Standard requirements for storing credit card information (PCI security): https://www.pcisecuritystandards.org/pci_security/

Source: https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks/

1. Denial of Service (DoS) and Distributed Denial of Service (DDoS)
    a. Explanation: Overwhelm system resources so that the server is not able to serve others (no direct benefit to attacker)
    b. Types:
        i. TCP SYN flood attack
            1. Explanation: Flood process queue with connection requests, but let them timeout, thus crashing the system or making it unusable
            2. Countermeasures: Place servers behind firewall, increase size of connection queue and decrease timeout on open connections
        ii. Teardrop attack
            1. Explanation: Send erroneous offset fields to system, causing the system to fail and crash when reconstructing
            2. Countermeasures: If no patches to protect from DoS attacks, disable SMBv2 and block ports 139 and 445
        iii. Smurf attack
            1. Explanation: uses IP spoofing to send requests from another's IP, such that returned responses to that IP overwhelm it
            2. Countermeasures: Disable IP-directed broadcasts at the routers
        iv. Ping-of-death attack
            1. Explanation: Sending IP packets with an erroneously large size, causing the system to have a buffer overflow and other crashes when reassembling the packet.
            2. Countermeasures: Use a firewall that checks fragmented IP packets for their maximum size
        v. Botnets
            1. Explanation: Millions of infected devices sending attacks against a system with the intent of overwhelming its bandwidth and processing capabilities.
            2. Countermeasures: RFC3704 filtering and Black hole filtering
2. Man-in-the-middle (MitM) attack
    a. Explanation: Hacker inserts self between communications of a client and a server
    b. Types:
        i. Session Hijacking:
            1. Explanation: Hijacks session between trusted client and network server by substituting its IP address for the trusted client's.
        ii. IP Spoofing:
            1. Explanation: Convince the system that it is communicating with a known, trusted entity so that it will provide the hacker with access to the system (uses trusted entity's IP).
        iii. Replay

        1. Explanation: Saves old messages and tries to send them again later, thus impersonating one of the participants.
        2. Countermeasures: Use session timestamps or nonce (random number/string that changes with time)
    c. General Countermeasures: No single prevent all MitM attacks. Encryption and **digital certificates** provide an effective safeguard.
3. *Driveby attack*
    a. Explanation: Hackers plant a malicious script into HTTP (HTML?) or PHP code on a page in a website.
    b. Countermeasure: Avoid sketchy sites and keep systems up to date
4. Password attack
    a. Explanation: stealing a person's password (can be my brute-force or a dictionary attack)
    b. Countermeasure: implement an account lockout policy
5. SQL injection attack
    a. Explanation: Taking advantage of input to SQL queries
    b. Countermeasure: (ASP.NET less likely to have this problem) apply least privilege (https://digitalguardian.com/blog/what-principle-least-privilege-polp-best-practice-information-security-and-compliance#:~:text=Examples%20of%20the%20Principle%20of%20Least%20Privilege&text=User%20Account%20with%20Least%20Privilege,add%20records%20to%20that%20database.) measure in permissions for database, use stored procedures,, and prepared statements (parameterized queries).
6. Cross-Site Scripting (XSS) attack
    a. Explanation: Injects malicious JS into website's database. When the victim requests the page, their browser executes the malicious script.
    b. Countermeasure: Sanitize data inputs by users in an HTTP request before reflecting it back. Validate all data and filter or escape it before echoing it back. Convert special characters to respective HTML and URL encoded equivalents. Give users the option to disable client-side scripts.
7. Eavesdropping attack
    a. Explanation: Occurs through interception of network traffic to obtain information sent over that network. Can be passive (listen for message transmission) or active (disguises self as a friendly unit and grabs the information -- called probing, scanning, or tampering).
    b. Countermeasure: Data encryption is the best option
8. Birthday attack
    a. Explanation: Made against hash algorithms that verify the integrity of a message, software, or digital signature. If able to find an input that generates the same message digest (MD) or output, then can replace the user's MD with own and pretend to be the user.
9. Malware attack
    a. Explanation: Unwanted software installed on your own system without consent.
    b. Types:

i. Macro viruses -- infect applications (like Microsoft Word)
ii. File infectors -- Attack to .exe files when the code for them is loaded
iii. System or boot-record infectors -- attaches to master boot record on hard disks
iv. Polymorphic viruses -- conceals self via encryption and decryption, infects an area of code
v. Stealth viruses -- take over system functions to conceal self
vi. Trojans -- hides in a useful program (unlike viruses, does not self replicate. Instead, establishes a backdoor for launching attacks)
vii. Logic bombs -- attack to applications and occur when specific logical condition met
viii. Worms -- Do not attack but instead propagate across networks and computers.
ix. Droppers -- Used to install viruses on computers
x. Ransomware -- Blocks access to victim's data and threatens to publish or delete if a random is not paid
xi. Adware -- Can be automatically downloaded when browsing a website
xii. Spyware -- collects information about users, their computers, and their browning habits

Source: https://www.toptal.com/security/10-most-common-web-security-vulnerabilities

10. Broken Authentication
    a. Explanation: URL contains session id thus leaking it, password not encrypted in storage or transit, session ids are predictable, session fixation is possible, session hijacking is possible (timeouts not implemented right or using HTTP)
    b. Countermeasure: Use of a framework (like ASP.NET)
11. XSS (same from earlier)
    a. Additional Countermeasure: Don't return HTML tags to the client. Sanitize returned strings and special characters, such as through the user of regular expressions.
12. Insecure Direct Object References
    a. Explanation: exposing an internal object (i.e. file or database key) to the user.
    b. Countermeasure: Have user authorization, sanitize user inputs, and use session variables
13. Sensitive data exposure
    a. Countermeasure: Always encrypt sensitive data, including in transit and at rest. Always have passwords be hashed.
        i. In transit: Use HTTPS in transit and do not accept non-HTTPS connections. Have secure flags on cookies.
        ii. In storage: Minimize exposure, encrypt sensitive data, hash passwords (such as with bcrypt), be PCI compliant for credit card data
    b. Store encryption keys in a separate, segregated, protected area
14. Missing function level access control
    a. Countermeasure: On the server side, ensure that authorization is always done.

15. Cross Site Request Forgery (CSRF)
    a. Explanation: When a third party site makes a request to a target site using the user's browser with their cookies and session.
    b. Countermeasure: Store a secret token in a hidden form field which is inaccessible from the third party site which is required to have for modifying sensitive data.
        i. Also, never use HTTP GET to alter resources
16. Server Side Request Forgery
    a. Server-Side Request Forgery (SSRF) is a web app vulnerability that allows attackers to force the web application server to make requests to resources it normally wouldn't. For example, a web app may have the functionality to produce screenshots of other websites when a user supplies a URL. This is perfectly valid functionality, however, URLs can also be made for internal IP addresses (e.g. 192.168.1.1, 10.10.10.10, 127.0.0.1 etc.) as well as internal-only hostnames (e.g. localhost, WIN2019SERV.CORP). If a web developer is not careful, an attacker could provide the app with these and manage to screenshot internal resources, which often have less protections.
    b. To counter this, user-provided URLs can be checked before they are requested, to ensure that malicious values are not being used. However, due to the complex nature of URLs themselves, there are often many things an attacker can do to bypass these checks.
    c. Note that while the example of SSRF used in this task is effectively a Remote File Inclusion (RFI) vulnerability as well, not every SSRF is. Some SSRF vulnerabilities only trigger a DNS lookup, while others may not return any kind of response to the web app, but can still be used to "port scan" internal systems by measuring the time each request takes to complete. In other cases, SSRF may be used as a form of Denial of Service (DoS) since the attacker can continually request that the server download large files simultaneously (taking up memory, disk space, and network bandwidth).

PCI Compliance (https://digitalguardian.com/blog/what-pci-compliance)
12 Requirements:

1. Use and maintain firewalls
2. Use proper password protections
3. Protect cardholder data
    a. Two-fold protection: Encrypt card data and then encrypt encryption key
4. Encrypt transmitted data
    a. Never send account numbers to unknown locations
5. Use and maintain anti-virus
6. Properly updated software
7. Restrict data access
8. Unique IDs for access
    a. All individuals with access to cardholder data must have their one unique credentials and identification for access

9. Restrict physical access
10. Create and maintain access logs
    a. Whenever card data is used, transferred, or accessed
11. Scan and test for vulnerabilities
12. Document policies