

- Q1.** कौन सी ओएसआई लेयर आईपी सेक एच (ऑथेंटिकेशन हेडर) सर्विस के लिए सबसे अच्छी है?
- (A) डेटा लिंक
 - (B) नेटवर्क
 - (C) ट्रांसपोर्ट
 - (D) सेशन

Which OSI layer best corresponds to the IPsec AH (Authentication Header) service?

- (A) Data Link
- (B) Network
- (C) Transport
- (D) Session

- Q2.** आइपीसेक टनल मोड का इस्तेमाल करके ठीक से कॉन्फिगर किए गए वीपीएन में, ओरिजिनल पैकेट का कौन सा हिस्सा इंटरमीडिएट राउटर से छिपा होता है?
- (A) ओरिजिनल आइपी हेडर
 - (B) उपयोग किया गया एन्क्रिप्शन एल्गोरिदम
 - (C) आउटर आइपी हेडर
 - (D) डेस्टिनेशन पोर्ट नंबर

In a properly configured VPN using IPsec tunnel mode, which part of the original packet is hidden from intermediate routers?

- (A) Original IP headers
- (B) Encryption algorithm used
- (C) Outer IP headers
- (D) Destination port numbers

- Q3.** कौन सी खासियत स्मर्फ अटैक को फ्रैगल अटैक से अलग करती है?
- (A) स्मर्फ यूडीपी उपयोग करता है, फ्रैगल आइसीएमपी उपयोग करता है
 - (B) स्मर्फ आइसीएमपी उपयोग करता है, फ्रैगल यूडीपी उपयोग करता है
 - (C) स्मर्फ जाली एआरपी रिप्लाई उपयोग करता है, फ्रैगल डीएचसीपी उपयोग करता है
 - (D) स्मर्फ राउटर को टारगेट करता है, फ्रैगल स्विच को टारगेट करता है

Which characteristic distinguishes a smurf attack from a fraggle attack?

- (A) Smurf uses UDP, fraggle uses ICMP
- (B) Smurf uses ICMP, fraggle uses UDP
- (C) Smurf uses forged ARP replies, fraggle uses DHCP
- (D) Smurf targets routers, fraggle targets switches

Q4. एक इनलाइन आइपीएस एक पैसिव आइडीएस से अलग होता है क्योंकि यह:

- (A) सिर्फ गड़बड़ियों का पता लगाता है
- (B) आउट-ऑफ-बैंड काम करता है
- (C) रियल-टाइम में ट्रैफिक को ब्लॉक या बदल सकता है
- (D) एन्क्रिप्टेड डेटा को इंस्पेक्ट नहीं कर सकता

An inline IPS differs from a passive IDS because it:

- (A) Detects anomalies only
- (B) Operates out-of-band
- (C) Can block or modify traffic in real-time
- (D) Cannot inspect encrypted data

Q5. इनमें से कौन सा आइपीसेक पैरामीटर सीधे रीप्ले अटैक को रोकता है?

- (A) सिक्योरिटी पैरामीटर इंडेक्स (एसपीआई)
- (B) डिफ़ी-हेलमैन ग्रुप
- (C) सीकेंस नंबर
- (D) इनिशियलाइज़ेशन वेक्टर (आईवी)

Which of the following IPSec parameters directly prevents replay attacks?

- (A) Security Parameter Index (SPI)
- (B) Diffie–Hellman Group
- (C) Sequence Numbers
- (D) Initialization Vector (IV)

Q6. डब्ल्यूपीए3 में, साइमलटेनियस ऑथेंटिकेशन ऑफ़ इक्वल्स (एसएइ) हैंडशेक ऑफ़लाइन डिक्शनरी अटैक का विरोध करता है क्योंकि:

- (A) यह कुछ समय के लिए वैल्यू वाले पासवर्ड-ऑथेंटिकेटेड की एक्सचेंज का इस्तेमाल करता है
- (B) पासवर्ड एसएचए-512 से हैश किए जाते हैं
- (C) ऑथेंटिकेशन से पहले सभी फ्रेम एन्क्रिप्ट किए जाते हैं
- (D) यह क्लाइंट डिवाइस पर सॉल्ट स्टोर करता है

In WPA3, the Simultaneous Authentication of Equals (SAE) handshake resists offline dictionary attacks because:

- (A) It uses a password-authenticated key exchange with ephemeral values
- (B) Passwords are hashed with SHA-512
- (C) All frames are encrypted before authentication
- (D) It stores salts on the client device

Q7. इनमें से कौन सा मोनोलिथिक फर्मवेयर को मॉड्यूलर फर्मवेयर आर्किटेक्चर से सबसे अच्छे तरीके से अलग करता है?

- (A) मोनोलिथिक फर्मवेयर बूटलोडर को कर्नल से अलग स्टोर करता है
- (B) मोनोलिथिक फर्मवेयर हमेशा माइक्रोकंट्रोलर पर चलता है
- (C) मॉड्यूलर फर्मवेयर कंपोनेंट को अलग-अलग अपडेट नहीं कर सकता
- (D) मॉड्यूलर फर्मवेयर कंपोनेंट को डायनामिक रूप से लोड करता है जबकि मोनोलिथिक फर्मवेयर सभी कंपोनेंट को एक इमेज में एम्बेड करता है

Which of the following best differentiates monolithic firmware from modular firmware architectures?

- (A) Monolithic firmware stores bootloader separately from kernel
- (B) Monolithic firmware always runs on microcontrollers
- (C) Modular firmware cannot update components individually
- (D) Modular firmware loads components dynamically while monolithic firmware embeds all components in one image

Q8. यूइएफआइ सिक्वोर बूट में, "dbx" डेटाबेस स्टोर करता है:

- (A) अलाउड बूटलोडर के हैश
- (B) साइन करने के लिए प्राइवेट की
- (C) रद्द या ब्लैकलिस्ट किए गए सर्टिफिकेट
- (D) रिकवरी इमेज

In UEFI Secure Boot, the "dbx" database stores:

- (A) Hashes of allowed bootloaders
- (B) Private keys for signing
- (C) Certificates revoked or blacklisted
- (D) Recovery images

Q9. अगर फर्मवेयर का डीएमए कंट्रोलर गलत तरीके से प्रोग्राम किया गया है, तो किस तरह की मेमोरी करप्शन वलनरेबिलिटी की सबसे ज्यादा संभावना है?

- (A) आर्बिट्ररी फिजिकल मेमोरी ओवरराइट
- (B) इंटीजर ओवरफ्लो
- (C) यूज़-आफ्टर-फ्री
- (D) स्टैक पिवट

Which type of memory corruption vulnerability is most likely if a firmware's DMA controller is improperly programmed?

- (A) Arbitrary physical memory overwrite
- (B) Integer overflow
- (C) Use-after-free
- (D) Stack pivot

Q10. फर्मवेयर अपडेट मैकेनिज्म में, इनमें से कौन सा तरीका सप्लाइ-चेन रोलबैक अटैक को सबसे सीधे तौर पर रोकता है?

- (A) फर्मवेयर को एडएस-256 से एन्क्रिप्ट करना
- (B) फर्मवेयर को एनएएनडी फ्लैश पर स्टोर करना
- (C) फर्मवेयर को उसी प्राइवेट की से साइन करना
- (D) सिग्नोर स्टोरेज से जुड़े मोनोटोनिक एंटी-रोलबैक काउंटर का उपयोग करना

In a firmware update mechanism, which of the following most directly prevents a supply-chain rollback attack?

- (A) Encrypting the firmware with AES-256
- (B) Storing the firmware on NAND flash
- (C) Signing firmware with the same private key
- (D) Using a monotonic anti-rollback counter tied to secure storage

Q.11 सिग्नोर फर्मवेयर अपडेट के दौरान, कौन सा मैकेनिज्म मुख्य रूप से यह पक्का करता है कि गलत तरीके से मॉडिफाइड लेकिन सही तरीके से साइन की गई इमेज एक्सेप्ट नहीं की जाती है, अगर वह किसी दूसरे हार्डवेयर प्लेटफॉर्म को टारगेट करती है?

- (A) फर्मवेयर पेलोड का सिमेट्रिक एन्क्रिप्शन
- (B) हार्डवेयर-बाउंड पब्लिक की वेरिफिकेशन
- (C) फर्मवेयर हेडर की सिग्नोर हैशिंग
- (D) फर्मवेयर में प्लेटफॉर्म आईडी का ऑबफस्केशन

During a secure firmware update, which mechanism primarily ensures that a maliciously modified but correctly signed image is not accepted if it targets a different hardware platform?

- (A) Symmetric encryption of the firmware payload
- (B) Hardware-bound public key verification
- (C) Secure hashing of the firmware header
- (D) Obfuscation of platform ID in the firmware

Q12. एक मैनुफैक्चरर डिवाइस के एनएएनडी फ्लैश पर फर्मवेयर एन्क्रिप्शन की स्टोर करने का फैसला करता है। सिग्नोरिटी बनाए रखने के लिए कौन सा एडिशनल मैकेनिज्म सबसे ज़रूरी है?

- (A) की को सील करने के लिए सिग्नोर एन्क्लेव या टीपीएम का उपयोग करना
- (B) की को प्लेनटेक्स्ट में लेकिन अस्पष्ट रूप से स्टोर करना
- (C) फर्मवेयर इमेज के साथ की को एन्क्रिप्ट करना
- (D) की लोकेशन की अस्पष्टता पर निर्भर रहना

A manufacturer decides to store the firmware encryption key on the device's NAND flash. Which additional mechanism is most critical to maintain security?

- (A) Using Secure Enclave or TPM to seal the key
- (B) Storing key in plaintext but obfuscated
- (C) Encrypting the key with the firmware image
- (D) Relying on obscurity of key location

- Q13.** एक अटैकर सर्च पेज पर एक गलत लिंक बनाता है। जब कोई लॉग-इन एडमिन लिंक पर क्लिक करता है, तो एक छिपा हुआ रिक्वेस्ट यूज़र रोल बदल देता है। यह _____ है।
- (A) एक्सएसएस
 - (B) एसक्यूएल इंजेक्शन
 - (C) सीएसआरएफ
 - (D) क्लिकजैकिंग

An attacker crafts a malicious link to a search page. When a logged-in admin clicks the link, a hidden request changes user roles. This is _____.

- (A) XSS
- (B) SQL Injection
- (C) CSRF
- (D) Clickjacking

- Q14.** मल्टी-टेनेंट ऐप में, एक टेनेंट एडमिन जावास्क्रिप्ट फ़ाइलें अपलोड कर सकता है जो एक शेयर्ड सीडीएन डोमेन के अंतर्गत सर्व की जाएंगी। यह दूसरे टेनेंट्स के लिए कौन सी एक्सएसएस क्लास बनाता है?
- (A) स्टोर्ड एक्सएसएस
 - (B) रिफ्लेक्टेड एक्सएसएस
 - (C) सिर्फ़ सेल्फ़-एक्सएसएस
 - (D) क्लिकजैकिंग

In a multi-tenant app, a tenant admin can upload JavaScript files that will be served under a shared CDN domain. What XSS class does this create for other tenants?

- (A) Stored XSS
- (B) Reflected XSS
- (C) Self-XSS only
- (D) Clickjacking

- Q15.** जेएसओएन वेब टोकन (जेडब्ल्यूटी) का उपयोग करने वाले एपीआई में, कौन सा कॉन्फ़िगरेशन इंटीग्रिटी प्रोटेक्शन को सबसे ज़्यादा कमज़ोर करता है?
- (A) काफ़ी मज़बूत सीक्रेट के साथ एचएस256 का उपयोग करना
 - (B) साइनिंग कीज़ को समय-समय पर रोटेट करना
 - (C) टोकन में कम से कम क्लेम शामिल करना
 - (D) एल्गोरिदम को "RS256" से "none" में स्विच करने देना

In an API using JSON Web Tokens (JWTs), which configuration most undermines integrity protection?

- (A) Using HS256 with a sufficiently strong secret
- (B) Rotating signing keys periodically
- (C) Including minimal claims in the token
- (D) Allowing algorithm switching from "RS256" to "none"

- Q16. एप्पल का सिक्वोर एन्क्लेव मुख्य रूप से _____ के लिए डिज़ाइन किया गया है।
- (A) सैंडबॉक्स थर्ड-पार्टी ऐप्स
 - (B) बायोमेट्रिक क्रेडेंशियल्स जैसे सेंसिटिव डेटा को सुरक्षित रूप से स्टोर और प्रोसेस करने
 - (C) पूरे फ़ाइल सिस्टम को एन्क्रिप्ट करने
 - (D) आइक्लाउड बैकअप प्रबंधित करने

Apple's Secure Enclave is primarily designed to:

- (A) Sandbox third-party apps
- (B) Store and process sensitive data like biometric credentials securely
- (C) Encrypt the entire file system
- (D) Manage iCloud backups

- Q17. कौन सा अटैक खास तौर पर मोबाइल ऐप्स में इनसिक्योर डीप लिंकिंग को टारगेट करता है?
- (A) इंटेन्ट स्पूफिंग
 - (B) फिशिंग
 - (C) टैपजैकिंग
 - (D) एसक्यूएल इंजेक्शन

Which attack specifically targets insecure deep linking in mobile apps?

- (A) Intent spoofing
- (B) Phishing
- (C) Tapjacking
- (D) SQL injection

- Q18. एपआर्मर, सेलिनक्स से मुख्य रूप से _____ से अलग है।
- (A) नेटवर्क ऑपरेशन की कंफाइनेमेंट को सपोर्ट नहीं करने
 - (B) सिर्फ कर्नेल मॉड्यूल लेवल पर पॉलिसी लागू करने
 - (C) पॉलिसी के बजाय कैपेबिलिटी का इस्तेमाल करने
 - (D) पॉलिसी लागू करने के लिए लेबल के बजाय फ़ाइल पाथ का इस्तेमाल करने

AppArmor differs from SELinux primarily by:

- (A) Not supporting confinement of network operations
- (B) Enforcing policies at the kernel module level only
- (C) Using capabilities instead of policies
- (D) Using file paths rather than labels for policy enforcement

- Q19.** आरएसए में, पब्लिक एक्सपोनेंट e की कौन सी प्रॉपर्टी एफिशिएंट एन्क्रिप्शन पक्का करती है लेकिन अगर सही तरीके से चुना जाए तो सिक्योरिटी भी देती है?
- (A) यह हमेशा प्राइम होता है और 17 के बराबर होता है
 - (B) यह आमतौर पर छोटा होता है (जैसे 65537) और $\varphi(n)$ का कोप्राइम होता है
 - (C) यह प्राइवेट एक्सपोनेंट d के बराबर होता है
 - (D) यह एक रैंडम 4096-बिट इंटीजर है

In RSA, what property of the public exponent e ensures efficient encryption but still security if chosen properly?

- (A) It is always prime and equal to 17
- (B) It is typically small (like 65537) and coprime to $\varphi(n)$
- (C) It equals the private exponent d
- (D) It is a random 4096-bit integer

- Q20.** एइएस अपने ट्रांसफॉर्मेशन करने के लिए अंदर से इनमें से किस स्ट्रक्चर का उपयोग करता है?
- (A) फीस्टल नेटवर्क
 - (B) लीनियर फीडबैक शिफ्ट रजिस्टर
 - (C) सब्स्टिट्यूशन-परम्यूटेशन नेटवर्क
 - (D) मर्कल-डेमगार्ड कंस्ट्रक्शन

AES uses which of the following structures internally to perform its transformations?

- (A) Feistel network
- (B) Linear feedback shift register
- (C) Substitution–permutation network
- (D) Merkle–Damgård construction

- Q21.** क्रिप्टोग्राफिक हैश फंक्शन की कौन सी प्रॉपर्टी यह पक्का करती है कि दिए गए हैश आउटपुट के साथ, ओरिजिनल इनपुट का पता लगाना मुमकिन नहीं है?
- (A) प्रीइमेज रेजिस्टेंस
 - (B) कोलिजन रेजिस्टेंस
 - (C) सेकंड प्रीइमेज रेजिस्टेंस
 - (D) एवलांच इफेक्ट

Which property of a cryptographic hash function ensures that given a hash output, it's infeasible to determine the original input?

- (A) Preimage resistance
- (B) Collision resistance
- (C) Second preimage resistance
- (D) Avalanche effect

Q22. एंटरप्राइज़ की मैनेजमेंट में, "की एस्करो" मैकेनिज्म का उपयोग आम तौर पर _____ के लिए किया जाता है।

- (A) कीज़ के ऑटोमैटिक रोटेशन की अनुमति देने
- (B) डिपार्टमेंट्स के बीच की शेयरिंग को रोकने
- (C) एन्क्रिप्शन को तेज़ करने
- (D) कड़े कंट्रोल में एन्क्रिप्टेड डेटा की ऑथराइज़्ड रिकवरी की अनुमति देने

In enterprise key management, a "key escrow" mechanism is typically used to:

- (A) Allow automatic rotation of keys
- (B) Prevent key sharing between departments
- (C) Speed up encryption
- (D) Allow authorized recovery of encrypted data under strict controls

Q23. कौन सा डिजिटल सिग्नेचर एल्गोरिदम डिस्क्रीट लॉगरिदम प्रॉब्लम पर निर्भर करता है?

- (A) आरएसए (रिवेस्ट-शमीर-एडलमैन)
- (B) डीएसए (डिजिटल सिग्नेचर एल्गोरिदम)
- (C) एडएस (एडवांस्ड एन्क्रिप्शन स्टैंडर्ड)
- (D) एमडी 5 (मैसेज-डाइजेस्ट एल्गोरिदम 5)

Which digital signature algorithm relies on the discrete logarithm problem?

- (A) RSA (Rivest–Shamir–Adleman)
- (B) DSA (Digital Signature Algorithm)
- (C) AES (Advanced Encryption Standard)
- (D) MD5 (Message-Digest Algorithm 5)

Q24. पीकेआई में, एक इंटरमीडिएट सीए का उपयोग _____ के लिए किया जाता है।

- (A) रूट सीए पर भरोसा कम करने
- (B) एंड-यूज़र्स के लिए सिमेट्रिक कीज़ बनाने
- (C) रूट सीए की प्राइवेट की को एक्सपोज़ किए बिना सर्टिफिकेट जारी करने का काम सौंपने
- (D) रूट और एंड-एंटिटी के बीच कम्युनिकेशन को एन्क्रिप्ट करने

In a PKI, an intermediate CA is used to:

- (A) Reduce trust in the root CA
- (B) Generate symmetric keys for end-users
- (C) Delegate certificate issuance without exposing the root CA's private key
- (D) Encrypt communications between root and end-entity

Q25. “फ़ाइललेस वर्म्स” डिस्क पर फ़ाइलें लिखे बिना सिस्टम में कैसे बने रहते हैं?

- (A) अटैचमेंट के साथ ईमेल भेजकर
- (B) C ड्राइव पर छिपी हुई फ़ाइलें बनाकर
- (C) सिर्फ़ रजिस्ट्री एंट्री में बदलाव करके
- (D) ओएस मेमोरी-रेसिडेंट सर्विस और शेड्यूल किए गए कामों का फ़ायदा उठाकर

How do “fileless worms” maintain persistence in a system without writing files to disk?

- (A) By sending emails with attachments
- (B) By creating hidden files on C drive
- (C) By modifying registry entries only
- (D) By exploiting OS memory-resident services and scheduled tasks

Q26. एक अटैकर माइक्रो सोशल इंजीनियरिंग का उपयोग करता है: किसी बड़ी रिक्वेस्ट से महीनों पहले भरोसा बनाने के लिए रोज़ाना कम रिस्क वाले सवाल पूछता है। कौन सा इनसाइडर रिस्क प्रोग्राम एलिमेंट इस स्लो बर्न अप्रोच को सबसे अच्छे से डिटेक्ट करता है?

- (A) सिर्फ़ सिंगल इवेंट एनॉमली डिटेक्शन
- (B) लॉन्गिट्यूडिनल बिहेवियर एनालिटिक्स जो समय के साथ असामान्य रिक्वेस्ट या प्रिविलेज के इस्तेमाल में धीरे-धीरे बढ़ोतरी का पता लगाता है
- (C) सिर्फ़ बहुत ज़्यादा एनॉमलस इवेंट्स पर नज़र रखता है
- (D) सभी बाहरी कम्युनिकेशन को ब्लॉक करता है

An attacker uses micro-social engineering: brief daily low-risk asks to build trust over months before a big request. Which insider-risk program element best detects this slow-burn approach?

- (A) Single-event anomaly detection only
- (B) Longitudinal behavior analytics that detect gradual escalation in unusual requests or privilege use over time
- (C) Only monitor for extreme anomalous events
- (D) Block all external communications

Q27. कौन सी चीज़ एपीटी लैटरल मूवमेंट को आम मैलवेयर प्रोपेगेशन से अलग करती है?

- (A) बिना टारगेट किए किसी भी उपलब्ध होस्ट पर फैलना
- (B) सिर्फ़ बाहरी सर्वर को इन्फेक्ट करना
- (C) सभी मिली हुई फ़ाइलों को एन्क्रिप्ट करना
- (D) स्ट्रेटेजिक वैल्यू के लिए खास होस्ट, एप्लिकेशन या प्रिविलेज्ड अकाउंट को टारगेट करना

Which element distinguishes APT lateral movement from ordinary malware propagation?

- (A) Spreading to any available host without targeting
- (B) Infecting only external-facing servers
- (C) Encrypting all discovered files
- (D) Targeting specific hosts, applications, or privileged accounts for strategic value

Q28. एपीआइ डिटेक्शन में "ड्वेल टाइम" एक ज़रूरी मेट्रिक क्यों है?

- (A) यह मापता है कि डिस्क पर मैलवेयर फ़ाइलें कितनी देर तक रहती हैं
- (B) यह कमज़ोरियों के लिए स्कैनिंग में लगने वाले समय को ट्रैक करता है
- (C) यह उस समय को दिखाता है जब तक कोई अटैकर पकड़ा नहीं जाता, जो संभावित डेटा लॉस से जुड़ा होता है
- (D) यह मैलवेयर द्वारा सीपीयू के उपयोग को मापता है

Why is "dwell time" an important metric in APT detection?

- (A) It measures how long malware files exist on disk
- (B) It tracks time spent scanning for vulnerabilities
- (C) It represents the duration an attacker remains undetected, which correlates with potential data loss
- (D) It measures CPU usage by malware

Q29. डीएनएस एम्प्लीफिकेशन डीडीओएस में, एम्प्लीफिकेशन इफ़ेक्ट मुख्य रूप से _____ की वजह से होता है।

- (A) छोटी क्वेरीज़ की तुलना में बड़े यूडीपी रिस्पॉन्स
- (B) इन्फेक्टेड होस्ट्स की ज़्यादा संख्या
- (C) टीसीपी थ्री-वे हैंडशेक
- (D) एचटीटीपी गेट रिक्वेस्ट

In a DNS amplification DDoS, the amplification effect is primarily due to:

- (A) Large UDP responses compared to small queries
- (B) High number of infected hosts
- (C) TCP three-way handshake
- (D) HTTP GET requests

Q30. कौन सा डीडीओएस टाइप नेटवर्क बैंडविड्थ के बजाय सीधे सर्वर स्टेट टेबल को टारगेट करता है?

- (A) यूडीपी फ्लड
- (B) एसवाइएन फ्लड
- (C) डीएनएस रिफ्लेक्शन
- (D) आइसीएमपी फ्लड

Which DDoS type directly targets server state tables rather than network bandwidth?

- (A) UDP flood
- (B) SYN flood
- (C) DNS reflection
- (D) ICMP flood

Q31. थ्रेट मॉडलिंग के दौरान, मॉडर्न फ्रेमवर्क में ड्रेड का उपयोग कम क्यों माना जाता है?

- (A) इसकी कैटेगरी स्ट्राइड के साथ बेकार हैं
- (B) ड्रेड अंदरूनी खतरों का मॉडल नहीं बना सकता
- (C) इसमें कम करने की स्ट्रेटेजी शामिल नहीं हैं
- (D) स्कोरिंग में सब्जेक्टिविटी से रिस्क प्रायोरिटी में अंतर हो सकता है

During threat modeling, why is DREAD considered less commonly used in modern frameworks?

- (A) Its categories are redundant with STRIDE
- (B) DREAD cannot model insider threats
- (C) It does not include mitigation strategies
- (D) Subjectivity in scoring can lead to inconsistent risk prioritization

Q32. थ्रेट मॉडलिंग के दौरान, स्ट्राइड को डेटा फ्लो डायग्राम (डीएफडी) के साथ पेयर करने से मुख्य रूप से _____ में मदद मिलती है।

- (A) अपने आप कमजोरियों को पैच करने
- (B) हर सिस्टम कंपोनेंट और डेटा फ्लो के लिए संभावित खतरों को मैप करने
- (C) ड्रेड स्कोरिंग को बदलना
- (D) जीरो-डे एक्सप्लॉइट का पता लगाना

During threat modeling, pairing STRIDE with Data Flow Diagrams (DFDs) primarily helps to:

- (A) Automatically patch vulnerabilities
- (B) Map potential threats to each system component and data flow
- (C) Replace DREAD scoring
- (D) Detect zero-day exploits

Q33. ड्रेड (DREAD) मॉडल में, "R" फैक्टर मुख्य रूप से _____ का आकलन करता है।

- (A) एक सफल एक्सप्लॉइट से होने वाला संभावित रेप्युटेशनल नुकसान
- (B) कॉम्प्रोमाइज़ के बाद सिस्टम को कितनी जल्दी रिस्टोर किया जा सकता है
- (C) एक बार पता चलने के बाद किसी अटैक को कितने भरोसे और लगातार तरीके से दोहराया जा सकता है
- (D) अगर वल्नरेबिलिटी का एक्सप्लॉइट किया जाता है तो होने वाला फ़ाइनेंशियल नुकसान

In the DREAD model, the "R" factor primarily assesses:

- (A) The potential reputational harm caused by a successful exploit
- (B) How quickly systems can be restored after compromise
- (C) How reliably and consistently an attack can be repeated once it is known
- (D) The expected financial loss if the vulnerability is exploited

- Q34.** ऑटोमेटेड सिक््योरिटी कंट्रोल में थ्रेट इंटेलिजेंस को इंटीग्रेट करते समय कौन सी चुनौती सबसे आम है?
- (A) इंटरनेट कनेक्शन में हाई लेटेंसी
 - (B) इनकंसिस्टेंट डेटा फॉर्मेट और कॉन्टेक्चुअल जानकारी
 - (C) बहुत ज़्यादा बैंडविड्थ यूसेज
 - (D) घटता एंडपॉइंट स्टोरेज

Which challenge is most common when integrating threat intelligence into automated security controls?

- (A) High latency in internet connections
- (B) Inconsistent data formats and contextual information
- (C) Excessive bandwidth usage
- (D) Decreasing endpoint storage

- Q35.** इनमें से कौन सा थ्रेट इंटेलिजेंस साइकिल को सबसे अच्छे से दिखाता है?
- (A) इकट्ठा करना, एनालाइज़ करना, शेयर करना, जवाब देना
 - (B) स्कैन करना, पैच करना, बैकअप लेना, ऑडिट करना
 - (C) एन्क्रिप्ट करना, डिक्लिप्ट करना, आर्काइव करना, डिलीट करना
 - (D) पहचानना, धोखा देना, एक्सप्लॉइट करना, रिपोर्ट करना

Which of the following best reflects a complete threat intelligence cycle?

- (A) Gather, Analyze, Share, Respond
- (B) Scan, Patch, Backup, Audit
- (C) Encrypt, Decrypt, Archive, Delete
- (D) Identify, Spoof, Exploit, Report

- Q36.** ज़ीरो-डे वलनरेबिलिटी का फ़ायदा उठाने वाले अटैक का पता चलने के बाद, टीम को बिना वेंडर पैच के जवाब देना होगा। कौन सी मिटिगेशन स्ट्रेटेजी सबसे असरदार है?
- (A) नेटवर्क सेगमेंटेशन लागू करें और कम्पेनसेटिंग कंट्रोल लागू करें
 - (B) मैलवेयर हटाने के लिए सभी अफेक्टेड सिस्टम को रीइंस्टॉल करें
 - (C) सिस्टम को प्रोडक्शन एनवायरनमेंट से पूरी तरह डिस्कनेक्ट करें
 - (D) एक्शन लेने से पहले वेंडर पैच का इंतज़ार करें

After detecting an attack exploiting a zero-day vulnerability, the team must respond without a vendor patch available. Which mitigation strategy is most effective?

- (A) Apply network segmentation and implement compensating controls
- (B) Reinstall all affected systems to remove malware
- (C) Disconnect systems entirely from production environment
- (D) Wait for the vendor patch before acting

Q37. गंभीरता और असर के आधार पर घटनाओं को प्राथमिकता देना क्यों ज़रूरी है?

- (A) यह हमलावरों को विशेषाधिकार बढ़ाने से रोकता है
- (B) यह रोकथाम की आवश्यकता को खत्म करता है
- (C) यह पक्का करता है कि सीमित संसाधनों का सही तरीके से उपयोग हो
- (D) यह खत्म करने में सफलता की गारंटी देता है

Why is prioritizing incidents based on severity and impact essential?

- (A) It prevents attackers from escalating privileges
- (B) It removes the need for containment
- (C) It ensures limited resources are applied effectively
- (D) It guarantees eradication success

Q38. मेटामॉर्फिक मैलवेयर के खिलाफ कौन सी स्टैटिक तकनीक सबसे कम असरदार है?

- (A) हैश-बेस्ड डिटेक्शन
- (B) एंट्रॉपी एनालिसिस
- (C) इंपोर्ट टेबल इंस्पेक्शन
- (D) सेक्शन एंट्रॉपी पैटर्न

Which static technique is least effective against metamorphic malware?

- (A) Hash-based detection
- (B) Entropy analysis
- (C) Import table inspection
- (D) Section entropy patterns

Q39. कौन सा स्टैटिक एनालिसिस तरीका इन्डायरेक्टली पॉलीमॉर्फिक मैलवेयर की पहचान करने में मदद करता है?

- (A) रनटाइम सिस्टम कॉल्स की मॉनिटरिंग
- (B) स्ट्रक्चरल एनालिसिस, एंट्रॉपी कैलकुलेशन, और अनपैकिंग रूटीन
- (C) सिर्फ़ एमडी 5 हैश का उपयोग करना
- (D) एंटीवायरस को डिसेबल करना

Which static analysis approach helps identify polymorphic malware indirectly?

- (A) Monitoring runtime system calls
- (B) Structural analysis, entropy calculation, and unpacking routines
- (C) Using MD5 hashes only
- (D) Disabling antivirus

Q40. सैंडबॉक्स एग्जीक्यूशन के दौरान, मैलवेयर बार-बार अलग-अलग पैरामीटर के साथ *NtQuerySystemInformation* को कॉल करता है, लेकिन कोई दिखने वाला एक्शन नहीं करता है। इसका संभावित उद्देश्य क्या है?

- (A) परसिस्टेंस के लिए प्रोसेस एन्यूमरेशन
- (B) इसके हैश सिग्नेचर को अपडेट करना
- (C) फ़ाइल एन्क्रिप्शन
- (D) एनवायरनमेंट फिंगरप्रिंटिंग और सैंडबॉक्स डिटेक्शन

During sandbox execution, malware repeatedly calls *NtQuerySystemInformation* with varying parameters but performs no visible actions. What is the likely purpose?

- (A) Process enumeration for persistence
- (B) Updating its hash signature
- (C) File encryption
- (D) Environment fingerprinting and sandbox detection

Q41. डायनामिक एनालिसिस से पता चलता है कि मैलवेयर हीप मेमोरी एलोकेट करता है, हाई-एंट्रॉपी डेटा लिखता है, फिर उस पर जंप करने से पहले उसे एग्जीक्यूटेबल के तौर पर मार्क करता है। यह किस बिहेवियर को सबसे ज्यादा दिखाता है?

- (A) एन्क्रिप्टेड पेलोड अनपैकिंग और एग्जीक्यूशन
- (B) रनटाइम डिले लूप के दौरान टाइमिंग-बेस्ड इवेजन
- (C) पैरेंट सिस्टम बाइनरी के अंदर प्रोसेस होलोइंग
- (D) एक बिनाइन प्रोसेस में रिमोट डीएलएल इंजेक्शन

Dynamic analysis shows the malware allocating heap memory, writing high-entropy data, then marking it as executable before jumping to it. What behavior does this most strongly indicate?

- (A) Encrypted payload unpacking and execution
- (B) Timing-based evasion during runtime delay loops
- (C) Process hollowing within a parent system binary
- (D) Remote DLL injection into a benign process

Q42. आपको ऐसा मैलवेयर मिलता है जो हैशिंग के ज़रिए एपीआई कॉल को छिपाता है और उन्हें सिर्फ रनटाइम पर ही ठीक करता है। कौन सा तरीका सबसे कम असरदार है?

- (A) बाइनरी को डीबगर में चलाना और एपीआई रिज़ॉल्यूशन रूटीन को हुक करना
- (B) डिसअसेंबली में एनोटेशन के लिए रनटाइम पर एपीआई एड्रेस कैच करना
- (C) हैश को एपीआई नामों से मैप करने के लिए रिज़ॉल्यूशन फ़ंक्शन को एम्युलेट करना
- (D) हैश फ़ंक्शन को समझे बिना हैश को ब्रूट-फ़ोर्स मैच करना

You encounter malware that obfuscates API calls via hashing and only resolves them at runtime. Which approach is least effective?

- (A) Running the binary in a debugger and hooking API resolution routines
- (B) Capturing API addresses at runtime for annotation in disassembly
- (C) Emulating the resolution function to map hashes to API names
- (D) Brute-force matching hashes without understanding the hash function

- Q43.** एक बाइनरी `CreateRemoteThread` के स्थान पर एपीसी क्यू का उपयोग करके दूसरे प्रोसेस में शेलकोड इंजेक्ट करता है। इस बिहेवियर को कैसे कैप्चर किया जाना चाहिए?
- (A) कर्नेल लेवल पर एपीआई कॉल्स को मॉनिटर करके या वर्चुअलाइजेशन-बेस्ड इंस्ट्रुमेंटेशन के ज़रिए
 - (B) सिर्फ़ पीइ सेक्शन के नाम चेक करके
 - (C) एग्जीक्यूटेबल्स के हैश की तुलना करके
 - (D) .rsrc में हाई-एंट्रॉपी डेटा खोजना

A binary injects shellcode into another process using APC queues rather than `CreateRemoteThread`. How should this behavior be captured?

- (A) By monitoring API calls at the kernel level or via virtualization-based instrumentation
 - (B) Only checking PE section names
 - (C) Comparing hashes of executables
 - (D) Searching for high-entropy data in .rsrc
- Q44.** मैलवेयर के खिलाफ़ कौन सी तकनीक सबसे असरदार है जो अपने इंपोर्ट एड्रेस टेबल को डायनैमिकली बदलता है?
- (A) सेक्शन एंट्रॉपी स्कैनिंग
 - (B) सिर्फ़ स्टैटिक इंपोर्ट टेबल इंस्पेक्शन
 - (C) रनटाइम पर रिज़ॉल्व किए गए एपीआई एड्रेस को देखना और डिसअसेंबली पर टिप्पणी करना
 - (D) पीइ सेक्शन का नाम अपनी मर्ज़ी से बदलना

Which technique is most effective against malware that modifies its Import Address Table dynamically?

- (A) Section entropy scanning
 - (B) Static import table inspection only
 - (C) Observing resolved API addresses at runtime and annotating disassembly
 - (D) Renaming PE sections arbitrarily
- Q45.** टीओसीटीओयू (टाइम-ऑफ़-चेक टू टाइम-ऑफ़-यूज़) वलनरेबिलिटी एक खास तरह की _____ है।
- (A) पॉइंटर अरिथमेटिक एरर की वजह से मेमोरी करप्शन फ़्लॉ
 - (B) प्रिविलेज वैलिडेशन और रिसोर्स एक्सेस के बीच रेस कंडीशन
 - (C) नॉन्स को दोबारा इस्तेमाल करने से क्रिप्टोग्राफ़िक कमजोरी
 - (D) सेशन टाइमआउट हैंडलिंग में लॉजिक एरर

A TOCTOU (Time-of-Check to Time-of-Use) vulnerability is a specific type of:

- (A) Memory corruption flaw due to pointer arithmetic errors
- (B) Race condition between privilege validation and resource access
- (C) Cryptographic weakness from reusing nonces
- (D) Logic error in session timeout handling

- Q46.** टीओसीटीओयू जैसी रेस कंडीशन वल्नरेबिलिटीज़ के एक्सप्लॉइटेशन के खिलाफ़ कौन सी मिटिगेशन टेक्नीक सबसे कम असरदार है?
- (A) फ़ाइलनेम-बेस्ड ऑपरेशन्स के बजाय फ़ाइल डिस्क्रिप्टर पास करना
 - (B) एटॉमिक फ़ाइलसिस्टम ऑपरेशन्स (जैसे, O_EXCL फ़्लैग) का उपयोग करना
 - (C) स्टैक कैनरीज़ को इनेबल करना
 - (D) पाथ-बेस्ड चेक्स के बजाय कैपेबिलिटी-बेस्ड एक्सेस कंट्रोल का उपयोग करना

Which mitigation technique is LEAST effective against exploitation of race condition vulnerabilities like TOCTOU?

- (A) File descriptor passing instead of filename-based operations
- (B) Using atomic filesystem operations (e.g., O_EXCL flag)
- (C) Enabling stack canaries
- (D) Employing capability-based access control instead of path-based checks

- Q47.** इनमें से कौन सा सीडब्ल्यूइ और सीवीइ के बीच के संबंध को सही-सही बताता है?
- (A) हर सीवीइ को ठीक एक सीडब्ल्यूइ से मैप करना चाहिए, और इसका उल्टा भी
 - (B) सीडब्ल्यूइ वल्नरेबिलिटी टाइप बताते हैं; सीवीइ प्रोडक्ट्स में उन टाइप के खास इंस्टेंस बताते हैं
 - (C) वल्नरबिलिटी ट्रैकिंग के लिए सीडब्ल्यूइ के पक्ष में सीवीइ को हटा दिया गया है
 - (D) सीडब्ल्यूइ नंबर एमआईटीआरइ द्वारा सीवीइ पब्लिकेशन के बाद ही दिए जाते हैं

Which of the following accurately describes the relationship between CWE and CVE?

- (A) Every CVE must map to exactly one CWE, and vice versa
- (B) CWEs describe vulnerability types; CVEs describe specific instances of those types in products
- (C) CVEs are deprecated in favor of CWEs for vulnerability tracking
- (D) CWE numbers are assigned by MITRE only after CVE publication

- Q48.** इनमें से कौन सा सिनेरियो हीप ओवरफ़्लो (स्टैक ओवरफ़्लो नहीं) का क्लासिक उदाहरण है?
- (A) ओवरसाइज़्ड scanf() इनपुट के ज़रिए लोकल फ़ंक्शन वेरिएबल्स को ओवरराइट करना
 - (B) फिक्स्ड-साइज़ चार ऐरे में gets() के ज़रिए सेव किए गए इआइपी को स्मैश करना
 - (C) execve() को पास किए गए एनवायरनमेंट वेरिएबल्स में शेलकोड इंजेक्ट करना
 - (D) डायनामिक रूप से एलोकेट किए गए बफ़र की बाउंड्स से आगे लिखकर मैलोक मेटाडेटा को करप्ट करना

Which of the following scenarios is a classic example of a heap overflow (not stack overflow)?

- (A) Overwriting local function variables via oversized scanf() input
- (B) Smashing saved EIP via gets() in a fixed-size char array
- (C) Injecting shellcode into environment variables passed to execve()
- (D) Corrupting malloc metadata by writing past bounds of dynamically allocated buffer

- Q49.** एक अटैकर का उद्देश्य ऐसी सर्विस पर यूज़-आफ्टर-फ्री का लाभ उठाना है जो फ्री मेमोरी को तेज़ी से दोबारा उपयोग करती है। कौन सी तकनीक सबसे ज़्यादा लाभ उठाने में मदद करती है?
- (A) एलोकेशन एडजेंसी और रीयूज़ पैटर्न पर असर डालने के लिए हीप ग्रूमिंग
 - (B) सर्वर पर लॉग वर्बोसिटी बढ़ाना
 - (C) सर्वर फाइलसिस्टम पर फाइल परमिशन बदलना
 - (D) फायदा उठाने से पहले बाइनरी का नाम बदलना

An attacker aims to exploit a use-after-free on a service that reuses freed memory quickly. Which technique most aids exploitation?

- (A) Heap grooming to influence allocation adjacency and reuse patterns
- (B) Increasing log verbosity on the server
- (C) Changing file permissions on the server filesystem
- (D) Renaming the binary before exploitation

- Q50.** एक प्रोग्राम 32-बिट लेंथ फ़ील्ड का उपयोग करता है लेकिन 64-बिट आर्किटेक्चर पर चलता है। किस कोडिंग एरर से सबसे अधिक एक्सप्लॉइटैबल इंटीजर ओवरफ़्लो होने की संभावना है?
- (A) बहुत ज़्यादा हेडर फ़ाइलें शामिल करना
 - (B) बिना फ़ॉर्मेटिंग स्पेसिफ़ायर के printf का इस्तेमाल करना
 - (C) वर्चुअल मशीन पर बाइनरी चलाना
 - (D) बिना वैलिडेशन के साइन किए हुए 32-बिट इनपुट को अनसाइन्ड 64-बिट साइज़ अरिथमेटिक में मिलाना

A program uses 32-bit length fields but runs on a 64-bit architecture. Which coding error most likely leads to an exploitable integer overflow?

- (A) Including too many header files
- (B) Using printf without formatting specifiers
- (C) Running the binary on a virtual machine
- (D) Mixing signed 32-bit inputs into unsigned 64-bit size arithmetic without validation

- Q51.** एक सर्विस एक API दिखाती है जो अनट्रस्टेड डेटा को डीसीरियलाइज़ करता है। अगर डीसीरियलाइज़ेशन अनसेफ है, तो किस तरह की वल्नरेबिलिटी सबसे ज़्यादा होने की संभावना है?
- (A) स्मार्ट कॉन्ट्रैक्ट में रीएंट्रेंसी
 - (B) बफ़र साइज़ बढ़ने से RCE अपने आप रुक जाता है
 - (C) मैलिशियसली क्राफ़्टेड सीरियलाइज़ेड पेलोड के ज़रिए रिमोट कोड एग्ज़िक्यूशन
 - (D) सीरियलाइज़ेशन हमेशा ग्रेसफुली फेल हो जाता है

A service exposes an API that deserializes untrusted data. Which class of vulnerability is most likely if deserialization is unsafe?

- (A) Reentrancy in smart contracts
- (B) Buffer size increases automatically prevent RCE
- (C) Remote code execution via maliciously crafted serialized payloads
- (D) Serialization always fails gracefully

Q52. पैसिव रेकनिसेंस के दौरान, कौन सी तकनीक टारगेट ऑर्गनाइज़ेशन के डब्ल्यूएफ या आइडीएस को ट्रिगर किए बिना सबडोमेन की गिनती करने देती है?

- (A) वर्डलिस्ट + वाइल्डकार्ड डीएनएस डिटेक्शन का उपयोग करके ffuf के साथ ब्रूट-फोर्सिंग
- (B) crt.sh के ज़रिए सर्टिफिकेट ट्रांसपेरेंसी (सीटी) लॉग की क्लेरी करना
- (C) ऑथेंटिक डीएनएस सर्वर के खिलाफ़ ज़ोन ट्रांसफर करना
- (D) गेस्ड vhost नामों पर एचटीटीपी हेड रिक्वेस्ट भेजना

During passive reconnaissance, which technique allows enumeration of subdomains without triggering target organization's WAF or IDS?

- (A) Brute-forcing with ffuf using wordlist + wildcard DNS detection
- (B) Querying Certificate Transparency (CT) logs via crt.sh
- (C) Performing zone transfers against authoritative DNS servers
- (D) Sending HTTP HEAD requests to guessed vhost names

Q53. Nmap में, बचने और भरोसेमंद होने के मामले में -sS (SYN स्कैन) और -sT (TCP कनेक्ट स्कैन) के बीच क्या फंक्शनल अंतर है?

- (A) -sS धीमा है लेकिन स्टेटफुल फायरवॉल को बायपास करता है; -sT तेज़ है लेकिन एप्लिकेशन द्वारा लॉग किया जाता है
- (B) -sS UDP सर्विस का पता नहीं लगा सकता; -sT TCP और UDP दोनों पोर्ट को स्कैन कर सकता है
- (C) -sT RST-बेस्ड फायरवॉल नियमों से बचता है; -sS WAF में SYN-ACK लॉगिंग को ट्रिगर करता है
- (D) -sS को रूट प्रिविलेज की ज़रूरत होती है और यह रॉ पैकेट भेजता है; -sT OS सॉकेट API का इस्तेमाल करता है और बिना प्रिविलेज के काम करता है

In Nmap, what is the functional difference between -sS (SYN scan) and -sT (TCP connect scan) in terms of evasion and reliability?

- (A) -sS is slower but bypasses stateful firewalls; -sT is faster but logged by applications
- (B) -sS cannot detect UDP services; -sT can scan both TCP and UDP ports
- (C) -sT evades RST-based firewall rules; -sS triggers SYN-ACK logging in WAFs
- (D) -sS requires root privileges and sends raw packets; -sT uses OS socket API and works unprivileged

Q54. रिस्ट्रिक्टेड एक्सप्लॉइट सिनेरियो में स्टेजलेस पेलोड के बजाय स्टेज्ड पेलोड (जैसे, windows/meterpreter/reverse_tcp) उपयोग करने का मुख्य लाभ क्या है?

- (A) शुरुआती स्टेजर छोटा होता है, जो साइज़ लिमिट और कुछ एवी ह्यूरिस्टिक को बायपास करने में मदद करता है
- (B) वे हमेशा डिफ़ॉल्ट रूप से दूसरे स्टेज के डाउनलोड को एन्क्रिप्ट करते हैं
- (C) उन्हें बिना किसी नेटवर्क कनेक्टिविटी के उपयोग किया जा सकता है
- (D) वे एग्ज़िक्यूशन के दौरान अपने आप प्रिविलेज बढ़ा देते हैं

What is the primary advantage of using staged payloads (e.g., windows/meterpreter/reverse_tcp) instead of stageless payloads in restricted exploit scenarios?

- (A) The initial stager is small, helping bypass size limits and some AV heuristics
- (B) They always encrypt the second stage download by default
- (C) They can be used without any network connectivity
- (D) They automatically escalate privileges during execution

Q55. **वल्नरेबिलिटी रिपोर्टिंग एथिक्स में “रिस्पॉन्सिबल डिस्क्लोजर” और “फुल डिस्क्लोजर” में क्या अंतर है?**

- (A) रिस्पॉन्सिबल डिस्क्लोजर के लिए रिसर्चर को पेमेंट करना ज़रूरी है; फुल डिस्क्लोजर के लिए नहीं
- (B) रिस्पॉन्सिबल डिस्क्लोजर कानूनी तौर पर ज़रूरी है; फुल डिस्क्लोजर के लिए नहीं
- (C) फुल डिस्क्लोजर तुरंत पब्लिक रिलीज़ ज़रूरी बनाता है; रिस्पॉन्सिबल वेंडर को सुधार का समय देता है
- (D) फुल डिस्क्लोजर के लिए सरकार से पहले से मंजूरी लेनी ज़रूरी है; रिस्पॉन्सिबल के लिए नहीं

What distinguishes “responsible disclosure” from “full disclosure” in vulnerability reporting ethics?

- (A) Responsible disclosure requires payment to the researcher; full disclosure does not
- (B) Responsible disclosure is legally binding; full disclosure is not
- (C) Full disclosure mandates immediate public release; responsible allows vendor remediation time
- (D) Full disclosure requires government pre-approval; responsible does not

Q56. **किसी ऑर्गनाइज़ेशन की वेबसाइट पर पोस्ट की गई “वल्नरेबिलिटी डिस्क्लोजर पॉलिसी (वीडीपी)” का मुख्य काम क्या है?**

- (A) रिसर्चर्स को अट्रैक्ट करने के लिए बग बाउंटि रिवॉर्ड्स का विज्ञापन करना
- (B) भविष्य की सभी सिक््योरिटी कमियों के लिए कानूनी तौर पर ज़िम्मेदारी माफ़ करना
- (C) पीसीआइ डीएसएस ज़रूरत 6.2 का पालन करना
- (D) बाहरी रिपोर्टर्स के लिए ऑथराइज़्ड चैनल और सेफ़ हार्बर बनाना

What is the primary function of a “Vulnerability Disclosure Policy (VDP)” posted on an organization’s website?

- (A) To advertise bug bounty rewards to attract researchers
- (B) To legally waive liability for all future security flaws
- (C) To comply with PCI DSS requirement 6.2
- (D) To establish authorized channels and safe harbor for external reporters

Q57. **वल्नरेबिलिटी रिस्पॉन्स में “रिमेडिएशन” और “मिटिगेशन” में क्या अंतर है?**

- (A) रिमेडिएशन कमी को खत्म करता है; मिटिगेशन असली वजह को ठीक किए बिना असर को कम करता है
- (B) मिटिगेशन के लिए कोड में बदलाव की ज़रूरत होती है; रिमेडिएशन फ़ायरवॉल नियमों का उपयोग करता है
- (C) रिमेडिएशन टेम्पररी होता है; मिटिगेशन परमानेंट होता है
- (D) मिटिगेशन वेंडर करते हैं; रिमेडिएशन कस्टमर करते हैं

What distinguishes “remediation” from “mitigation” in vulnerability response?

- (A) Remediation eliminates the flaw; mitigation reduces impact without fixing root cause
- (B) Mitigation requires code changes; remediation uses firewall rules
- (C) Remediation is temporary; mitigation is permanent
- (D) Mitigation is performed by vendors; remediation by customers

Q58. कौन सा एरर हैंडलिंग तरीका प्रोडक्शन एपीआई में जानकारी लीक होने का खतरा सबसे ज्यादा बढ़ाता है?

- (A) रिस्पॉन्स बॉडी में स्टैक ट्रेस के साथ एचटीटीपी 500 रिटर्न करना
- (B) क्लाइंट को दिखाए बिना सर्वर-साइड पर पूरी एक्सेप्शन डिटेल्स लॉग करना
- (C) खास फेलियर के लिए इंटरनली मैप किए गए "E1001" जैसे जेनेरिक एरर कोड का उपयोग करना
- (D) सिर्फ खराब इनपुट के लिए स्कीमा वैलिडेशन एरर के साथ एचटीटीपी 400 रिटर्न करना

Which error handling approach MOST increases risk of information leakage in production APIs?

- (A) Returning HTTP 500 with stack trace in response body
- (B) Logging full exception details server-side without exposing them to client
- (C) Using generic error codes like "E1001" mapped internally to specific failures
- (D) Returning HTTP 400 with schema validation errors only for malformed input

Q59. एक डेवलपमेंट टीम `/\<script.*?\>/gi` जैसी रेगैक्स ब्लैकलिस्ट का उपयोग करके संभावित XSS को ब्लॉक करती है, लेकिन व्हाइटलिस्ट लागू नहीं करती है। इनमें से कौन सा तरीका इस तरीके की मुख्य कमजोरी को सबसे अच्छे से बताता है?

- (A) ब्लैकलिस्ट एन्कोडेड या फ्रैगमेंटेड पेलोड को मिस कर सकती हैं, जिससे बाईपास हो सकता है
- (B) ब्लैकलिस्ट हाई-वॉल्यूम सिस्टम में इनपुट हैंडलिंग को धीमा कर सकती हैं, जिससे डीओएस आसान हो जाता है
- (C) ब्लैकलिस्ट अनजाने में एचटीएमएल कमेंट्स और स्टाइल टैग जैसे सही मार्कअप को ब्लॉक कर सकती हैं
- (D) ब्लैकलिस्ट ब्राउज़र पार्सिंग की कमियों पर निर्भर करती हैं और सिर्फ जाने-पहचाने खराब इनपुट को एन्क्रिप्ट कर सकती हैं

A development team blocks potential XSS by using a regex blacklist such as `/\<script.*?\>/gi` but does not implement a whitelist. Which of the following best describes the core weakness of this approach?

- (A) Blacklists can miss encoded or fragmented payloads, allowing bypasses
- (B) Blacklists can slow down input handling in high-volume systems, making DoS easier
- (C) Blacklists may unintentionally block legitimate markup like HTML comments and style tags
- (D) Blacklists depend on browser parsing quirks and may encrypt only known bad inputs

- Q60.** सिक्वोर कोडिंग में, कौन सी प्रैक्टिस सबसे सीधे सेकंड-ऑर्डर इंजेक्शन अटैक (पेलोड पहले स्टोर करना, बाद में एग्जीक्यूट करना) को रोकती है?
- (A) सिर्फ स्टोर किए गए डेटा को दिखाते या उपयोग करते समय सैनिटाइज़ करना
 - (B) सिर्फ एंटी के समय डेटा की लंबाई को वैलिडेट करना
 - (C) प्रोडक्शन में वर्बोज़ एरर मैसेज चालू करना
 - (D) स्टोरेज से पहले यूज़र इनपुट के एमडी5 हैश का उपयोग करना

In secure coding, which practice most directly prevents second-order injection attacks (payload stored first, executed later)?

- (A) Sanitizing only when displaying or using stored data
- (B) Validating data length at entry only
- (C) Enabling verbose error messages in production
- (D) Using MD5 hashes of user input before storage

- Q61.** सेंसिटिव क्रेडेंशियल्स को हैंडल करने वाले प्रोजेक्ट में, कौन सा मेमोरी हैंडलिंग तरीका सिक्वोर कोडिंग प्रिंसिपल्स के साथ सबसे ज़्यादा अलाइन है?
- (A) प्लेन एरे में क्रेडेंशियल्स स्टोर करना और प्रोसेस से बाहर निकलने पर ओएस को मेमोरी रीक्लेम करने देना
 - (B) डेडिकेटेड सिक्वोर मेमोरी एपीआई का उपयोग करना जो उपयोग के बाद ज़ीरो बफ़र्स करते हैं और स्वैपिंग को रोकते हैं
 - (C) क्रेडेंशियल्स को एन्क्रिप्ट करना लेकिन डिफ़िप्शन कीज़ को ग्लोबल वेरिएबल्स में रखना
 - (D) रिडंडेंसी पक्का करने के लिए क्रेडेंशियल्स को कई बफ़र्स में कॉपी करना

In a project handling sensitive credentials, which memory handling approach is most aligned with secure coding principles?

- (A) Storing credentials in plain arrays and letting the OS reclaim memory on process exit
- (B) Using dedicated secure memory APIs that zero buffers after use and resist swapping
- (C) Encrypting credentials but keeping decryption keys in global variables
- (D) Copying credentials into multiple buffers to ensure redundancy

- Q62.** यूज़-आफ्टर-फ्री वल्नरेबिलिटी को रोकने के लिए कौन सी मेमोरी प्रैक्टिस कोड रिव्यू को सबसे अच्छे से पूरा करती है?
- (A) एलोकेशन को कम करने के लिए हमेशा फ्री किए गए पॉइंटर्स का दोबारा उपयोग करें
 - (B) सभी हीप यूसेज को स्टैटिक बफ़र्स से बदलें
 - (C) फ्री करने के तुरंत बाद पॉइंटर्स को नलिफाई करें और डीलोकेशन को सेंट्रलाइज़ करें
 - (D) डीबगिंग के लिए फ्री किए गए पॉइंटर्स को लॉग में कॉपी करें

Which memory practice best complements code reviews to prevent use-after-free vulnerabilities?

- (A) Always reuse freed pointers to minimize allocations
- (B) Replace all heap usage with static buffers
- (C) Immediately nullify pointers after freeing and centralize deallocation
- (D) Copy freed pointers into logs for debugging

- Q63.** एक टीम का दावा है कि उसने अपने एसडीएल(सी) में “सिक््योरिटी को बाईं ओर शिफ्ट” कर दिया है। कौन सी प्रैक्टिस इस दावे को सबसे सीधे तौर पर साबित करती है?
- (A) प्रोडक्शन रिलीज़ के समय एक ऑटोमेटेड डीएएसटी स्कैन जोड़ना
 - (B) डेवलपर्स से एक सिक््योर कोडिंग पॉलिसी पर साइन करवाना
 - (C) पहले बड़े वर्शन के बाद पेन टेस्ट की ज़रूरत
 - (D) कोड कमिट के दौरान एसएएसटी चलाना और मर्ज से पहले नतीजों को देखना

A team claims it has “shifted security left” in its SDLC. Which practice most directly proves that claim?

- (A) Adding an automated DAST scan at production release time
- (B) Having developers sign a secure coding policy
- (C) Requiring pen tests after the first major version
- (D) Running SAST during code commits and addressing findings before merge

- Q64.** सिक््योर एसडीएलसी कॉन्टेक्ट में, इनमें से कौन सा एसएएसटी को डीएएसटी से सबसे अच्छे तरीके से अलग करता है?

- (A) एसएएसटी जाने-पहचाने इनपुट के साथ कोड चलाकर कमज़ोरियों का पता लगाता है, जबकि डीएएसटी कोड को स्टैटिकली इंस्पेक्ट करता है
- (B) एसएएसटी कोड को बिना चलाए एनालाइज़ करता है, जबकि डीएएसटी चल रहे एप्लिकेशन को बाहर से टेस्ट करता है
- (C) एसएएसटी सिर्फ़ रनटाइम कॉन्फ़िगरेशन की कमियाँ ढूँढ सकता है, जबकि डीएएसटी नहीं
- (D) एसएएसटी अपने आप कमज़ोरियों को ठीक करता है, जबकि डीएएसटी नहीं

In the Secure SDLC context, which of the following best differentiates SAST from DAST?

- (A) SAST finds vulnerabilities by executing the code with known inputs, while DAST inspects code statically
- (B) SAST analyzes code without running it, while DAST tests the running application from the outside
- (C) SAST can only find runtime configuration flaws, while DAST can't
- (D) SAST automatically remediates vulnerabilities, while DAST doesn't

- Q65.** काफी बड़े क्वांटम कंप्यूटर पर शोर के एल्गोरिदम से किस क्रिप्टोग्राफ़िक प्रिमिटिव को सबसे ज़्यादा खतरा है?

- (A) सिमेट्रिक एन्क्रिप्शन (एडएस-256)
- (B) हैश फ़ंक्शन (एसएचए-3)
- (C) इंटीजर फ़ैक्टराइज़ेशन और डिस्क्रीट लॉगरिदम प्रॉब्लम (आरएसए, इसीसी)
- (D) की डेरिवेशन फ़ंक्शन (एचकेडीएफ)

Which cryptographic primitive is MOST threatened by Shor's algorithm on a sufficiently large quantum computer?

- (A) Symmetric encryption (AES-256)
- (B) Hash functions (SHA-3)
- (C) Integer factorization and discrete logarithm problems (RSA, ECC)
- (D) Key derivation functions (HKDF)

- Q66.** कौन सी एनआइएसटी-सिलेक्टेड पीक्यूसी एल्गोरिदम फ़ैमिली हाई-डाइमेंशनल लैटिस में शॉर्ट वेक्टरस को खोजने की हार्डनेस पर निर्भर करती है?
- (A) कोड-बेस्ड (क्लासिक मैकएलीस)
 - (B) हैश-बेस्ड (एसपीएचआइएनसीएस+)
 - (C) लैटिस-बेस्ड (क्रिस्टल्स-काइबर, डिलिथियम)
 - (D) मल्टीवेरिएट-क्वाड्रेटिक (रेनबो)

Which NIST-selected PQC algorithm family relies on the hardness of finding short vectors in high-dimensional lattices?

- (A) Code-based (Classic McEliece)
- (B) Hash-based (SPHINCS+)
- (C) Lattice-based (CRYSTALS-Kyber, Dilithium)
- (D) Multivariate-quadratic (Rainbow)

- Q67.** मॉडर्न एप्लिकेशन्स में इसीसी के लिए इनमें से कौन सा आम इस्तेमाल है?

- (A) सिमेट्रिक साइफर के बिना बड़ी मात्रा में डेटा एन्क्रिप्ट करना
- (B) पासवर्ड स्टोरेज में टाइमिंग अटैक को रोकना
- (C) ब्लॉकचेन में हैश फ़ंक्शन को पूरी तरह से बदलना
- (D) डिजिटल सिग्नेचर, सिक्चोर की एक्सचेंज, और टीएलएस हैंडशेक

Which of the following is a common use case for ECC in modern applications?

- (A) Encrypting large volumes of data without symmetric ciphers
- (B) Preventing timing attacks in password storage
- (C) Replacing hash functions in blockchain entirely
- (D) Digital signatures, secure key exchange, and TLS handshakes

- Q68.** की एक्सचेंज के लिए इसीसी का उपयोग करते समय, कौन सी प्रॉपर्टी फॉरवर्ड सीक्रेसी पक्का करती है?

- (A) सेशन की के बिना स्टैटिक पब्लिक/प्राइवेट की पेयर
- (B) हर सेशन के लिए इफेमेरल की पेयर बनाना (इफेमेरल इसीडीएच)
- (C) डिप्लॉयमेंट के समय एक बार बांटी गई फिक्स्ड शेयर्ड की का उपयोग करना
- (D) इसीसी को एमडी5 हैशिंग के साथ मिलाना

When using ECC for key exchange, which property ensures forward secrecy?

- (A) Static public/private key pairs without session keys
- (B) Generating ephemeral key pairs for each session (ephemeral ECDH)
- (C) Using fixed shared keys distributed once at deployment
- (D) Combining ECC with MD5 hashing

- Q69. इनमें से कौन सा ब्लॉकचेन में कोलिजन-रेसिस्टेंट हैश फ़ंक्शन को सही ढंग से बताता है?
- (A) यह पक्का करता है कि कोई भी दो अलग-अलग इनपुट एक जैसा हैश आउटपुट न दें
 - (B) यह गारंटी देता है कि सभी ब्लॉक एक ही समय पर माइन किए जाएं
 - (C) यह एक ही ट्रांज़ैक्शन को कई ब्लॉक में दिखने देता है
 - (D) यह एक सिमेट्रिक की का उपयोग करके ब्लॉकचेन डेटा को एन्क्रिप्ट करता है

Which of the following correctly describes a collision-resistant hash function in blockchain?

- (A) It ensures no two distinct inputs produce the same hash output
- (B) It guarantees that all blocks are mined at the same time
- (C) It allows the same transaction to appear in multiple blocks
- (D) It encrypts the blockchain data using a symmetric key

- Q70. ब्लॉकचेन नेटवर्क में लाइट क्लाइंट के लिए मर्कल ट्री का फ़ायदा कौन सा सिनेरियो दिखाता है?
- (A) मर्कल ट्री अपने आप ट्रांज़ैक्शन कंटेंट को एन्क्रिप्ट करते हैं।
 - (B) नोड्स मर्कल ट्री का उपयोग करके तेज़ी से ब्लॉक माइन कर सकते हैं।
 - (C) माइनर्स मर्कल रूट का उपयोग करके भविष्य के ब्लॉक का अनुमान लगा सकते हैं।
 - (D) एक मोबाइल वॉलेट पूरा ब्लॉकचेन डाउनलोड किए बिना ट्रांज़ैक्शन को वेरिफ़ाई कर सकता है।

Which scenario demonstrates the advantage of Merkle trees for light clients in a blockchain network?

- (A) Merkle trees automatically encrypt transaction contents.
- (B) Nodes can mine blocks faster using Merkle trees.
- (C) Miners can predict future blocks using the Merkle root.
- (D) A mobile wallet can verify a transaction without downloading the full blockchain

- Q71. प्रैक्टिकल बाइजेंटाइन फॉल्ट टॉलरेंस (पीबीएफटी) का उपयोग करने वाले एक परमिशन वाले ब्लॉकचेन में, लाइवनेस और सेफ्टी बनाए रखते हुए 7 मैलिशियस नोड्स को टॉलरेट करने के लिए कम से कम कुल कितने नोड्स की ज़रूरत होती है?

In a permissioned blockchain using Practical Byzantine Fault Tolerance (PBFT), what is the *minimum* number of total nodes required to tolerate 7 malicious nodes while maintaining liveness and safety?

- (A) 14
- (B) 15
- (C) 21
- (D) 22

- Q72.** इनमें से कौन सा प्रूफ-ऑफ-स्टेक सिस्टम में “लॉन्ग-रेंज अटैक” को सबसे अच्छे से बताता है?
- (A) एक अटैकर रिसेंट हिस्ट्री को फिर से रीराइट करने के लिए हैशिंग पावर रेंट पर लेता है
 - (B) एक अटैकर बहुत पहले की पुरानी प्राइवेट कुंजियों का उपयोग करके जेनेसिस से एक अल्टरनेटिव चेन बनाता है
 - (C) एक वैलिडिटर समय में बहुत आगे के ब्लॉक प्रपोज़ करता है
 - (D) एक नोड 1 साल से पुराने स्नैपशॉट से सिंक करता है

Which of the following best describes “long-range attack” in Proof-of-Stake systems?

- (A) An attacker rents hashing power to rewrite recent history
- (B) An attacker uses old private keys from long ago to build an alternative chain from genesis
- (C) A validator proposes blocks too far ahead in time
- (D) A node syncs from a snapshot older than 1 year

- Q73.** प्रूफ-ऑफ-वर्क (पीओडब्ल्यू) की तुलना में प्रूफ-ऑफ-स्टेक (पीओएस) की मुख्य कमी क्या है?

- (A) यह स्वाभाविक रूप से बहुत ज़्यादा बिजली की खपत करता है
- (B) यह ब्लॉकचेन को बदलना नामुमकिन बना देता है
- (C) यह ट्रांज़ैक्शन को वैलिडेट नहीं कर सकता
- (D) अगर कुछ पार्टिसिपेंट्स के पास ज़्यादातर स्टेक्स हों तो इससे सेंट्रलाइज़ेशन हो सकता है

Which is a key drawback of Proof-of-Stake (PoS) compared to Proof-of-Work (PoW)?

- (A) It inherently consumes excessive electricity
- (B) It makes blockchain immutability impossible
- (C) It cannot validate transactions
- (D) It may lead to centralization if a few participants hold the majority of stakes

- Q74.** कौन सी वल्नरेबिलिटी तब आती है जब कॉन्ट्रैक्ट का लॉजिक यह मान लेता है कि मौजूदा एग्ज़िक्यूशन पूरा होने से पहले एक्सटर्नल कॉल्स री-एंटर नहीं हो सकतीं, जिससे स्टेट इनकंसिस्टेंसी होती है?

- (A) इंटीजर अंडरफ्लो
- (B) रीएंट्रेंसी
- (C) एक्सेस कंट्रोल बाईपास
- (D) फ्रंट-रनिंग

Which vulnerability arises when a contract’s logic assumes that external calls cannot re-enter before the current execution completes, leading to state inconsistencies?

- (A) Integer underflow
- (B) Reentrancy
- (C) Access control bypass
- (D) Front-running

- Q75. सॉलिडिटी में "सेल्फडिस्ट्रक्ट(एड्रेस)" फ़ंक्शन का गलत उपयोग होने पर किस तरह का सिक्योरिटी रिस्क आता है?
- (A) गैस ऑप्टिमाइज़ेशन फेलियर
 - (B) टोकन की बिना इजाज़त मिंटिंग
 - (C) इम्म्यूटेबिलिटी का नुकसान और संभावित फंड लॉकिंग
 - (D) मेटाडेटा लीकेज

The "selfdestruct(address)" function in Solidity introduces which class of security risk if misused?

- (A) Gas optimization failure
- (B) Unauthorized minting of tokens
- (C) Loss of immutability and potential fund locking
- (D) Metadata leakage

- Q76. जब कोई कॉन्ट्रैक्ट किसी अनट्रस्टेड या अपग्रेडेबल लाइब्रेरी से लॉजिक को एग्जीक्यूट करने के लिए *delegatecall* का उपयोग करता है, तो कौन सी वलनरेबिलिटी आती है?
- (A) रीएंटरेंसी
 - (B) एबीआई एन्कोडिंग मिसमैच
 - (C) गैस ग्रिफिंग
 - (D) स्टोरेज कोलिजन / अनइंटेंडेड स्टेट म्यूटेशन

What vulnerability is introduced when a contract uses *delegatecall* to execute logic from an untrusted or upgradable library?

- (A) Reentrancy
- (B) ABI encoding mismatch
- (C) Gas griefing
- (D) Storage collision / unintended state mutation

- Q77. डब्ल्यू3सी स्टैंडर्ड के हिसाब से डीसेंट्रलाइज़्ड आइडेंटिटी (DID) सिस्टम में, "डीआईडी डॉक्यूमेंट" क्या रोल निभाता है?
- (A) यूज़र ऑथेंटिकेशन के लिए बायोमेट्रिक डेटा स्टोर करता है
 - (B) डीआईडी को क्रिप्टोग्राफ़िक की, सर्विस एंडपॉइंट और वेरिफ़िकेशन मेथड से मैप करता है
 - (C) सभी जारी किए गए क्रेडेंशियल की ग्लोबल रजिस्ट्री के तौर पर काम करता है
 - (D) जीडीपीआर-कम्प्लायंट डेटा डिलीट करने की रिक्वेस्ट को लागू करता है

In decentralized identity (DID) systems compliant with W3C standards, what role does the "DID Document" serve?

- (A) Stores biometric data for user authentication
- (B) Maps a DID to cryptographic keys, service endpoints, and verification methods
- (C) Acts as a global registry of all issued credentials
- (D) Enforces GDPR-compliant data deletion requests

- Q78.** “सिबिल-रेसिस्टेंट डिसेंट्रलाइज़्ड आइडेंटिफ़ायर” खास तौर पर एंटरप्राइज़ एसएसओ सिस्टम में किस खतरे को रोकने का लक्ष्य रखते हैं?
- (A) क्रेडेंशियल स्टफ़िंग
 - (B) लॉगिन पोर्टल में क्रॉस-साइट स्क्रिप्टिंग
 - (C) टीएलएस हैंडशेक के दौरान मैन-इन-द-मिडल
 - (D) ऑथेंटिकेशन सर्वर पर कब्ज़ा करने के लिए असीमित नकली आइडेंटिटी बनाना

Which threat does “Sybil-resistant decentralized identifiers” specifically aim to prevent in enterprise SSO systems?

- (A) Credential stuffing
- (B) Cross-site scripting in login portals
- (C) Man-in-the-middle during TLS handshake
- (D) Creation of unlimited fake identities to overwhelm authentication servers

- Q79.** रॉ सेंसिटिव सप्लाय चैन डेटा को स्टोर करने के लिए पब्लिक ब्लॉकचेन (जैसे इथेरियम) का उपयोग करने की बुनियादी लिमिटेशन क्या है?
- (A) ब्लॉक साइज़ लिमिट बड़े डेटासेट को रोकती है
 - (B) स्मार्ट कॉन्ट्रैक्ट का एग्ज़िक्यूशन बहुत धीमा है
 - (C) पब्लिक में दिखने वाला डेटा कॉन्फिडेंशियलिटी और रेगुलेटरी कम्प्लायंस का उल्लंघन करता है
 - (D) आइओटी डिवाइस सिग्नेचर के लिए सपोर्ट की कमी

What is the fundamental limitation of using public blockchains (like Ethereum) for storing raw sensitive supply chain data?

- (A) Block size limits prevent large datasets
- (B) Smart contract execution is too slow
- (C) Publicly visible data violates confidentiality and regulatory compliance
- (D) Lack of support for IoT device signatures

- Q80.** रेगुलेटेड इंडस्ट्रीज़ के लिए ब्लॉकचेन-बेस्ड ऑडिट लॉग में “कैमेलियन हैश” उपयोग करने का मुख्य लाभ क्या है?
- (A) यह रेगुलेटर्स को चैन इंटीग्रिटी को तोड़े बिना सेंसिटिव एंट्रीज़ को हटाने की सुविधा देता है
 - (B) ब्लॉक प्रोपेगेशन को तेज़ करता है
 - (C) क्वांटम-रेज़िस्टेंट सिग्नेचर को इनेबल करता है
 - (D) स्टोरेज को 90% तक कंप्रेस करता है

What is the main advantage of using a “chameleon hash” in blockchain-based audit logs for regulated industries?

- (A) Allows regulators to redact sensitive entries without breaking chain integrity
- (B) Speeds up block propagation
- (C) Enables quantum-resistant signatures
- (D) Compresses storage by 90%

- Q81.** सुपरवाइज्ड एमआइ-बेस्ड मैलवेयर डिटेक्शन में, “कॉन्सेप्ट ड्रिफ्ट” एक बड़ी चुनौती क्यों है?
- (A) मैलवेयर ऑथर एन्क्रिप्शन का उपयोग करना बंद कर देते हैं
 - (B) फ्रीचर डिस्ट्रिब्यूशन समय के साथ बदलते हैं क्योंकि अटैकर अपनी टैक्टिक्स बदलते हैं, जिससे मॉडल की एक्यूरेसी कम हो जाती है
 - (C) ट्रेनिंग डेटासेट स्टोर करने के लिए बहुत बड़े हो जाते हैं
 - (D) इनफेरेंस के दौरान जीपीयू ओवरहीट हो जाते हैं

In supervised ML-based malware detection, why is “concept drift” a critical challenge?

- (A) Malware authors stop using encryption
- (B) Feature distributions change over time as attackers evolve tactics, reducing model accuracy
- (C) Training datasets become too large to store
- (D) GPUs overheat during inference

- Q82.** एसओसी अलर्ट ट्राइएज वर्कफ्लो में ब्लैक-बॉक्स एमएल मॉडल के फ़ैसलों को समझाने के लिए किस तकनीक का उपयोग किया जाता है?

- (A) ग्रेडिएंट डिसेंट ट्यूनिंग
- (B) एसएचएपी (एसहैपले एडिटिव एक्सप्लानेशन्स) या लाइम
- (C) डॉपआउट रेगुलराइज़ेशन
- (D) बैच नॉर्मलाइज़ेशन

Which technique is used to explain black-box ML model decisions in SOC alert triage workflows?

- (A) Gradient descent tuning
- (B) SHAP (SHapley Additive exPlanations) or LIME
- (C) Dropout regularization
- (D) Batch normalization

- Q83.** एंटरप्राइज़ एंडपॉइंट थ्रेट डिटेक्शन के लिए फ़ेडरेटेड लर्निंग में, प्राइवैसी बनाए रखने का मुख्य लाभ क्या है?

- (A) मॉडल कभी भी रॉ यूज़र डेटा नहीं देखते हैं — सिर्फ़ पैरामीटर अपडेट सेंट्रली एग्रीगेट किए जाते हैं
- (B) सभी एंडपॉइंट एक जैसे हार्डवेयर का उपयोग करते हैं
- (C) ट्रेनिंग सिर्फ़ वीकेंड पर होती है
- (D) डेटा ब्लॉकचेन में स्टोर होता है

In federated learning for enterprise endpoint threat detection, what is the main privacy-preserving advantage?

- (A) Models never see raw user data — only parameter updates are aggregated centrally
- (B) All endpoints use identical hardware
- (C) Training occurs only on weekends
- (D) Data is stored in blockchain

- Q84.** एआइ से चलने वाले एसआइइएम प्लेटफॉर्म में, "टेम्पोरल कोरिलेशन ग्राफ" का मुख्य उद्देश्य क्या है?
- (A) समय के साथ सीपीयू के उपयोग को दिखाना
 - (B) फायरवॉल नियम अपने आप बनाना
 - (C) सस्ते स्टोरेज के लिए लॉग डेटा को कम्प्रेस करना
 - (D) समय और एंटीटी के हिसाब से कम गंभीरता वाले अलर्ट को हाई-फिडेलिटी अटैक कैंपेन से जोड़ना

In an AI-driven SIEM platform, what is the primary purpose of "temporal correlation graphs"?

- (A) To visualize CPU usage over time
- (B) To auto-generate firewall rules
- (C) To compress log data for cheaper storage
- (D) To link low-severity alerts across time and entities into high-fidelity attack campaigns

- Q85.** एआइ-एनआइडीएस में, बहुत ज़्यादा असंतुलित नेटवर्क ट्रैफ़िक (99.99% मामूली) में "एक्यूरेसी" को मुख्य मेट्रिक के तौर पर उपयोग करने का क्या नतीजा होता है?

- (A) मॉडल बहुत सटीक लगता है लेकिन लगभग सभी अटैक से बच जाता है
- (B) लेटेंसी ज़ीरो हो जाती है
- (C) स्विच हर घंटे ऑटो-रीबूट हो जाते हैं
- (D) टीएलएस 1.3 असुरक्षित हो जाता है

In an AI-NIDS, what is the consequence of using "accuracy" as the primary metric in highly imbalanced network traffic (99.99% benign)?

- (A) Model appears highly accurate but misses nearly all attacks
- (B) Latency drops to zero
- (C) Switches auto-reboot every hour
- (D) TLS 1.3 becomes insecure

- Q86.** स्टेटफुल एलएसटीएम -बेस्ड एनआइडीएस, स्लो-एंड-लो अटैक का पता लगाने में स्टेटलेस एमएल मॉडल से बेहतर क्यों परफॉर्म करते हैं?

- (A) वे आइपीवी6 को डिसेबल कर देते हैं
- (B) उन्हें ज़्यादा रैम की ज़रूरत होती है
- (C) एलएसटीएम पैकेट/फ्लो में कनेक्शन हिस्ट्री और सेशन कॉन्टेक्ट याद रखते हैं
- (D) वे सभी यूडीपी ट्रैफ़िक को ऑटो-ब्लॉक कर देते हैं

Why do stateful LSTM-based NIDS outperform stateless ML models in detecting slow-and-low attacks?

- (A) They disable IPv6
- (B) They require more RAM
- (C) LSTMs remember connection history and session context across packets/flows
- (D) They auto-block all UDP traffic

Q87. आइओटी-फोकस्ड एआई-एनआईडीएस में लाइटवेट एमएल मॉडल्स का “एज डिप्लॉयमेंट” क्यों ज़रूरी है?

- (A) क्लाउड एपीआई हर रिक्वेस्ट के हिसाब से चार्ज करते हैं
- (B) लेटेंसी कंस्टेंट और बैंडविड्थ लिमिटेशन के लिए क्लाउड पर राउंड ट्रिपिंग किए बिना लोकल इनफेरेंस की ज़रूरत होती है
- (C) आइओटी डिवाइस को डॉकर कंटेनर पसंद हैं
- (D) एमक्यूटीटी प्रोटोकॉल के लिए एमएल वेट का इंटीजर होना ज़रूरी है

Why is “edge deployment” of lightweight ML models critical in IoT-focused AI-NIDS?

- (A) Cloud APIs charge per request
- (B) Latency constraints and bandwidth limitations require local inference without round tripping to cloud
- (C) IoT devices love Docker containers
- (D) MQTT protocol requires ML weights to be integers

Q88. कौन सा डिफेंस संभावित एडवर्सरियल पर्टर्बेशन को हटाने के लिए इंफेरेंस टाइम पर इनपुट डेटा को साफ तौर पर मॉडिफाई करता है?

- (A) फीचर स्क्वीजिंग
- (B) वेट डिफेंस
- (C) लर्निंग रेट शेड्यूलिंग
- (D) क्रॉस-वैलिडेशन

Which defense explicitly modifies input data at inference time to remove potential adversarial perturbations?

- (A) Feature squeezing
- (B) Weight decay
- (C) Learning rate scheduling
- (D) Cross-validation

Q89. अगर किसी ऑटोनॉमस गाड़ी का ऑब्जेक्ट डिटेक्टर “फिजिकल-वर्ल्ड एडवर्सरियल पैच” के प्रति कमज़ोर हो, तो क्या रिस्क होता है?

- (A) टायर तेज़ी से घूमते हैं
- (B) जीपीएस सिग्नल ज़्यादा मज़बूत हो जाते हैं
- (C) स्टिकर वाले स्टॉप साइन को स्पीड लिमिट साइन के तौर पर गलत तरीके से क्लासिफ़ाई किया जाता है — जिससे खतरनाक ड्राइविंग फ़ैसले लिए जाते हैं
- (D) विंडशील्ड वाइपर रैंडमली एक्टिवेट हो जाते हैं

What risk arises if an autonomous vehicle’s object detector is vulnerable to “physical-world adversarial patches”?

- (A) Tires rotate faster
- (B) GPS signals become stronger
- (C) Stop signs with stickers are misclassified as speed limit signs — causing dangerous driving decisions
- (D) Windshield wipers activate randomly

- Q90.** एडवर्सरियल मशीन लर्निंग में, कौन सा अटैक खास तौर पर ब्लैक-बॉक्स एपीआई से एक्सपोज़्ड कॉन्फिडेंस स्कोर का फायदा उठाकर एक सरोगेट मॉडल को फिर से बनाता है जो टारगेट मॉडल की डिजीजन बाउंड्री का अंदाज़ा लगाता है?
- (A) बाउंड्री अटैक
 - (B) मेंबरशिप इंफरेंस
 - (C) मॉडल एक्सटैक्शन
 - (D) ग्रेडिएंट वैनिशिंग

In adversarial machine learning, which attack specifically exploits exposed confidence scores from black-box APIs to reconstruct a surrogate model that approximates the target model's decision boundary?

- (A) Boundary attack
 - (B) Membership inference
 - (C) Model extraction
 - (D) Gradient vanishing
- Q91.** कौन सी तकनीक मॉडल की उपयोगिता को बहुत ज़्यादा कम किए बिना मेंबरशिप इंफरेंस अटैक के खिलाफ़ सबसे मज़बूत सुरक्षा देती है?
- (A) सिग्मॉइड एक्टिवेशन के बजाय आरइएलयू का उपयोग करना
 - (B) मॉडल की गहराई को 50 लेयर तक बढ़ाना
 - (C) टाइट ϵ बजट के साथ डिफरेंशियल प्राइवेसी (डीपी) के ज़रिए प्रेडिक्शन में लैपलेस नॉइज़ जोड़ना
 - (D) 10,000 एक्स्ट्रा एपॉच के लिए ट्रेनिंग

Which technique provides the strongest protection against membership inference attacks without significantly degrading model utility?

- (A) Using ReLU instead of sigmoid activations
 - (B) Increasing model depth by 50 layers
 - (C) Adding Laplace noise to predictions via differential privacy (DP) with tight ϵ budget
 - (D) Training for 10,000 extra epochs
- Q92.** एक अटैकर मशीन लर्निंग मॉडल को बार-बार क्वेरी करता है और आउटपुट का इस्तेमाल करके यह अंदाज़ा लगाता है कि ट्रेनिंग सेट में कोई खास रिकॉर्ड मौजूद है या नहीं। यह किस तरह का अटैक है?
- (A) ग्रेडिएंट लीकेज अटैक
 - (B) मॉडल इनवर्जन अटैक
 - (C) बैकडोर ट्रिगर एक्टिवेशन
 - (D) मेंबरशिप इनफरेंस अटैक

An attacker queries a machine learning model repeatedly and uses outputs to infer whether a specific record exists in the training set. Which type of attack is this?

- (A) Gradient leakage attack
- (B) Model inversion attack
- (C) Backdoor trigger activation
- (D) Membership inference attack

- Q93.** ट्रेनिंग के दौरान, एक अटैकर कुछ सैंपल में एक ट्रिगर डालता है, इसलिए मॉडल तभी गलत काम करता है जब वह ट्रिगर दिखाई देता है। कौन सी तकनीक डिप्लॉयमेंट से पहले ऐसे बैकडोर पाइज़निंग का पता लगाने में सबसे ज़्यादा मदद कर सकती है?
- (A) फ़्रीचर एक्टिवेशन का स्पेक्ट्रल सिग्नेचर एनालिसिस
 - (B) हर एपोक पर डेटासेट को शफ़ल करना
 - (C) क्वांटाइज़ेशन के साथ ट्रेन्ड मॉडल को कम्प्रेस करना
 - (D) इनफ़रेंस के दौरान ड्रॉपआउट का इस्तेमाल करना

During training, an attacker inserts a trigger into a few samples so the model misbehaves only when that trigger appears. Which technique can most help uncover such backdoor poisoning before deployment?

- (A) Spectral signature analysis of feature activations
- (B) Shuffling the dataset at each epoch
- (C) Compressing the trained model with quantization
- (D) Using dropout during inference

- Q94.** कौन सा ट्रांसपेरेंसी गैप सबसे ज़्यादा तब होता है जब मॉडल ऐसे हाई-डाइमेंशनल लेटेंट फ़ीचर्स पर निर्भर होते हैं जिन्हें इंसान समझ नहीं पाते?
- (A) "क्रिप्टोग्राफ़िक ओपेसिटी," जहाँ मॉडल वेट एन्क्रिप्टेड होते हैं
 - (B) "एल्गोरिदमिक बायस," जहाँ ट्रेनिंग डेटा तिरछा होता है
 - (C) "हार्डवेयर ओपेसिटी," जहाँ जीपीयू पाइपलाइन नॉन-ट्रांसपेरेंट होती हैं
 - (D) "सिमेंटिक ओपेसिटी," जहाँ इनपुट-आउटपुट मैपिंग को समझने लायक कॉन्सेप्ट से लिंक नहीं किया जा सकता

Which transparency gap most often arises when models rely on high-dimensional latent features not interpretable by humans?

- (A) "Cryptographic opacity," where model weights are encrypted
- (B) "Algorithmic bias," where training data is skewed
- (C) "Hardware opacity," where GPU pipelines are non-transparent
- (D) "Semantic opacity," where input-output mappings can't be linked to understandable concepts

- Q95.** एलआइएमइ या एसएचएपी जैसे पोस्ट-हॉक एक्सप्लेनेशन टूल्स में, कौन सी लिमिटेशन रेगुलेटरी कम्प्लायंस के लिए उनके रिलायबिलिटी को सबसे ज़्यादा खतरा पहुँचाती है?
- (A) वे सिर्फ़ जीपीयू पर चलते हैं, जीपीयू पर नहीं
 - (B) एक्सप्लेनेशन समीपता होती है जो डिस्ट्रीब्यूशन शिफ्ट के तहत असली मॉडल लॉजिक से अलग हो सकते हैं
 - (C) उन्हें हर एक्सप्लेनेशन के लिए मॉडल रीट्रेनिंग की ज़रूरत होती है
 - (D) वे डिप्लॉयमेंट के दौरान मॉडल लेटेंसी को एक फिक्स्ड 500 एमएस तक बढ़ा देते हैं

In post-hoc explanation tools like LIME or SHAP, which limitation most threatens their reliability for regulatory compliance?

- (A) They only run on CPUs and not GPUs
- (B) Explanations are approximations that may differ from true model logic under distribution shifts
- (C) They require model retraining for each explanation
- (D) They increase model latency during deployment by a fixed 500 ms

Q96. एल्गोरिदमिक अकाउंटैबिलिटी के संदर्भ में, कौन सा शब्द किसी ऑर्गनाइज़ेशन की उस ज़िम्मेदारी को बताता है जिसमें वह अपने AI सिस्टम के फ़ैसलों को सही ठहराता है, जब वे फ़ैसले लोगों पर बुरा असर डालते हैं?

- (A) एल्गोरिदमिक फेयरनेस
- (B) पोस्ट-हॉक जस्टिफ़िबिलिटी
- (C) नॉर्मेटिव अकाउंटैबिलिटी
- (D) एक्सप्लेनेटरी गैप क्लोजर

In the context of algorithmic accountability, which term refers to the obligation of an organization to justify its AI system's decisions when those decisions adversely affect individuals?

- (A) Algorithmic Fairness
- (B) Post-hoc Justifiability
- (C) Normative Accountability
- (D) Explanatory Gap Closure

Q97. एडवर्सरियल मशीन लर्निंग में, "अटैक के अंतर्गत बायस एम्प्लीफिकेशन" का अर्थ है:

- (A) अटैकर मॉडल की फेयरनेस को कम करने के लिए बायस सैंपल इंजेक्ट करते हैं
- (B) डिफेंसिव मैकेनिज्म अनजाने में इवेजन अटैक के दौरान गुप्स में फॉल्स पॉजिटिव रेट में अंतर बढ़ाते हैं
- (C) पॉइज़निंग के बाद रीट्रेन होने पर मॉडल्स मेजोरिटी गुप्स के लिए ज़्यादा एक्यूरेट हो जाते हैं
- (D) फेयरनेस कंस्ट्रेंट रोबस्टनेस को कम करते हैं, जिससे मॉडल्स को बेवकूफ बनाना आसान हो जाता है

In adversarial machine learning, "bias amplification under attack" refers to:

- (A) Attackers injecting biased samples to degrade model fairness
- (B) Defensive mechanisms unintentionally increasing disparity in false positive rates across groups during evasion attacks
- (C) Models becoming more accurate for majority groups when retrained after poisoning
- (D) Fairness constraints reducing robustness, making models easier to fool

Q98. "सिक््योरिटी-फेयरनेस ट्रेडऑफ़" घटना से पता चलता है कि:

- (A) फेयर मॉडल असल में खराब उदाहरणों के लिए कम मज़बूत होते हैं
- (B) रेगुलेटरी कम्प्लायंस ज़्यादा जोखिम वाले माहौल में मॉडल की सटीकता को कम करता है
- (C) हमलों के खिलाफ़ मॉडल को मज़बूत बनाने से प्रायः डेमोग्राफिक ग्रुप में परफॉर्मंस में अंतर बढ़ जाता है
- (D) एक्सप्लेनेबिलिटी टूल ऐसी कमज़ोरियाँ लाते हैं जिनका हमलावर फ़ायदा उठा सकते हैं

The "security-fairness tradeoff" phenomenon suggests that:

- (A) Fair models are inherently less robust to adversarial examples
- (B) Regulatory compliance reduces model accuracy in high-risk environments
- (C) Hardening models against attacks often exacerbates performance disparities across demographic groups
- (D) Explainability tools introduce vulnerabilities exploitable by attackers

- Q99. बॉर्डर कंट्रोल के लिए फेशियल रिकग्निशन सिस्टम को ट्रेनिंग देते समय, ज्योग्राफिकली स्क्यूड ट्रेनिंग डेटा का उपयोग करने से मुख्य रूप से किस तरह का बायस होता है?
- (A) लेबल चॉइस बायस
 - (B) एग्रीगेशन बायस
 - (C) रिप्रेजेंटेशन बायस
 - (D) मेज़रमेंट बायस

When training a facial recognition system for border control, using geographically skewed training data leads primarily to which type of bias?

- (A) Label Choice Bias
- (B) Aggregation Bias
- (C) Representation Bias
- (D) Measurement Bias

- Q100. रियल-टाइम नेटवर्क इंट्रूज़न डिटेक्शन सिस्टम में बायस की ऑडिटिंग के लिए कौन सा इवैल्यूएशन प्रोटोकॉल ज़रूरी है?
- (A) स्टैटिक होल्डआउट टेस्टिंग
 - (B) काउंटरफैक्टुअल फेयरनेस स्टेस-टेस्टिंग
 - (C) स्ट्रैटिफाइड सैंपलिंग के साथ के-फोल्ड क्रॉस-वैलिडेशन
 - (D) स्लाइडिंग विंडो पैरिटी चेक के साथ कंटीन्यूअस फेयरनेस मॉनिटरिंग

Which evaluation protocol is essential for auditing bias in real-time network intrusion detection systems?

- (A) Static holdout testing
- (B) Counterfactual fairness stress-testing
- (C) K-fold cross-validation with stratified sampling
- (D) Continuous fairness monitoring with sliding window parity checks

