Secure you part of the deal: Security in Clouds and OWASP.

Agenda

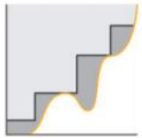
- Clouds
- Type of Clouds
- Model of Services
- Share Responsibility: Cloud
- Share Responsibility: Owasp
- Defense also is creative

Clouds

Cloud

Cloud computing is an on-demand service that provides virtual ITsevices.

Move from risk-laden up-front expense to flexible variable expense



Stop guessing at capacity planning

Remove complicated infrastructure management that adds little business value

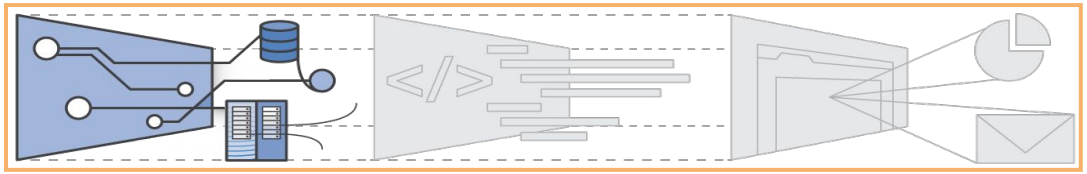
Go global in minutes

AVIRA

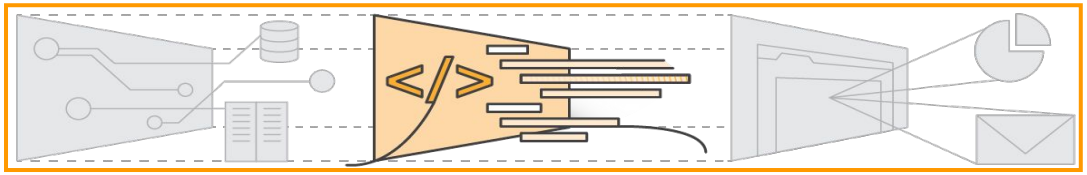
"We`ve reduced the time to predict and budget our services infrastructure by 80 percent when using AWS" – Cristian Toader – Avira Cloud Services Manager

Categories of cloud computing

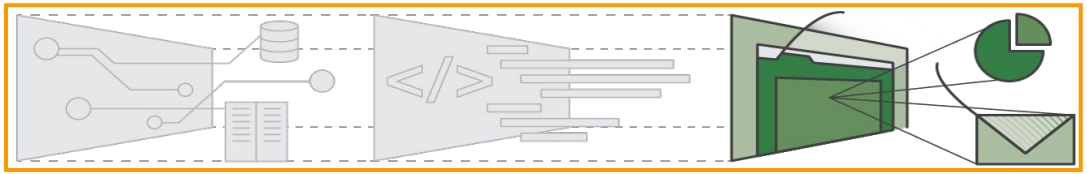
Infrastructure as a Service (IaaS):



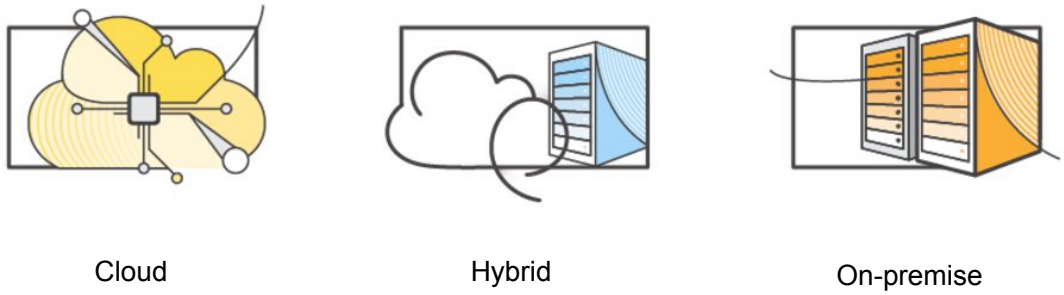
Platform as a Service (PaaS):

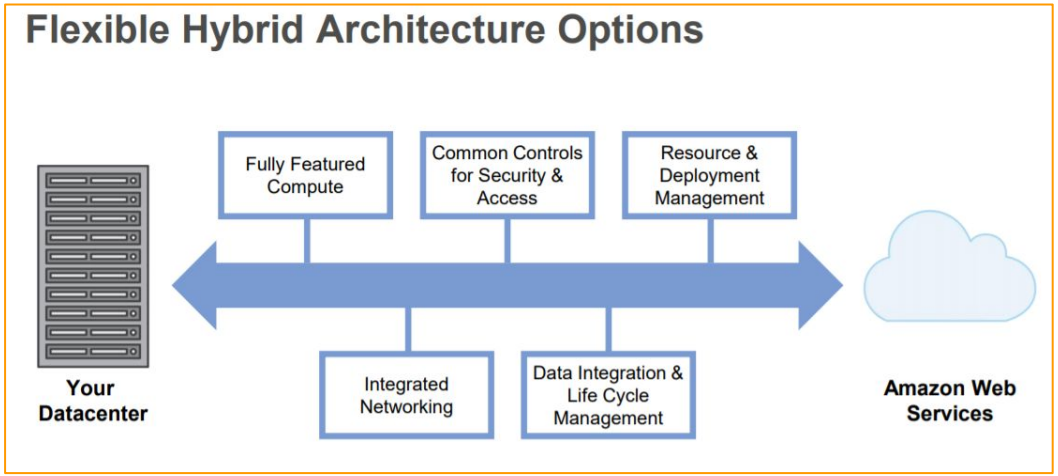


Software as a Service (SaaS):



Models of Services



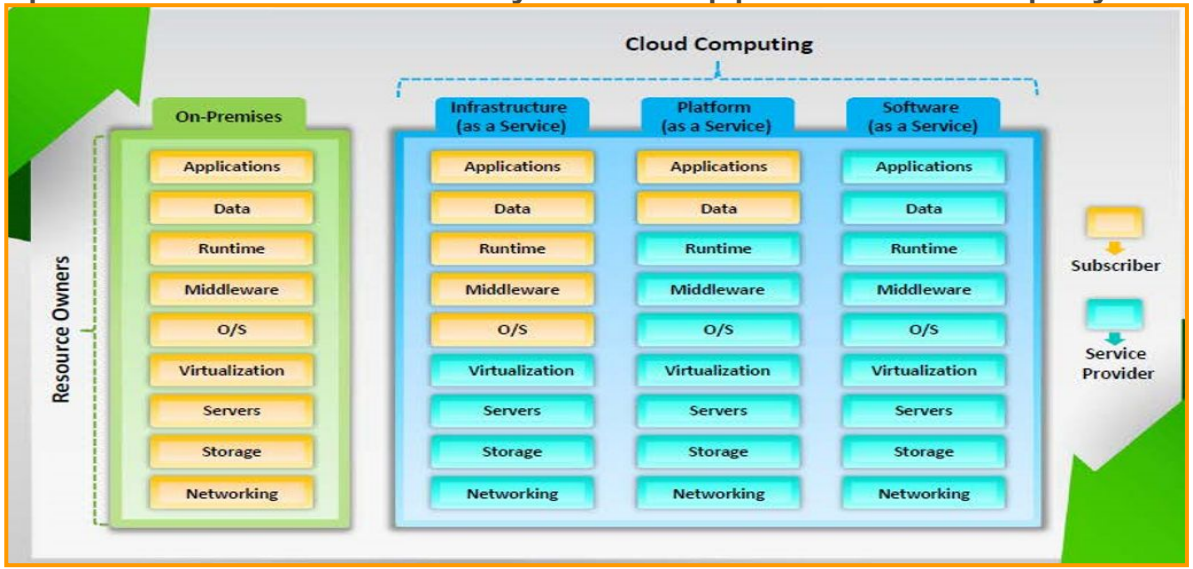


Share Responsibility: Cloud Security

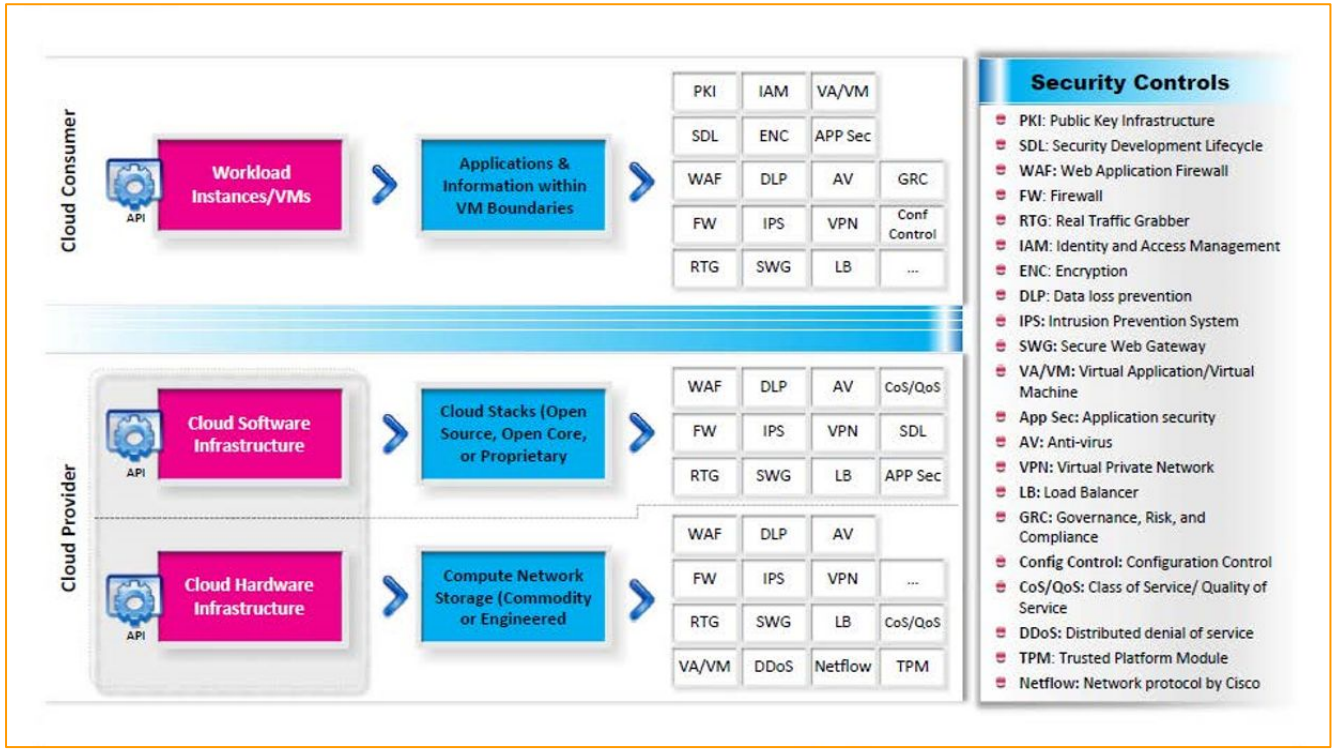
Share Responsibility

security model is based on a shared contract –responsibility:

- Is responsible for the security of the hardware and operating system
- The user is responsible for the security of the apps that are deployed in the cloud.



Clouds security Tools



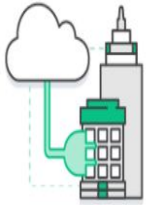
AWS migration tools

AWS Application Discovery Service

Show the data about configuration

AWS Snowball

Transfer petabyte-scale data

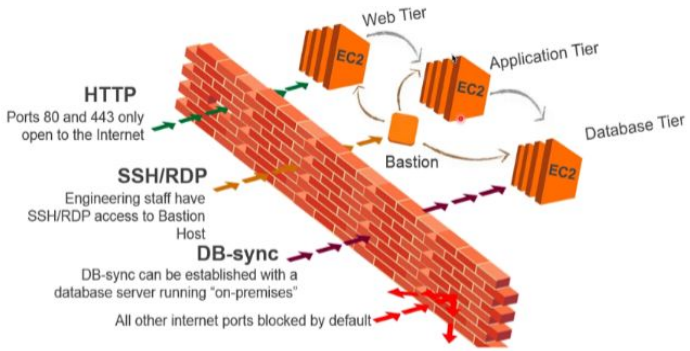


AWS Direct Connect

Make a direct connection between your network and an ability zone

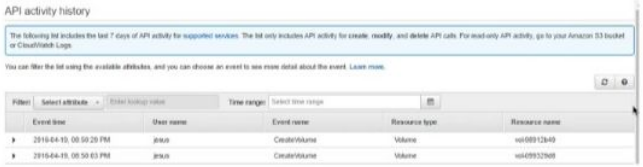
AWS security

Grupos de seguridad

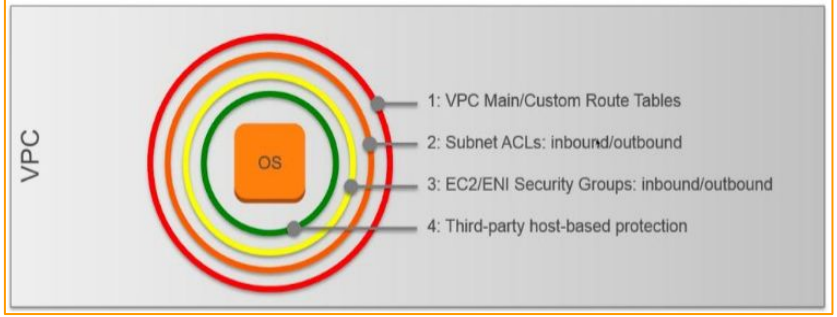


CloudTrail

- ✓ Guarda todas las llamadas de API realizadas en una cuenta
- ✓ Cada log lleva información de que llamada de API se realizó, el medio por el cual se realizó, el usuario, la IP desde la que se realizó, hora y fecha, etc



Seguridad en todos los niveles



Web Application Firewall

- ✓ Te protege de ataques web habituales como SQL injection, Shell scripting o DDoS
- ✓ Plantillas de reglas habituales
- ✓ Reglas específicas en base a tu aplicativo

Share Responsibility: Using Owasp

Security TOP 3 Threat

Data Breach/Loss

Data loss issues include:

- **Data is erased**, modified or decoupled (lost)
- **Encryption keys are lost**, misplaced or stolen
- **Illegal access to the data** in cloud due to Improper authentication, authorization, and access controls
- **Misuse of data** by CSP

Abuse of Cloud Services

Attackers **create anonymous access to cloud services** and perpetrate various attacks such as:

- **Password** and **key** cracking
- Building rainbow tables
- **CAPTCHA**-solving farms
- Launching **dynamic attack points**
- Hosting **exploits** on cloud platforms
- Hosting **malicious data**
- **Botnet** command or control
- **DDoS**

Insecure Interfaces and APIs

Insecure interfaces and APIs related risks:

- Circumvents **user defined polices**
- Is not credential leak proof
- Breach in **logging and monitoring facilities**
- Unknown API dependencies
- Reusable **passwords/tokens**
- Insufficient input-data validation

Security Spending

Table 9. Technology Spending and Effectiveness

Technology Options	Spending Rank	Spending	Big Win Rank	Big Wins	Effective Rank	Effective
Access and authentication	1	88.1%	1	30.6%	1	45.5%
Advanced malware prevention (IPS/UTM, other)	2	80.2%	2	28.9%	3	42.1%
SIEM	11	57.9%	3	25.6%	14T	26.4%
Vulnerability management	8	64.3%	4	24.8%	9	31.4%
Continuous monitoring	5	69.0%	5	24.0%	6T	36.4%
Network traffic visibility (monitoring, decryptors, etc.)	7	66.7%	6	22.3%	7	35.5%
Data protection (DLP)/Encryption	4T	69.8%	7T	20.7%	8T	33.1%
Analytics (including visualization)	9T	59.5%	7T	20.7%	15T	24.0%
Incident response tools	12	54.0%	8T	18.2%	6T	36.4%
Log management	6	67.5%	8T	16.5%	5	38.0%
Mobile device management	10	58.7%	9	16.5%	10	30.6%
Security device management	13T	53.2%	10	15.7%	12	28.9%
Wireless security	4T	69.8%	11T	14.9%	4	41.3%
Cyberthreat intelligence services	15	47.6%	11T	14.9%	15T	24.0%
Endpoint security (other than BYOD protections)	3	74.6%	12	14.0%	2	43.8%
Application security—secure development	14T	51.6%	13T	11.6%	11	29.8%
DDoS protection	13T	53.2%	13T	11.6%	14T	26.4%
BYOD security (MDM/NAC, etc.)	9T	59.5%	14	10.7%	8T	33.1%
Application security (life-cycle management or monitoring)	14T	51.6%	15	9.1%	13T	27.3%
Security intelligence platform	16	35.7%	16	7.4%	13T	27.3%
Embedded device security or monitoring (IoT)	17	27.8%	17	4.1%	16	19.0%

Top ten security skills required

Implementing Secure Coding

Key references

- Stopping XSS in your web application: OWASP
  - XSS (Cross Site Scripting) Prevention Cheat Sheet
- General information about injection:
  - Top 10 2013-A1-Injection

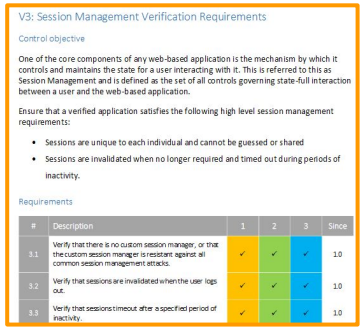
Key tools

- OWASP Java Encoder Project
- Microsoft .NET AntiXSS Library
- OWASP ESAPI
- OWASP Encoder Comparison Reference Project

Response Headers

- HTTP Strict Transport Security (HSTS)
- Public Key Pinning Extension for HTTP (HPKP)
- X-Frame-Options
- X-XSS-Protection
- X-Content-Type-Options
- Content-Security-Policy
- X-Permitted-Cross-Domain-Policies

Training → Design → Develop-ment → Validation → Maintenance



V3: Session Management Verification Requirements

Control objective

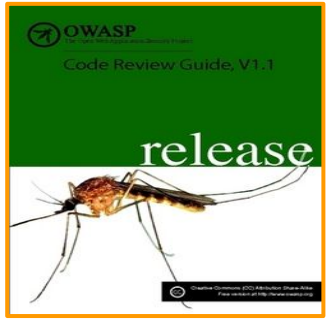
One of the core components of any web-based application is the mechanism by which it controls and maintains the state for a user interacting with it. This is referred to this as Session Management and is defined as the set of all controls governing state-full interaction between a user and the web-based application.

Ensure that a verified application satisfies the following high level session management requirements:

- Sessions are unique to each individual and cannot be guessed or shared
- Sessions are invalidated when no longer required and timed out during periods of inactivity.

Requirements

#	Description	1	2	3	Since
3.1	Verify that there is no custom session manager, or that the custom session manager is resistant against all common session management attacks.	✓	✓	✓	1.0
3.2	Verify that sessions are invalidated when the user logs out.	✓	✓	✓	1.0
3.3	Verify that sessions timeout after a specified period of inactivity.	✓	✓	✓	1.0



Checklist for Secure Token

Requisitos

#	Descripción	1	2	3	Desde
3.1	Verificar que no se utiliza un gestor de sesiones personalizado, o que, si el gestor de sesiones es personalizado, éste sea resistente contra los ataques más comunes.	✓	✓	✓	1.0
3.2	Verificar que las sesiones se invalidan cuando el usuario cierra la sesión.	✓	✓	✓	1.0
3.3	Verificar que las sesiones se invalidan luego de un período determinado de inactividad.	✓	✓	✓	1.0
3.4	Verificar que las sesiones se invalidan luego de un período determinado de tiempo, independientemente de que se esté registrando actividad (timeout absoluto).			✓	1.0
3.5	Verificar que todas las páginas que requieren autenticación poseen acceso fácil y visible a la funcionalidad de cierre de sesión.	✓	✓	✓	1.0
3.6	Verificar que el identificador de sesión nunca se revele en URLs, mensajes de error o registros de bitácora. Esto incluye verificar que la aplicación no es compatible con la re-escritura de URL incluyendo el identificador de sesión.	✓	✓	✓	1.0
3.7	Verificar que toda autenticación exitosa y re-autenticaciones generen un nuevo identificador de sesión.	✓	✓	✓	1.0
3.10	Verificar que sólo los identificadores de sesión generados por la aplicación son reconocidos como activos por ésta.		✓	✓	1.0
3.11	Verificar que los identificadores de sesión son suficientemente largos, aleatorios y únicos para las sesiones activas.	✓	✓	✓	1.0
3.12	Verificar que los identificadores de sesión almacenados en cookies poseen su atributo “path” establecido en un valor adecuadamente restrictivo y que además contenga los atributos "Secure" y "HttpOnly"	✓	✓	✓	3.0
3.16	Verificar que la aplicación limita el número de sesiones concurrentes activas.	✓	✓	✓	3.0
3.17	Verificar que una lista de sesiones activas esté disponible en el perfil de cuenta o similar para cada usuario. El usuario debe ser capaz de terminar cualquier sesión activa.	✓	✓	✓	3.0
3.18	Verificar que al usuario se le sugiera la opción de terminar todas las otras sesiones activas después de un proceso de cambio de contraseña exitoso.	✓	✓	✓	3.0

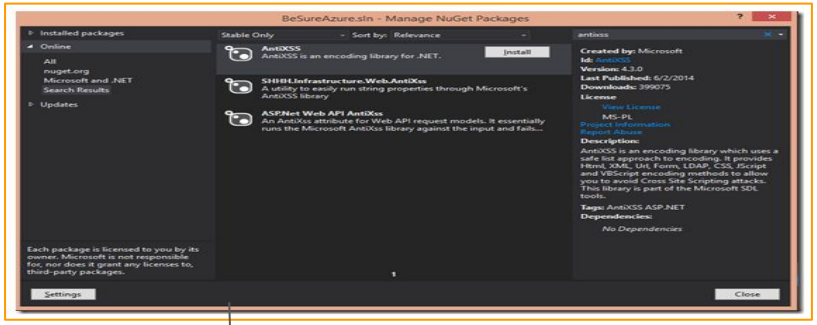
The defense also is creative

Methodology

Ciclo de vida de MS Security Development (MS SDL): Uno de los primeros de su tipo, MS SDL fue propuesto por Microsoft de acuerdo a las fases de un SDLC

1. TRAINING	. REQUIREMENTS	3. DESIGN	4. IMPLEMENTATION	5. VERIFICATION	6. RELEASE	7. RESPONSE
1. Core Security Training	2. Establish Security Requirements	5. Establish Design Requirements	8. Use Approved Tools	11. Perform Dynamic Analysis	14. Create an Incident Response Plan	Execute Incident Response Plan
	3. Create Quality Gates/Bug Bars	6. Perform Attack Surface Analysis/ Reduction	9. Deprecate Unsafe Functions	12. Perform Fuzz Testing	15. Conduct Final Security Review	
	4. Perform Security and Privacy Risk Assessments	7. Use Threat Modeling	10. Perform Static Analysis	13. Conduct Attack Surface Review	16. Certify Release and Archive	

Tools in the IDE



Anti xss.Net

```
private static readonly Tuple<string, string>[] WhiteList = (new string[]
    {
        "<b>", "</b>", "<i>", "</i>"
    })
    .Select(tag => Tuple.Create(AntiXss.Encoder.HtmlEncode(tag), tag))
    .ToArray();

public static string Sanitize(string html)
{
    var safeHtml = new StringBuilder(AntiXss.Encoder.HtmlEncode(html));

    for (int index = 0; index < WhiteList.Length; index++)
    {
        string encodedTag = WhiteList[index].Item1;
        string decodedTag = WhiteList[index].Item2;
        safeHtml.Replace(encodedTag,decodedTag);
    }

    return safeHtml.ToString();
}
```

```
Server.HtmlEncode(string)
```

Using the HTML encoder example for a form:

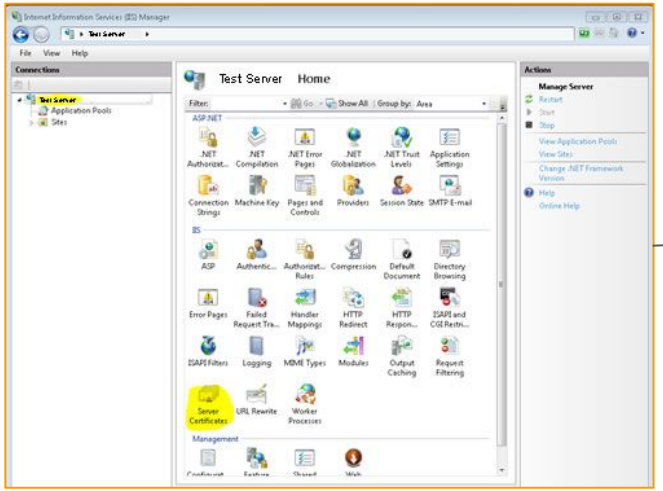
Text Box: <%@ Page Language="C#" ValidateRequest="false" %>

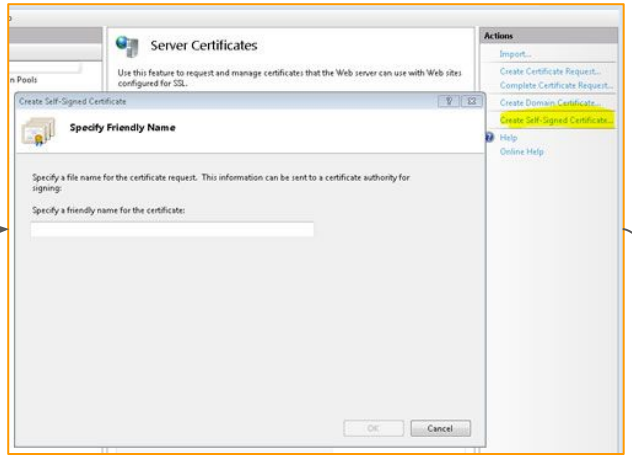
```
<script runat="server">
void searchBtn _Click(object sender, EventArgs e) {
Response.Write(HttpUtility.HtmlEncode(inputTxt.Text)); }
</script>
<html>
<body>
<form id="form1" runat="server">
<asp:TextBox ID="inputTxt" Runat="server" TextMode="MultiLine" Width="382px" Height="152px">
</asp:TextBox>
<asp:Button ID="searchBtn" Runat="server" Text="Submit" OnClick=" searchBtn _Click" />
</form>
</body>
</html>
```

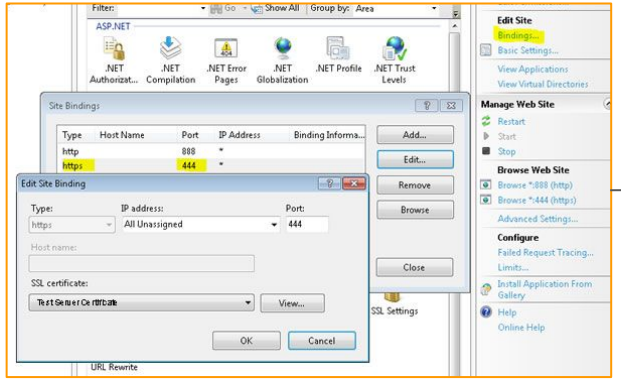
```
<pages validateRequest="true" ... />
```

```
<%@ Page validateRequest="false" %>
```

Self Sing Certificate







Contact me!

Email Address: Ing.Arreaza@gmail.com
WebSite:www.seguridadaplicativos.com

Securing your code you also
Secure your Clouds

T H A N K Y O U !