PTI512 Keamanan Informasi EU003 7690

20190801296
Aryo Yudisthiro Wibowo
EU003 7690
SESI 7
Kelas EU003 7690

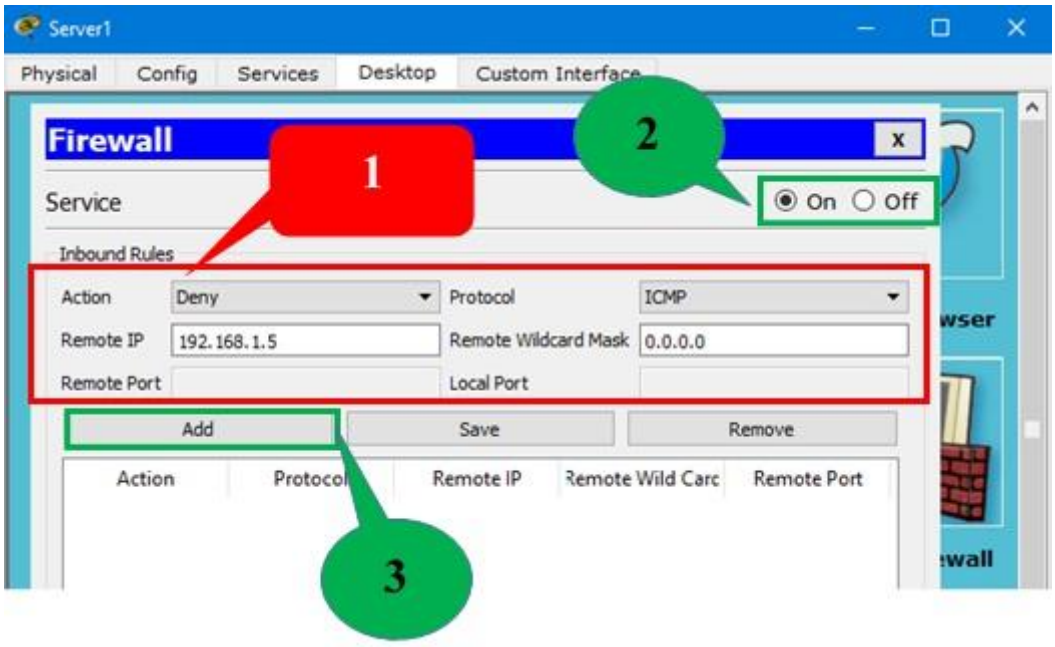
UNIVERSITAS ESA UNGGUL

2021

A. **Menentukan PC yang ingin di blokir**

Pada langkah ini, kita terlebih dahulu menentukan PC klien yang akan kita blokir. Kasus ini jika ada klien yang tidak diberi akses ke server. Langkah pertama yaitu dengan klik server kemudian klik dekstop setelah itu pilih firewall Ipv4 (karena IP yang sekarang digunakan masih menggunakan IP Versi 4). Setelah itu halaman kerja, untuk **Action** pilih "Deny/menolak" sedangkan pada **Protocol** pilih "ICMP". Langkah selanjutnya pada Remote IP, ketik IP yang ingin di blokir/menolak. Contoh PC2 dengan IP "192.168.1.5" ini bertujuan agar PC2 tidak bisa mengakses Server ataupun sebaliknya. Setelah itu,Langkah terakhir Remote Wildcard Mask, dalam menetukan PC mana yang ingin di blokir/Deny Wildcar Mask nya default 0.0.0.0.

Setelah point 3 – 5 telah di kerjakan, Klik ON pada pojok kanan atas dan dilanjutkan dengan Klik Add pada halaman kerja. Untuk lebih jelasnya akan di tunjukan pada gambar berikut ini:



Gambar Pengaturan PC2 yang diblokir

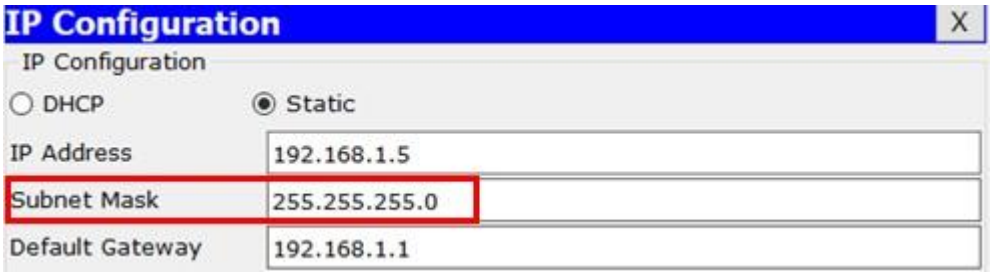
2. Menentukan PC yang bisa mengakses server

Pada pengaturan ini kita menentukan PC klien yang bisa akses ke server sedangkan yang lain tidak bisa akses. Kasus ini digunakan jika server hanya boleh di akses oleh IP tertentu dan pastinya hanya administrator jaringan yang bisa akses server tersebut. Langkah-langkahnya yaitu: Langkah pertama, pada halaman kerja, untuk Action pilih "Allow/mengijinkan" sedangkan pada Protocol pilih "ICMP", kemudian pada Remote

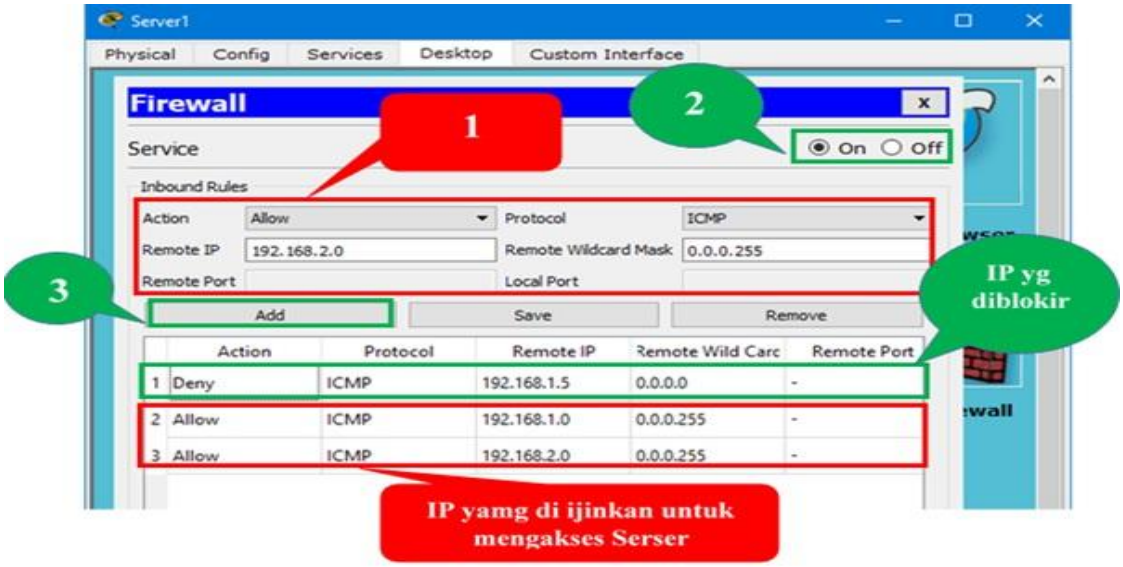
IP, ketik IP Gateway. Contoh IP Gateway yang ada sebelah Kiri (PC0, PC1, dan PC2) dengan IP Gateway “192.168.1.0” dan yang sebelah Kanan (PC3, PC4, dan PC5) dengan IP Gateway “192.168.2.0”. Ini bertujuan agar semua PC yang ada sebelah Kanan maupun yang sebelah Kiri bisa mengakses Server, kecuali IP yang telah di Blokir/Deny, seperti yang di bahas sebelumnya. Untuk Remote Wildcard Mask, dapat dengan menggunakan Rumus yaitu:

Wildcard mask=Subnet- subnet mask

Bagaimana cara menentukan subnet masknya? Untuk Subnet Mask di dapat dari IP Configuration yaitu : klik PC2 kemudian pilih Desktop dan klik pada IP Configuration setelah itu akan muncul gambar dibawah ini:



Jadi Wildcard mask untuk Subnet Mask pada IP di atas yaitu 255.255.255.255-255.255.255.0 maka Wildcard mask=0.0.0.255



Gambar Pengaturan PC yang bisa mengakses server

Sehingga jika PC di block dan dilakukan pengiriman paket maka akan menampilkan seperti pada gambar berikut ini:

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num
	Successful	PC1	Server1	ICMP		0.000	N	0
	Successful	PC4	Server1	ICMP		0.000	N	1
	Failed	PC2	Server1	ICMP		0.000	N	2

Gambar Status firewall pada server

3. Firewall pada router

Router digunakan sebagai penghubung antar perangkat dalam satu jaringan maupun di luar jaringan yang berbeda. Nah pada langkah ini router juga dapat digunakan sebagai firewall. Bagaimana caranya? Langkah-langkahnya yaitu klik router yang akan install firewall kemudian pada tampilan router pilih CLI (command line instruction) karena menggunakan CLI maka perintah yang di inputkan berupa command seperti cmd pada windows. Perintahnya dapat dilihat pada gambar berikut ini. Pada kasus ini komputer yang di blok yaitu IP 192.168.2.2 (IP milik PC2)

B.

```
<?php

/**
 * Check if the 'id' GET variable is set
 * Example - http://localhost/?id=1
 */
if (isset($_GET['id'])){
  $id = $_GET['id'];
  /**
   * Validate data before it enters the database. In this case, we need to check that
   * the value of the 'id' GET parameter is numeric
   */
  if ( is_numeric($id) == true){
   try{ // Check connection before executing the SQL query
     /**
      * Setup the connection to the database This is usually called a database handle (dbh)
      */
     $dbh = new PDO('mysql:host=localhost;dbname=sql_injection_example', 'dbuser', 'dbpasswd');

     /**
      * Use PDO::ERRMODE_EXCEPTION, to capture errors and write them to
      * a log file for later inspection instead of printing them to the screen.
      */
     $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

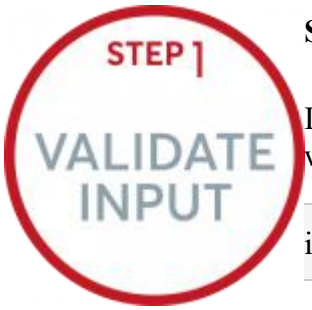
```
    /**
     * Before executing, prepare statements by binding parameters.
     * Bind validated user input (in this case, the value of $id) to the
     * SQL statement before sending it to the database server.
     *
     * This fixes the SQL injection vulnerability.
     */
    $q = "SELECT username
       FROM users
       WHERE id = :id";
    // Prepare the SQL query string.
    $sth = $dbh->prepare($q);
    // Bind parameters to statement variables.
    $sth->bindParam(':id', $id);
    // Execute statement.
    $sth->execute();
    // Set fetch mode to FETCH_ASSOC to return an array indexed by column name.
    $sth->setFetchMode(PDO::FETCH_ASSOC);
    // Fetch result.
    $result = $sth->fetchColumn();
    /**
     * HTML encode our result using htmlentities() to prevent stored XSS and print
the
     * result to the page
     */
    print( htmlentities($result) );
```

```
    //Close the connection to the database.
    $dbh = null;
  }
  catch(PDOException $e){
    /**
     * You can log PDO exceptions to PHP's system logger, using the
     * log engine of the operating system
     *
     * For more logging options visit http://php.net/manual/en/function.error-log.php
     */
    error_log('PDOException - ' . $e->getMessage(), 0);
    /**
     * Stop executing, return an Internal Server Error HTTP status code (500),
     * and display an error
     */
    http_response_code(500);
    die('Error establishing connection with database');
  }
 } else{
  /**
   * If the value of the 'id' GET parameter is not numeric, stop executing, return
   * a 'Bad request' HTTP status code (400), and display an error
   */
  http_response_code(400);
  die('Error processing bad or malformed request');
 }
}
```

C.

How to Prevent SQL Injection in PHP – Step By Step

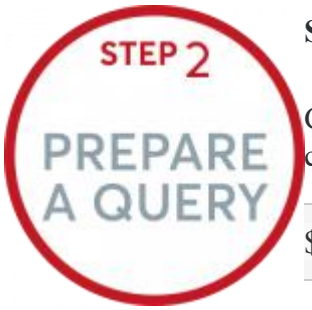
To prevent SQL Injection vulnerabilities in PHP, use PHP Data Objects (PDO) to create parametrized queries (prepared statements).



Step 1: Validate input

If possible, validate the data supplied by the user against a whitelist:

```
if (is_numeric($id) == true) { ... }
```



Step 2: Prepare a query

Create your query using parameter names preceded with colons as placeholders:

```
$q = "SELECT username FROM users WHERE id = :id";
```

Step 3: Create the prepared statement

Create the prepared SQL statement:

```
$sth = $dbh->prepare($q);
```

Step 4: Bind the parameters to the prepared statement

Bind your parameters to the query. For example, to bind the value of ID, use:

```
$sth->bindParam(':id', $id);
```

Step 5: Execute your query

After you pass the parameters, you may execute your query. For example:

```
$sth->execute();
```

Step 6: Fetch the result

After you execute the query, you may fetch its result to use further on. For example:

```
$result = $sth->fetchColumn();
```

Step 7: Validate your application

To make sure that your application is secure, use Acunetix and run a website vulnerability scan for your website.

D.

```
<?php

$config = array(

        'config'          =>
'F:/UTSKI/htdocs/UTSKI/openssl.cnf',

        'default_md'      => 'sha1024',
```

```
        'private_key_bits'      => '1024',
        'private_key_type'      => 'OPENSSL_KEYTYPE_RSA',
        'input_password'        => 'Aryoyw1603',
        'output_password'       => 'Aryoyw1603',
);

// key Pair
$keypair = openssl_pkey_new($config);

// Private Key
openssl_pkey_export($keypair, $privKey, null, $config);

//Public Key
$publickey = openssl_pkey_get_details($keypair);
$pubKey = $publickey['key'];

?>

<!DOCTYPE html>
<html>
<head>
    <meta chaset="utf-8">
    <title></title>
```

```
<head>
<body>
        <textarea cols="100" rows="30"><?php echo $privKey?></textarea>
        <textarea cols="100" rows="30"><?php echo $pubKey ?></textarea>
</body>
</html>
```