# Recovery From Web Application Security Incidents

## Objectives:

At the end of this episode, I will be able to:

Understand what recovery from Web Application security incidents is.

Explain what the steps that the IH&R team should follow for recovery from web application security incidents are.

Identify what Web Application Fuzz Testing (fuzzing) is.

Define what the three Fuzz Testing Strategies that IH&R team members should be aware of are.

## External Resources:

Recovery from Web Application Security Incidents

What are the steps that the IH&R team should follow for Recovery? -

```
▪ Identify the vulnerabilities attackers had exploited and patch them

▪ Scan all the web application resources such as servers and databases for
malware and remove them

▪ Increase the log storage limit and increase disk space

▪ Check the web application backups for traces of attack and clean them

▪ Change the administrative passwords of all devices and resources

▪ Configure firewall, IDS, and other security solutions to detect the
identified attack using signatures and behavior analysis

▪ Improve the security of the network perimeter by implementing strict WAF,
IDS, and ACLs policies and rules

▪ Use the cleaned, verified, and patched backup version of the web
application to restore the services

▪ Restart any services terminated as a part of containment process

▪ Use an access control to matrix and define access control rules with list
of accessible and authorized requests
```

What is Web Application Fuzz Testing (fuzzing)? -

A black box testing method. It is a quality checking and assurance technique used to identify coding errors and security loopholes in web applications. Huge amounts of random data called ‘fuzz’ will be generated by the fuzz testing tools (Fuzzers) and used against the target web application to discover vulnerabilities that can be exploited by various attacks.

Incident responders and web developers employ this fuzz testing technique to test the robustness and immunity of the developed web application against attacks like buffer overflow, DOS, XSS, and SQL injection.

Fuzz Testing Strategies

```
▪ Mutation-Based - the current data samples create new test data and the new
test data will again mutate to generate further random data. This type of
testing starts with a valid sample and keeps mutating until the target is reached.

▪ Generation-Based - the new data will be generated from scratch and the
amount of data to be generated are predefined based on the testing model.

▪ Protocol-Based - protocol fuzzer sends forged packets to the target
application that is to be tested. This type of testing requires detailed
knowledge of protocol format being tested.
```