past its Expiration Date

A02:2021-Cryptographic Failures

Description: The product uses a cryptographic key or password past its expiration date, which diminishes its safety significantly by increasing the timing window for cracking attacks against that key.

Business Impact: When cryptographic keys or certificates are used beyond their expiration date as outlined in CWE-324, critical security measures become compromised, leaving sensitive data and

communications vulnerable to unauthorized access, data breaches, and cyberattacks. Such lapses can result in significant financial losses due to theft, regulatory penalties, and legal liabilities. Moreover, expired keys can disrupt operational continuity, leading to service downtime, decreased productivity, and erosion of customer trust. To mitigate these risks, businesses must implement robust key management practices, including timely key updates and renewals, to ensure the ongoing integrity and security of their cryptographic systems.

3. **CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')** :

A03:2021-Injection

Description: The product constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component.

Business Impact: CWE-78 exposes businesses to significant risks by enabling attackers to execute unauthorized commands on the underlying operating system through unvalidated input. This can lead to unauthorized data access, system compromise, and potential exfiltration of sensitive information. The impact includes financial losses due to data breaches, downtime, and recovery costs. Moreover, customer trust can erode, affecting reputation and brand value. To counter these threats, businesses must adopt

secure coding practices, input validation, and parameterized queries to prevent this type of attack and its potential ramifications.

4. **CWE-522: Insufficiently Protected Credentials:**

A04:2021-Insecure Design

Description: The product transmits or stores authentication credentials, but it uses an insecure method that is susceptible to unauthorized interception and/or retrieval.

Business Impact: CWE-522 poses significant business risks as it leaves sensitive credentials inadequately protected, allowing unauthorized access to systems and data. Attackers exploiting this vulnerability can gain control over accounts, leading to data breaches, financial losses, and

reputational damage. Stolen credentials might also lead to further attacks within the organization or against its customers. Compliance violations, legal liabilities, and disrupted operations are potential outcomes. To mitigate these risks, businesses should employ strong encryption, multi-factor authentication, and secure credential management practices, ensuring that sensitive information remains well-guarded against malicious actors.

5. **CWE-260: Password in Configuration File** :

A05:2021-Security Misconfiguration

Description: The product stores a password in a configuration file that might be accessible to actors who do not know the password.

Business Impact: CWE-260 introduces significant business risks by storing passwords and credentials in plain text within configuration files. This exposes sensitive information to unauthorized access and potential breaches, leading to data compromise and financial losses. Attackers exploiting this weakness can gain unauthorized system access, potentially disrupting operations and damaging reputation. Regulatory non-compliance, legal liabilities, and loss of customer trust are also possible consequences. To mitigate these risks, businesses should adopt secure credential storage practices, such as encryption and secure key management, to ensure the confidentiality and integrity of sensitive information in configuration files.

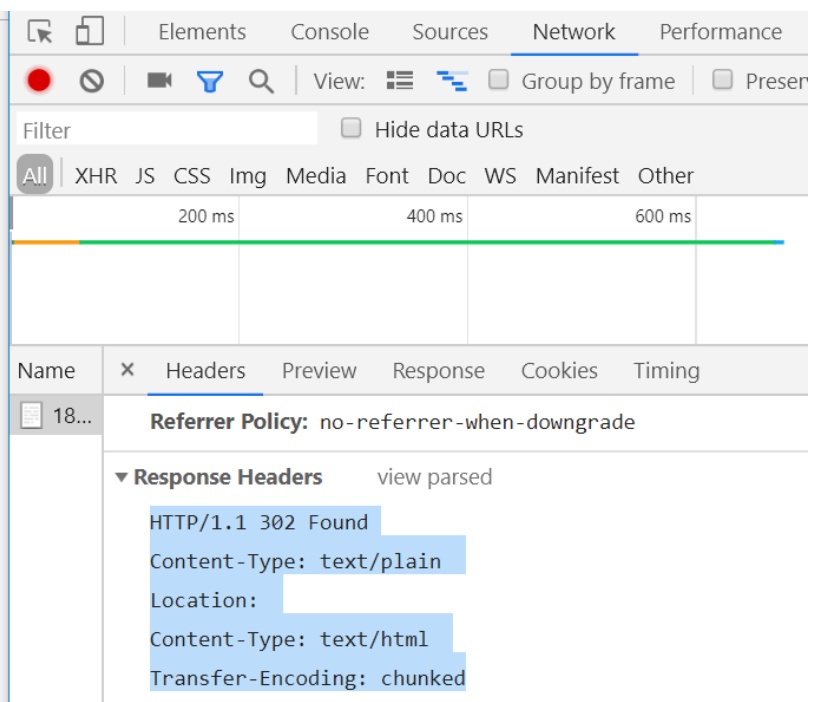
Assessment-4

1. HTTP Response Splitting:

HTTP Response Splitting is a web security flaw where attackers inject harmful content into a web application's response. This happens when the app doesn't properly handle user inputs in response headers. Attackers add line breaks or special characters to create new headers or split responses. Consequences include cache poisoning, cross-site scripting, and session attacks. To prevent this, developers should validate and sanitize user input before adding it to headers, keeping software updated and following secure coding practices.

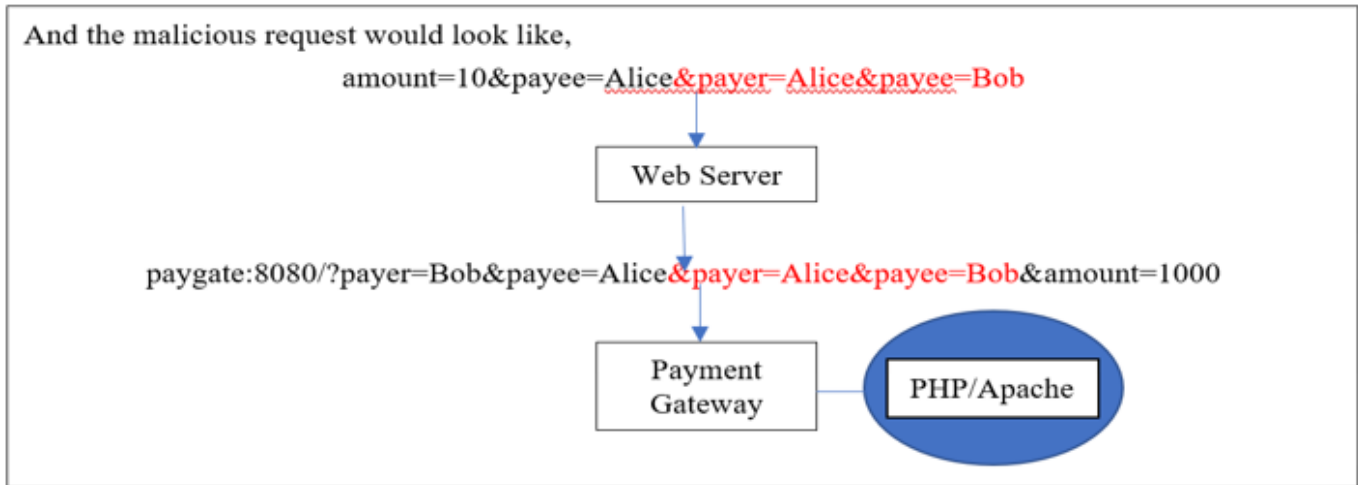
hacked

Content-Type: text/plain Date: Thu, 13 Jun 2019 16:12:20 GMT



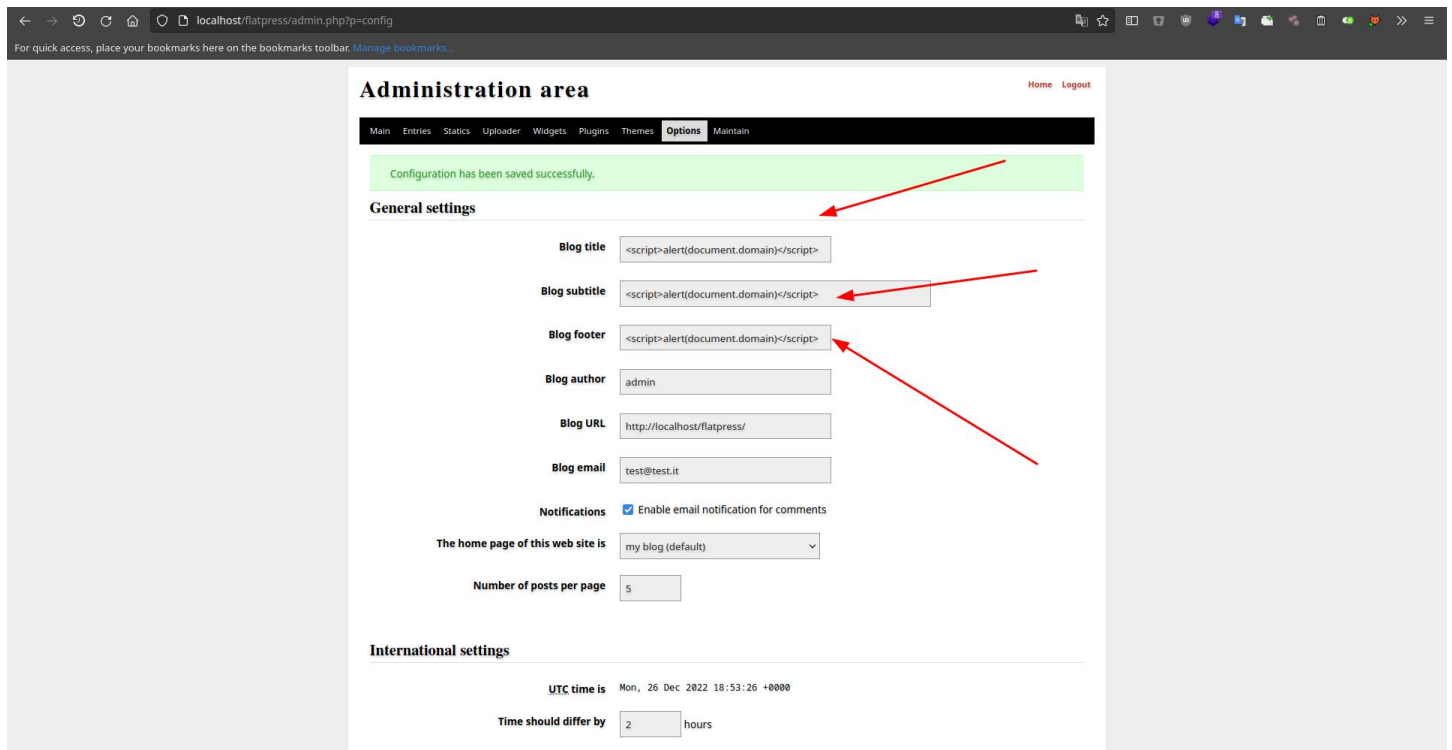
2. Parameter Delimiter:

Parameter delimiters are symbols, like the "&" sign commonly seen in URLs after the question mark "?," that play a crucial role in web applications by separating distinct parameters within the query string. These parameters, structured as key-value pairs (e.g., "key=value"), facilitate the communication of user input and preferences to the server. However, improper handling of these delimiters can result in severe security vulnerabilities. If web applications fail to adequately validate or sanitize user inputs within these parameters, attackers might exploit the situation to conduct various attacks such as SQL injection or cross-site scripting (XSS). To mitigate these risks, developers need to implement robust input validation and sanitization mechanisms, ensuring that user-provided data is rigorously examined and cleansed before processing. By doing so, the potential for security breaches arising from mishandled parameter delimiters can be significantly reduced.



3. XSS in subtitle:

Parameter delimiters, such as the "&" sign in URLs, are pivotal for web apps to separate parameters within query strings. Yet, careless handling can lead to serious security vulnerabilities, including the notorious cross-site scripting (XSS) attack. If user inputs within these parameters aren't properly validated or sanitized, attackers might exploit this opening to inject malicious scripts. Developers must diligently validate and sanitize user inputs to thwart these risks and ensure that their applications remain safeguarded against XSS and other potential threats stemming from mishandled parameter delimiters.

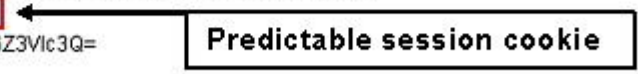


4. Password Spraying:

A password spraying attack strategically targets weak passwords by trying a few common ones across numerous accounts, evading immediate detection. Exploiting users' inclination towards predictable passwords, attackers employ this method. Organizations can bolster their defenses by enforcing strong password policies, implementing multi-factor authentication, closely monitoring login patterns for anomalies, and educating users about robust password practices.

practices, employing HTTPS, and implementing mechanisms to detect and respond to suspicious activities. These steps safeguard user sessions and thwart unauthorized access.

```
GET http://janaina:8180/WebGoat/attack?Screen=17&menu=410 HTTP/1.1
Host: janaina:8180
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
Referer: http://janaina:8180/WebGoat/attack?Screen=17&menu=410
Cookie: JSESSIONID=user01
Authorization: Basic Z3Vlc3Q6Z3Vlc3Q=
```



A diagram consisting of a black arrow pointing from a box labeled "Predictable session cookie" to the "Cookie: JSESSIONID=user01" line in the HTTP request. The "Cookie: JSESSIONID=user01" line is highlighted with a red rectangular border.