Blacklisting Versus Whitelisting

In this lesson, we'll study blacklisting and whitelisting.

We'll cover the following

- Blacklisting
  - The problem with blacklisting
- A practical example
- Blacklisting vs whitelisting

Blacklisting

When implementing systems that require discarding elements based on an input (e.g., rejecting requests based on an IP address or a comment based on certain words) you might be tempted to use a blacklist in order to filter elements out.

The problem with blacklisting

The inherent problem with blacklisting is the approach we're taking. It allows us to specify which elements we think are unsafe, making the assumption that we know everything that could hurt us. From a security perspective, that's the equivalent of us wearing summer clothes because we're well into June, without looking out the window in order to make sure it's actually sunny. We make assumptions without having the whole picture, and it could hurt us.

If you were thinking of filtering out comments based on a blacklist of words, you would probably start by describing a blacklist of five to ten words. When coming up with the list you might forget words such as *j3rk*, or reject genuine comments mentioning "Dick Bavetta", a retired NBA referee.

Now, comments aren't always the most appropriate example in terms of security, but you get the gist of what we're talking about. It's hard to know everything that's going to hurt us well in advance, so whitelisting is generally a more cautious approach, allowing us to specify what input we trust.

A practical example

A practical example

A more practical example would be logging. You will want to whitelist what can be logged rather than the opposite. Take an example object such as:

```
{
    email: "lebron@james.com",
    password: "King_James",
    credit_card: "1111 2222 3333 4444",
    birthday: "1984-12-30",
}
```

You could possibly create a blacklist that includes `password` and `credit_card`, but what would happen when another engineer in the team changes fields from snake_case to camelCase?

Our object would become:

```
{
    email: "lebron@james.com",
    password: "King_James",
    creditCard: "1111 2222 3333 4444",
    birthday: "1984-12-30",
}
```

You might end up forgetting to update your blacklist, leading to the credit card number of your customers being leaked all over your logs.

Blacklisting vs whitelisting

As you've probably realized, the choice of utilizing a blacklist or a whitelist depends on the context you're operating in. If you're exposing a service on the internet (such as facebook.com), then blacklisting is definitely not going to work, as that would mean knowing the IP address of every genuine visitor, which is impossible.

From a security perspective, whitelisting is a better approach but is often impractical. Choose your strategy carefully after reviewing both options, none of the above is suitable without prior knowledge of your system, constraints and requirements.

In the next lesson, we'll study some logging secrets.