ICT2203 Network Security

# Attacks and Defense of IP Networks with Firewalls II

## Lab 6.2 Notes

## Application Layer Protocol Inspection

# What To Do in This LAB?

SINGAPORE INSTITUTE OF TECHNOLOGY

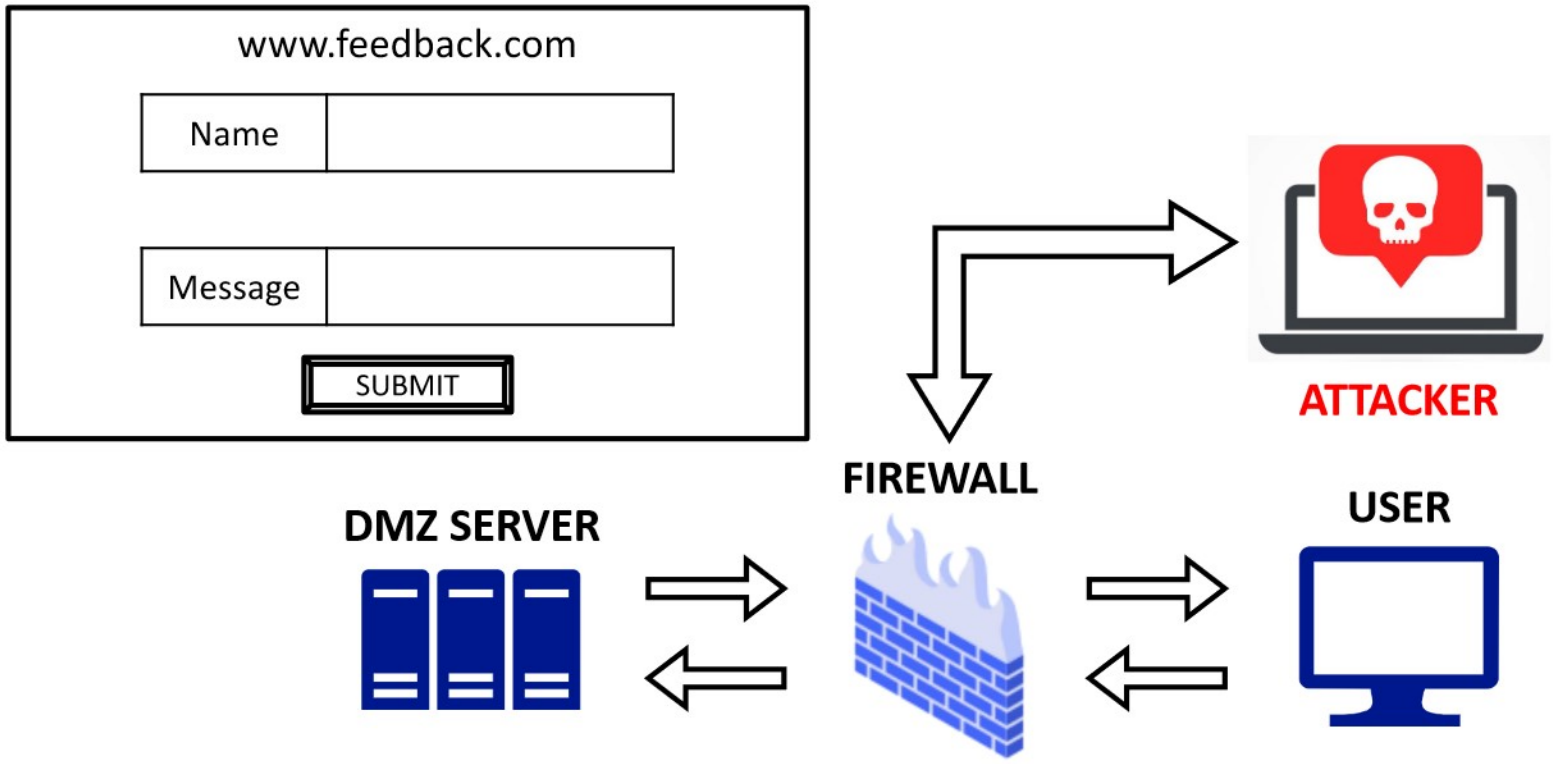
- Different Application Layer Attacks: SQL Injection, Cross-Site scripting, Session Hijacking, Remote Code Execution, File Upload Abuse, Distributed Denial of Service and so on….

- In this LAB, we will implement Cross-Site Scripting (XSS) Attack using DVWA

- We will then use ASA to defend the Cross-Site Scripting Attack

# Cross-Site Scripting (XSS) Attack

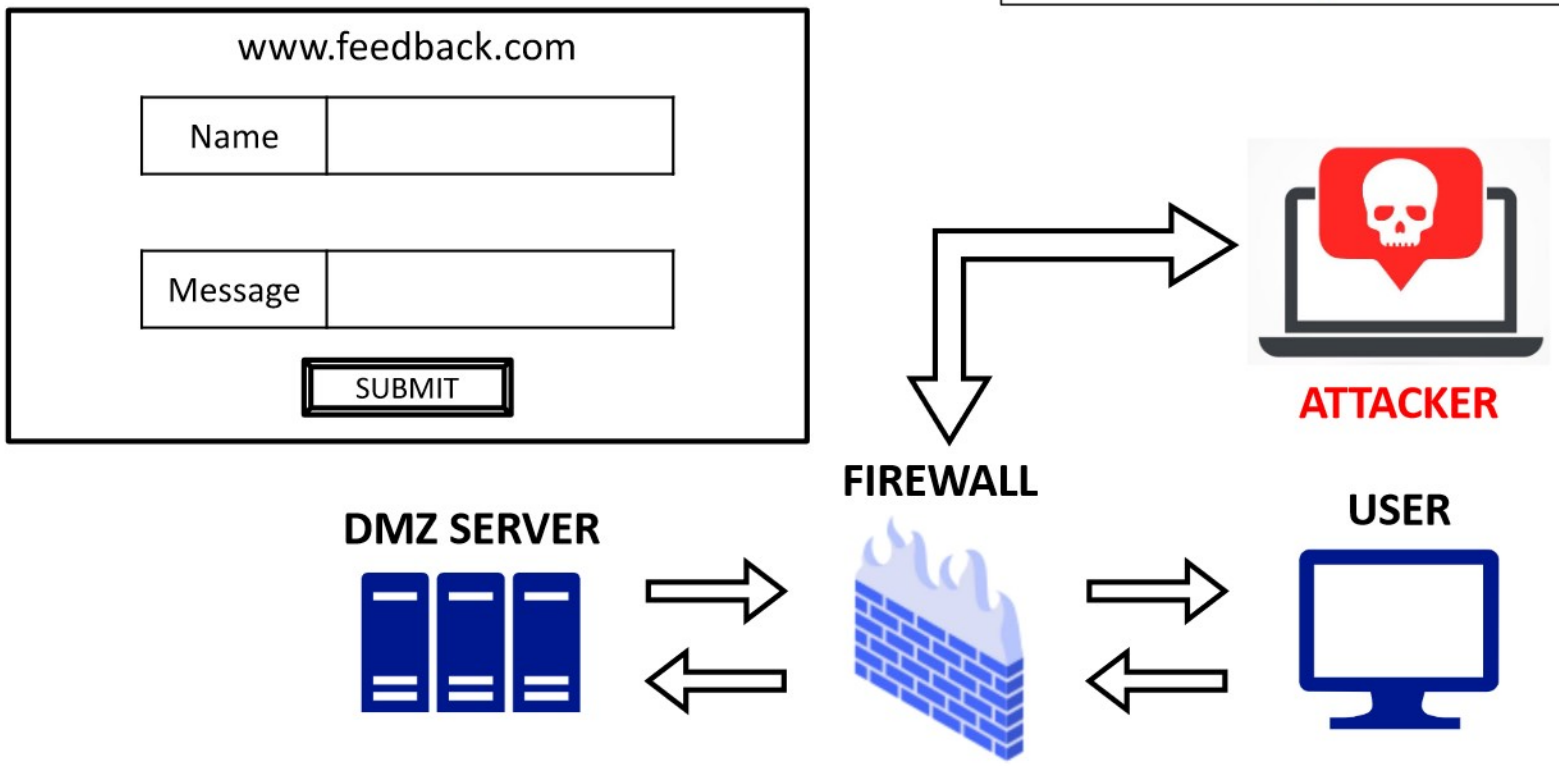
- What is XSS?
  - Occurs when an attacker injects code.
  - Types: Reflected, **Stored** and DOM-Based.



# Cross-Site Scripting (XSS) Attack

**Step 1**

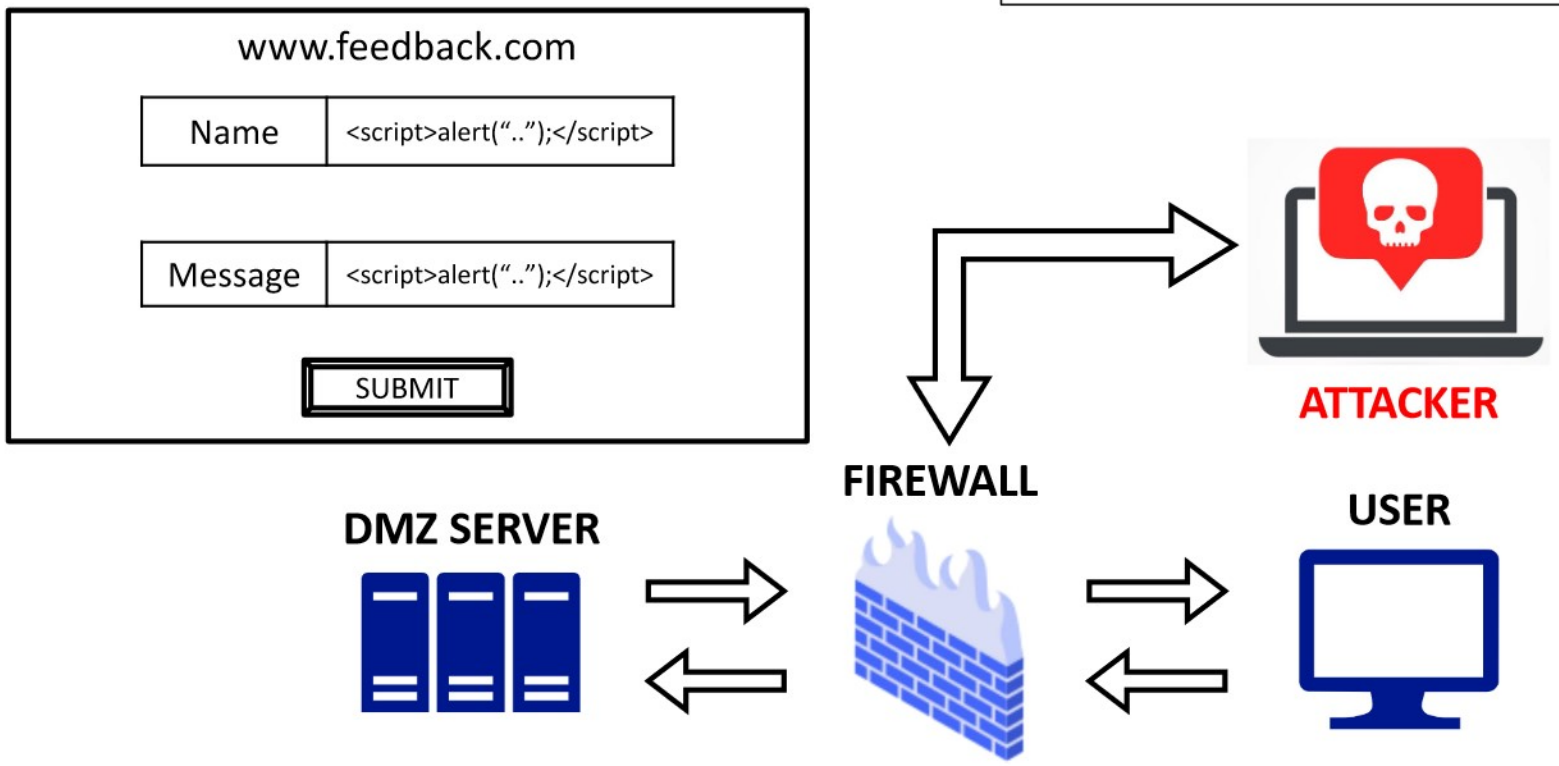
The attacker opens the website on his/her PC



# Cross-Site Scripting (XSS) Attack

**Step 2**

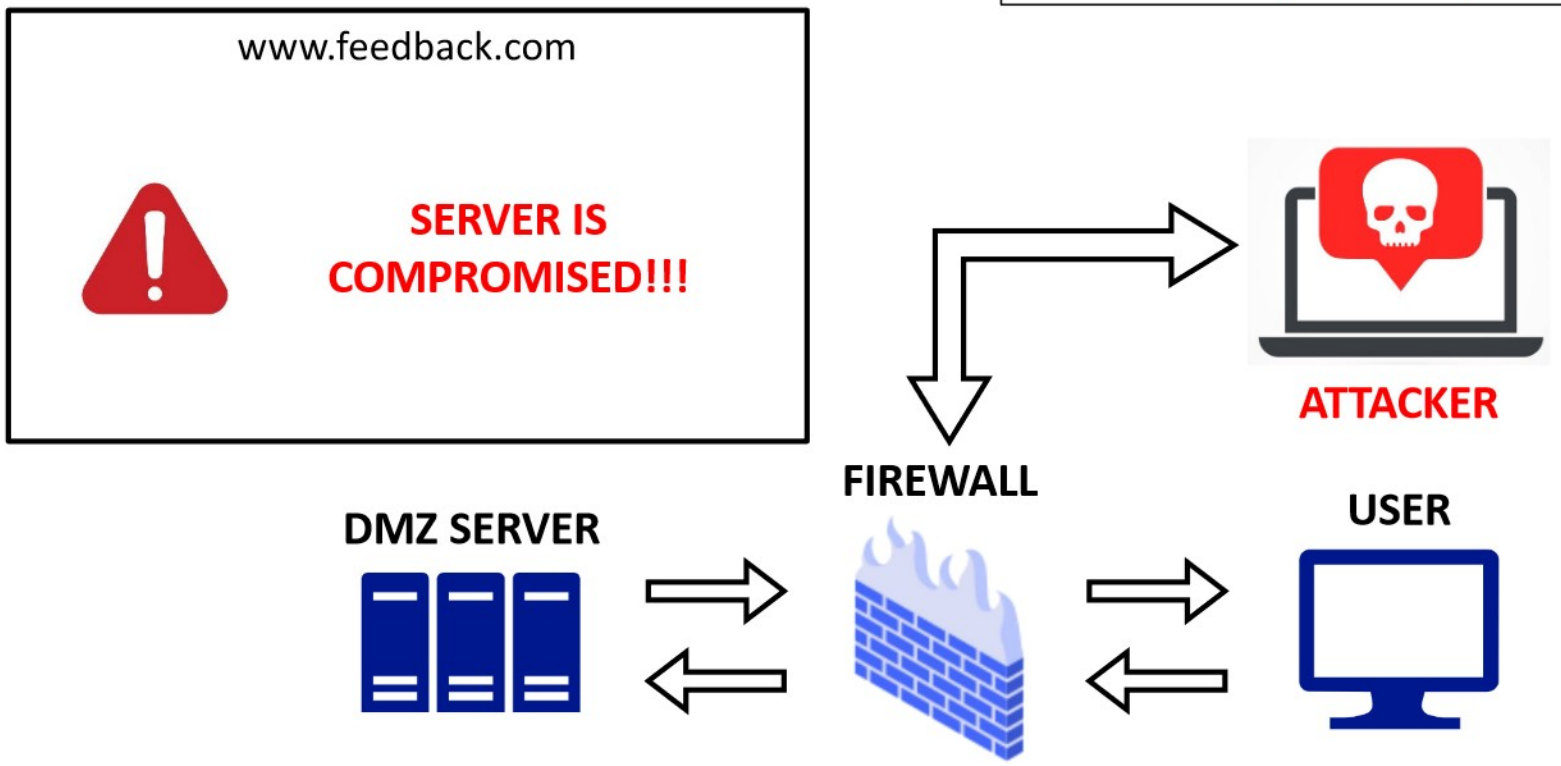
Insert HTML/Java Script in the Feedback



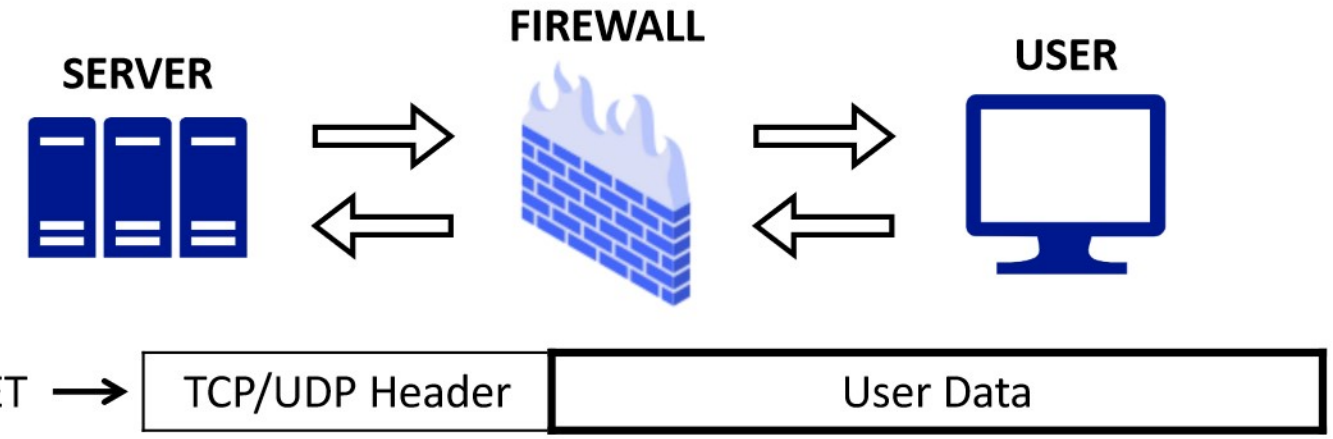
# Cross-Site Scripting (XSS) Attack

**Step 3**

Access the website from the USER PC and your compromised



# XSS – Defence:




Application Layer Inspection is used to inspect this part of the packet

**ASA's Application Layer Protocol Inspection**


https://www.cisco.com/c/en/us/td/docs/security/asa/asa72/configuration/guide/conf_gd/inspect.html#wp1514315

# ASA: Application Layer Inspection

SINGAPORE INSTITUTE OF TECHNOLOGY

## Configuration of Application Inspection in ASA

| Step 1 | Step 2 | Step 3 |
| --- | --- | --- |
| Configure application layer protocol inspection class map | Configure application layer protocol inspection policy map | Apply inspection policy map over MPF layer 3/4 policy map discussed in the last lecture |

# ASA: Application Layer Inspection

SINGAPORE INSTITUTE OF TECHNOLOGY

## Step 1: Application layer protocol inspection class map

```
host(config)# class-map type inspect application [match-all | match-any]
              application_class_map
```

## Step 2: Application layer protocol inspection policy map

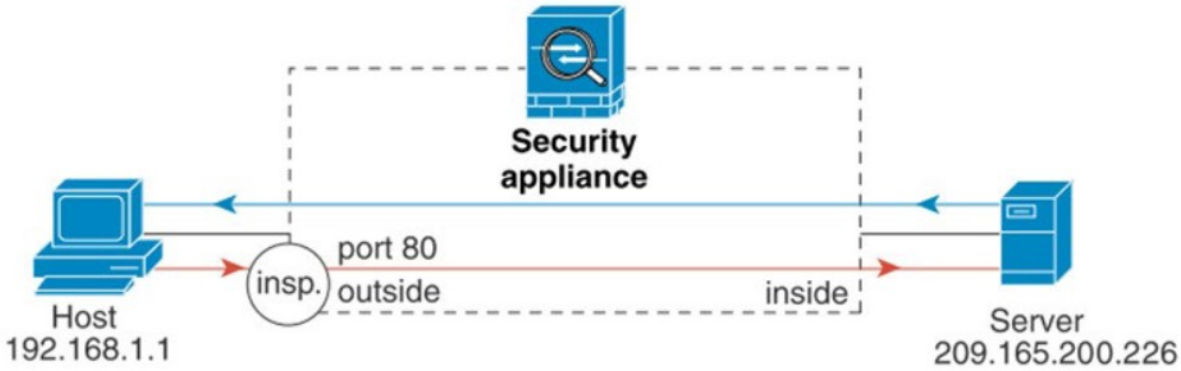
```
host(config)# policy-map type inspect application application_policy_map
```

## Step 3: Inspection policy map over MPF layer 3/4 policy map

```
host(config)# policy-map policy_map_name
host(config-pmap)# class class_map_name
host(config-pmap-c)# inspect application application_policy_map
```

# How to Defend XSS?

We will use the HTTP Application Layer Protocol Inspection




```
ciscoasa(config)# class-map type inspect http [match-all | match-any] cmap_name
ciscoasa(config-cmap)# match [not] {request | response | req-resp} ...
ciscoasa(config-cmap)# ...
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map type inspect http pmap_name
ciscoasa(config-pmap)# class cmap_name
ciscoasa(config-pmap-c)# {drop-connection [log] | log | reset [log]}
ciscoasa(config-pmap-c)# exit
```

# HTTP Inspection – STEP 1

HTTP inspection policy map can be configured using the match commands as below

```
ciscoasa(config-cmap)# match [not] request header {[field]
[regex [regex_name | class regex_class_name]] |
[length gt max_length_bytes | count gt max_count_bytes]}

ciscoasa(config-cmap)# match [not] request method {[method] |
[regex [regex_name | class regex_class_name]]

ciscoasa(config-cmap)# match [not] request uri {regex [regex_name | class
regex_class_name] | length gt max_bytes}

ciscoasa(config-cmap)# match [not] request args regex [regex_name | class
regex_class_name]

ciscoasa(config-cmap)# match [not] request body {regex [regex_name | class
regex_class_name] | length gt max_bytes}

ciscoasa(config-cmap)# match [not] req-resp content-type mismatch

ciscoasa(config-cmap)# match [not] response header {[field]
[regex [regex_name | class regex_class_name]] |
[length gt max_length_bytes | count gt max_count]}

ciscoasa(config-cmap)# match [not] response status-line {regex [regex_name | class
regex_class_name]}

ciscoasa(config-cmap)# match [not] response body {[active-x] | [java-applet] |
[regex [regex_name | class regex_class_name]] | length gt max_bytes}
```

# HTTP Inspection – STEP 1

Regular expression (regex) is a powerful way to specify search patterns for matching and can be configured in two ways as follows:

## Single **Regular Expression**

```
ciscoasa(config)# regex regex_name regular_expression
ciscoasa(config)# regex ...
```

## Group of **Regular Expressions** using class-map

```
ciscoasa(config)# class-map type regex match-any regex_cmap_name
ciscoasa(config-cmap)# match regex regex_name
ciscoasa(config-cmap)# match regex ...
ciscoasa(config-cmap)# exit
```

## While formulating your regular expression, you may test it as follows:

```
ciscoasa# test regex input_text regular_expression
```

```
ciscoasa# test regex https://shadycontent.com/toolz https?://
INFO: Regular expression match succeeded.
```

# HTTP Inspection – STEP 1

SINGAPORE INSTITUTE OF TECHNOLOGY

## The list of regular expression metacharacters and their usages

| Metacharacter | Name | Function |
|---|---|---|
| . | Dot | Matches any single character.<br>Example: b.d matches bad, bbd, bcd, bdd, bed, and so on |
| ( ) | Subexpression | Groups the characters inside the parentheses as a single expression for matching with other metacharacters. |
| \| | Or | Matches either expression that \| separates.<br>Example: com\|net matches whatever.com or whatever.net<br>Example: Ma(r\|y) matches Mar or May |
| ? | Question mark | Matches 0 or 1 of the expression just before the ?.<br>Example: e?smtp matches smtp (zero e's) or esmtp (1 e)<br>Example: (12)? matches 4444, 12444, 1212444, and so on |
| * | Asterisk | Matches 0, 1, or any number of the expression just before the *.<br>Example: w* matches cisco.com and www.cisco.com |
| + | Plus | Matches at least one of the expressions just before the +.<br>Example: w+ matches www.cisco.com, but not cisco.com |

# HTTP Inspection – STEP 1

SINGAPORE INSTITUTE OF TECHNOLOGY

## The list of regular expression metacharacters and their usages

| Metacharacter | Name | Function |
|---|---|---|
| {n} | Repeat | Matches if the expression just before {n} is repeated exactly n times.<br>Example: (test){2} matches testtest but not testtesttest |
| {n,} | Minimum repeat | Matches if the expression just before {n,} is repeated at least n times.<br>Example: (test){2} matches testtest and also testtesttest |
| [abc] | Character class | Matches any of the characters listed between the square brackets.<br>Example: [dfhl]og matches dog, fog, hog, and log, but not frog |
| [^abc] | Not character class | Matches any character that is not listed between the brackets.<br>Example: [^dfhl]og matches cog, but not dog, fog, hog, or log |
| [a-c] | Character range class | Matches any character in the range from a to c.<br>Example: [a-z] matches any lowercase letter, [A-Z] matches any uppercase letter, [0-9] matches any digit |

# HTTP Inspection – STEP 1

SINGAPORE INSTITUTE OF TECHNOLOGY

## The list of regular expression metacharacters and their usages

| Metacharacter | Name | Function |
|---|---|---|
| ^ | Caret | Matches the beginning of a line; any expression following the caret will be matched only if it appears at the beginning of a line.<br>Example: ^Dear matches "Dear John" but not "John Dear" |
| \ | Escape | The metacharacter following \ will be treated as a literal character; this is useful when you need to match against something that is normally interpreted as a metacharacter.<br>Example: *Test matches *Test* |
| \r | Carriage return | Matches a carriage return character (ASCII 13 or 0x0d). |
| \n | Newline | Matches a newline character (ASCII 10 or 0x0a). |
| \t | Tab | Matches a tab character (ASCII 9 or 0x09). |
| \f | Form feed | Matches a form feed character (ASCII 12 or 0x0c). |
| \xNN | Escaped hex number | Matches an ASCII character that has the two-digit hex code NN.<br>Example: \x20 matches a space (ASCII 32) |
| \NNN | Escaped octal number | Matches an ASCII character that has the three-digit octal code NNN.<br>Example: \040 matches a space (ASCII 32) |

# HTTP Inspection – STEP 2

Configure HTTP inspection policy map to specify actions to be performed on matched traffic

```
ciscoasa(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] |
rate-limit message_rate}
```

- **drop** keyword drops the packets
- **send-protocol-error** keyword sends a protocol error message
- **drop-connection** keyword drops the packet and closes the connection
- **mask** keyword masks out the matching portion of the packet
- **reset** keyword drops the packet, closes the connection, and sends a TCP reset
- **log** keyword sends a system log message
- **rate-limit** *message_rate* argument limits the rate of messages

Note that not all actions are allowed for different match command.

Use CLI ? help for exact options available.

# HTTP Inspection – STEP 2

In addition, HTTP inspection policy map has special parameters option for specifying action to take if HTTP protocol violation occurs.

To enter into **parameters configuration** mode in inspection policy map:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

To **configure** action for **HTTP protocol violation**, enter the command:

```
ciscoasa(config-pmap-p)# protocol-violation [action
                          [drop-connection | reset | log]]
```

# HTTP Inspection – STEP 1 & 2

To summarise, here is an example of HTTP inspection class map and inspection policy map.

```
ciscoasa(config)# regex url_example example\.com
ciscoasa(config)# regex url_example2 example2\.com
ciscoasa(config)# class-map type regex match-any URLs
ciscoasa(config-cmap)# match regex url_example
ciscoasa(config-cmap)# match regex url_example2

ciscoasa(config)# class-map type inspect http match-all http-traffic
ciscoasa(config-cmap)# match req-resp content-type mismatch
ciscoasa(config-cmap)# match request body length gt 1000
ciscoasa(config-cmap)# match not request uri regex class URLs

ciscoasa(config)# policy-map type inspect http http-map1
ciscoasa(config-pmap)# class http-traffic
ciscoasa(config-pmap-c)# drop-connection log
ciscoasa(config-pmap-c)# match req-resp content-type mismatch
ciscoasa(config-pmap-c)# reset log
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# protocol-violation action log
```

# HTTP Inspection – STEP 3

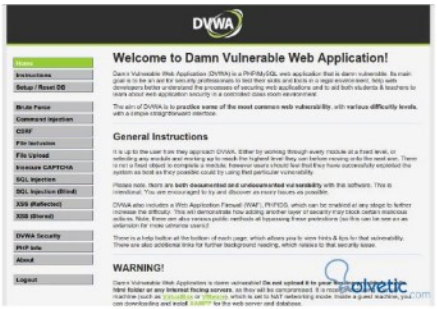
```
ciscoasa(config)# policy-map test
ciscoasa(config-pmap)# class test
ciscoasa(config-pmap-c)# inspect http http-map1

ciscoasa(config)# service-policy test interface outside
```

Note: If you modify an in-use HTTP inspection policy map, you must remove and reapply it for the changes to take effect.

```
ciscoasa(config)# policy-map test
ciscoasa(config-pmap)# class http
ciscoasa(config-pmap-c)# no inspect http http-map1
ciscoasa(config-pmap-c)# inspect http http-map1
```

# LAB 6.2: What To Do in the LAB?

**DAMN** Vulnerable Web Application







**Defending using ASA? How?**