Web Security I

Question 1  *Intrusion Detection*  
FooCorp is deciding which intrusion detection method to employ in a few target scenarios. In the following parts, consider which of the intrusion detection methods learned in class would be most appropriate (NIDS, HIDS, or logging), and justify why.

(a) FooCorp is hosting a web application over HTTPS and needs to detect any use of blacklisted characters in real time.

(b) FooCorp is hosting a web application over HTTP and wants to pass all user traffic through an anomaly detection algorithm (which uses some computationally expensive machine learning). The web application needs to have low latency when many users are online during the day.

(c) FooCorp uses the Simple Mail Transfer Protocol (SMTP) for email and wants to be able to quickly detect phishing attacks against any of their internal computers. SMTP runs on port 25 and is unencrypted.

(d) FooCorp doesn’t trust its employees and sets-up a NIDS to monitor their traffic. However, many employees use TLS, hindering what can be monitored.

FooCorp decides to turn their NIDS into a *Man-in-the-Middle*, giving it a certificate that all the employee’s computers trust. Whenever an employee visits a website they complete a TLS handshake with the NIDS, the NIDS connects to the requested website using TLS, and any traffic between the employee and website is forwarded across the two TLS links by the NIDS.

Which security principle does this violate? Describe everything an attacker can do if they compromise the NIDS.
FooCorp now needs to decide which intrusion detection technique to employ in a few target scenarios. In the following parts, consider which technique would be most appropriate (signature-based, anomaly-based, specification-based, or behavioral), and justify why.

(e) FooCorp wants to detect script kiddies (hackers who primarily use publically available tools or exploits)

(f) FooCorp wants to detect a seasoned l33t h4x0r who uses crafts custom exploits for each attack

(g) FooCorp wants to detect publically-available malware that a hacker manually tweaks to avoid signature checks

(h) FooCorp wants to detect any attempts by their employees to access the protected /etc/passwd file
Question 2  *Second-order linear... err I mean SQL injection*

Alice likes to use a startup, NotAmazon, to do her online shopping. Whenever she adds an item to her cart, a POST request containing the field `item` is made. On receiving such a request, NotAmazon executes the following statement:

```go
cart_add := fmt.Sprintf("INSERT INTO cart (session, item) " +
    "VALUES ('%s', '%s')", sessionToken, item)
db.Exec(cart_add)
```

Each item in the cart is stored as a separate row in the `cart` table.

(a) Alice is in desperate need of some toilet paper, but the website blocks her from adding more than 72 rolls to her cart 😞 Describe a POST request she can make to cause the `cart_add` statement to add 100 rolls of toilet paper to her cart.

When a user visits their cart, NotAmazon populates the webpage with links to the items. If a user only has one item in their cart, NotAmazon optimizes the query (avoiding joins) by doing the following:

```go
cart_query := fmt.Sprintf("SELECT item FROM cart " +
    "WHERE session='%s' LIMIT 1", sessionToken)
item := db.Query(cart_query)
link_query = fmt.Sprintf("SELECT link FROM items WHERE item='%s'", item)
db.Query(link_query)
```

After part (a), Alice recognizes a great business opportunity and begins reselling all of NotAmazon's toilet paper at inflated prices. In a panic, NotAmazon fixes the vulnerability by parameterizing the `cart_add` statement.

(b) Alice claims that parameterizing the `cart_add` statement won’t stop her toilet paper trafficking empire. Describe how she can still add 100 rolls of toilet paper to her cart. Assume that NotAmazon checks that `sessionToken` is valid before executing any queries involving it.