CRSP - OWASP

Motivación

Arquitectura de despliegue

Mod_Security

Conclusiones

CRSP - OWASP

Introducción

- Definiciones IT != TI
- Actores
  - Entes reguladores
  - Clientes
  - Proveedores
  - Población General

CRSP – OWASP

GRAFO DE INTERRELACIONES DE ACTORES

- Publicación de Activos
  - Trade off seguridad vs. disponibilidad
  - Expansión de la superficie de ataque

CRSP - OWASP

¿paranoia? (themokneygroup.info)

- Falla en config routers de borde de ANTEL
- Reportado en 2011(varios Certs)
- 2013 aún activo ¿que falló?
- Objetivos
  - ADSL gratis (free as in free beer)
  - Estudio de distribución de claves en el mercado local
- Crawling
- +50 SSID,PSK/hr
- +50 USR,PWD/hr

CRSP - OWASP

TIPOS DE AMENAZAS

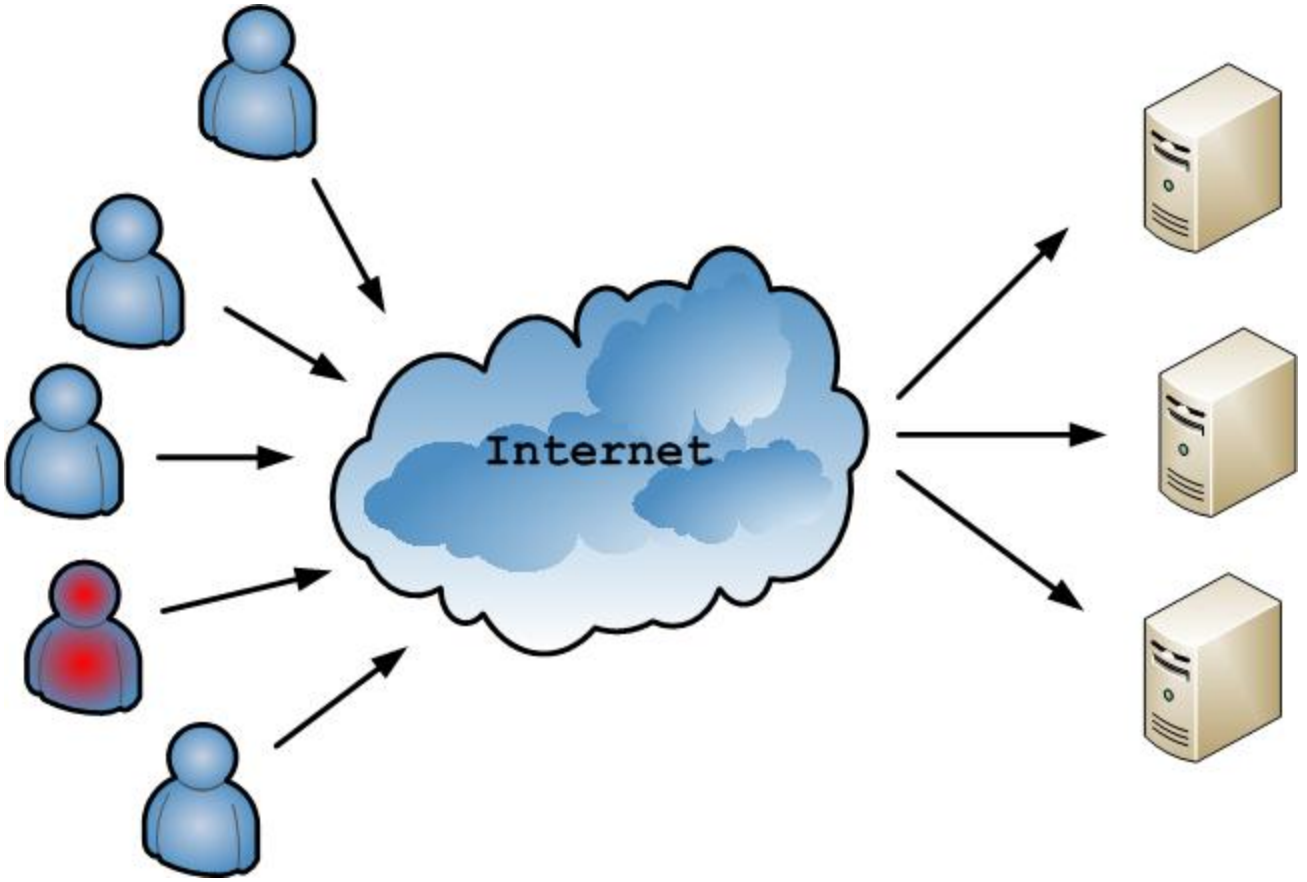
- DoS
- Brute Force
- SQL Injection
- XSS
- Web Crawler

NO CONSIDERAMOS

- DDoS
- Buffer Overrun
- Escalada de privilegios
- DNS Poison
- Defacement
- Missinformation

CRSP - OWASP

ARQUITECTURAS DE DESPLIEGUE 1/4



CRSP - OWASP

ARQUITECTURAS DE DESPLIEGUE ¼

Pro:

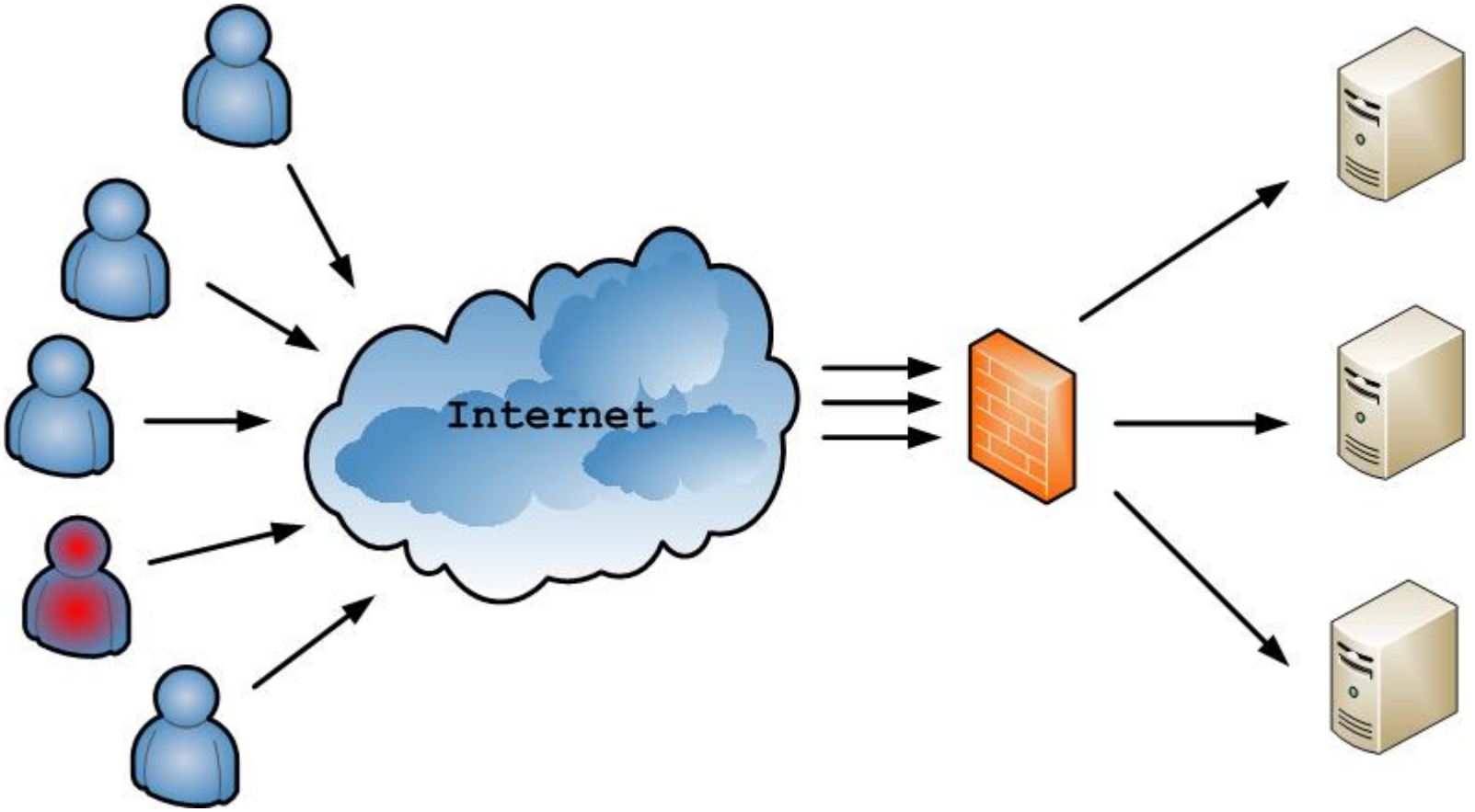
- Cost Effective

Contra:

- ¿?

CRSP - OWASP

ARQUITECTURAS DE DESPLIEGUE 2/4



CRSP - OWASP

ARQUITECTURAS DE DESPLIEGUE 2/4

Pro:

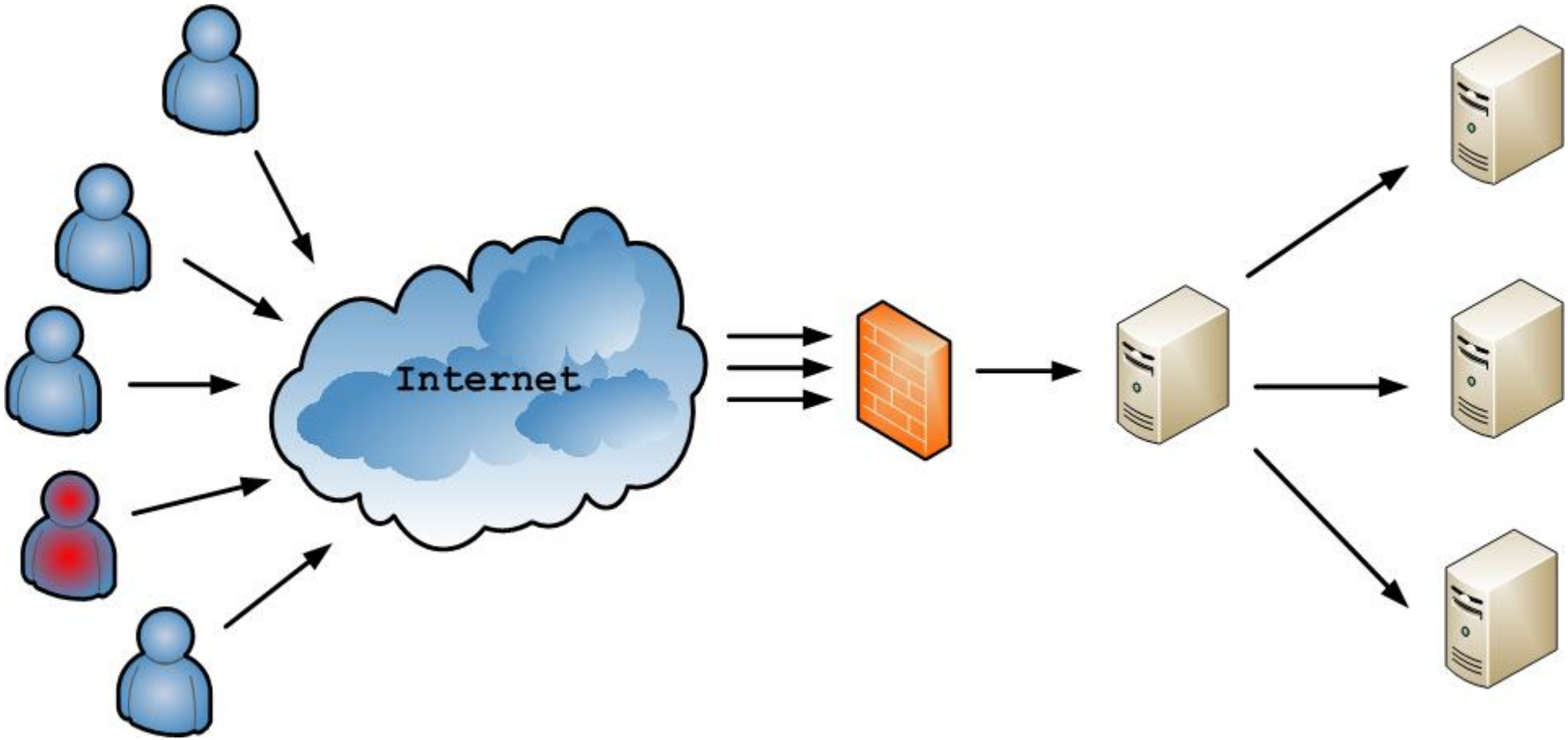
- Cost Effective

Contra:

- Administración engorrosa
- Despliegue de nuevos servicios multiplexando puertos
- Sin control en el flujo de información
- Varios puntos de control

CRSP - OWASP

ARQUITECTURAS DE DESPLIEGUE 3/4



CRSP - OWASP

ARQUITECTURAS DE DESPLIEGUE 3/4

Pro:

- Cost Effective
- Rapido despliegue de nuevos servicios

Contra:

- No permite estratificar niveles de seguridad
- Recordar que hay que securitizar el acceso por IP
- Recordar cambiar la firma de los servidores

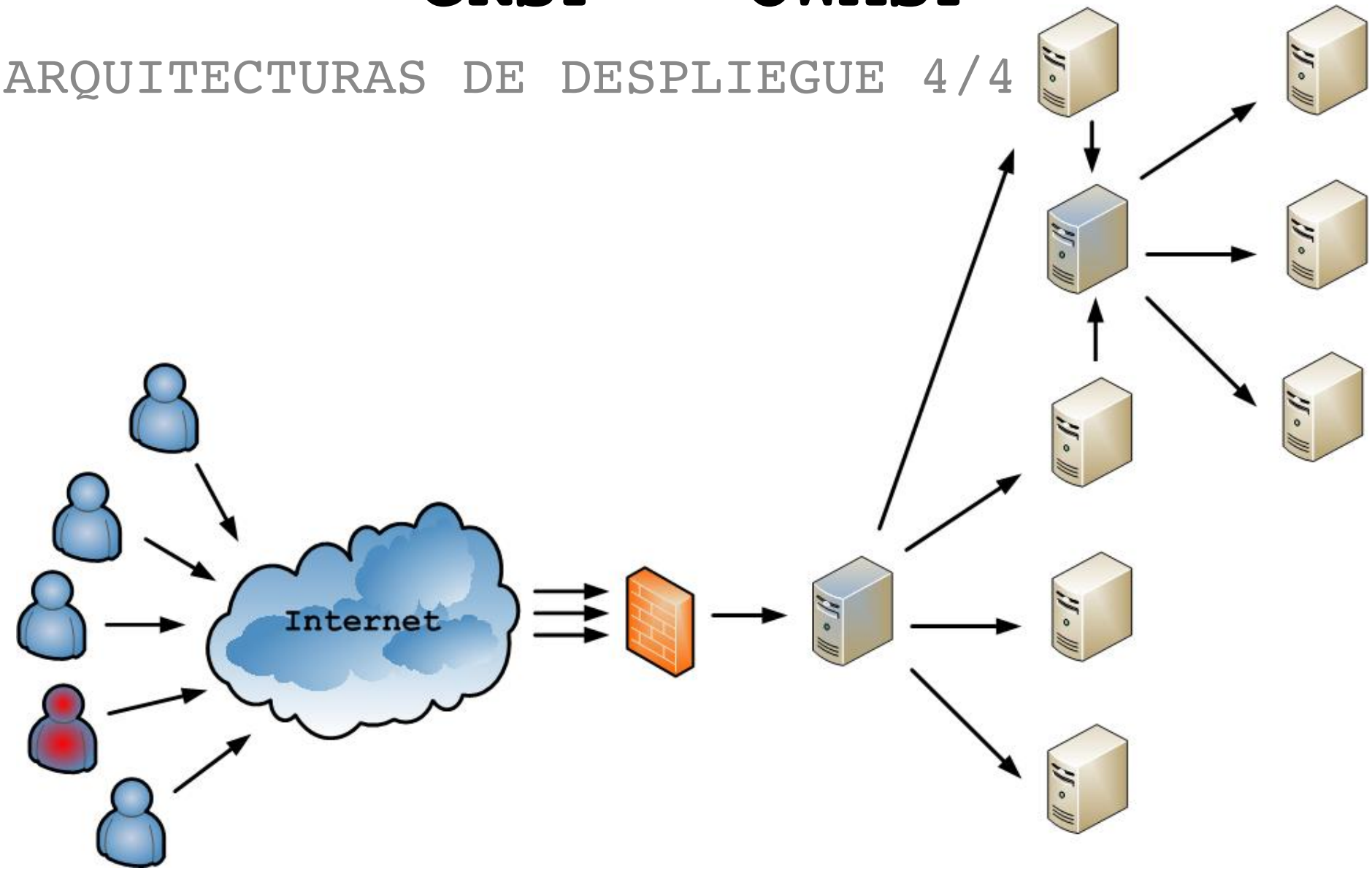
CRSP – OWASP

Scan report for www.agesic.gub.uy (190.64.2.190)
PORT STATE SERVICE VERSION
80/tcp open http Apache httpd 2.2.15 ((CentOS))
| http-methods: GET HEAD POST TRACE OPTIONS
| Potentially risky methods: TRACE

Scan report for www.minterior.gub.uy (190.64.6.131)
PORT STATE SERVICE VERSION
80/tcp open http?
|_http-generator: Joomla! - Open Source Content Management
|_http-methods: No Allow or Public header in OPTIONS response (status code 302)

CRSP - OWASP

ARQUITECTURAS DE DESPLIEGUE 4/4



CRSP - OWASP

ARQUITECTURAS DE DESPLIEGUE 4/4

Pro:

- Cost Effective

Contra:

- Comienzan a surgir costos en la administración de las reglas de los proxies reversos y/o reglas de firewall

CRSP - OWASP

¿COMO PROTEGER LAS APLICACIONES?

- CLIENTES REQUIRIENDO DESARROLLOS CON CONTROLES ESPECIFICOS
- EMPRESAS DISPUESTAS A INVERTIR EN DESARROLLO DE APLICACIONES ‘SEGURAS’
- INFRA ($)
- MOD_SECURITY

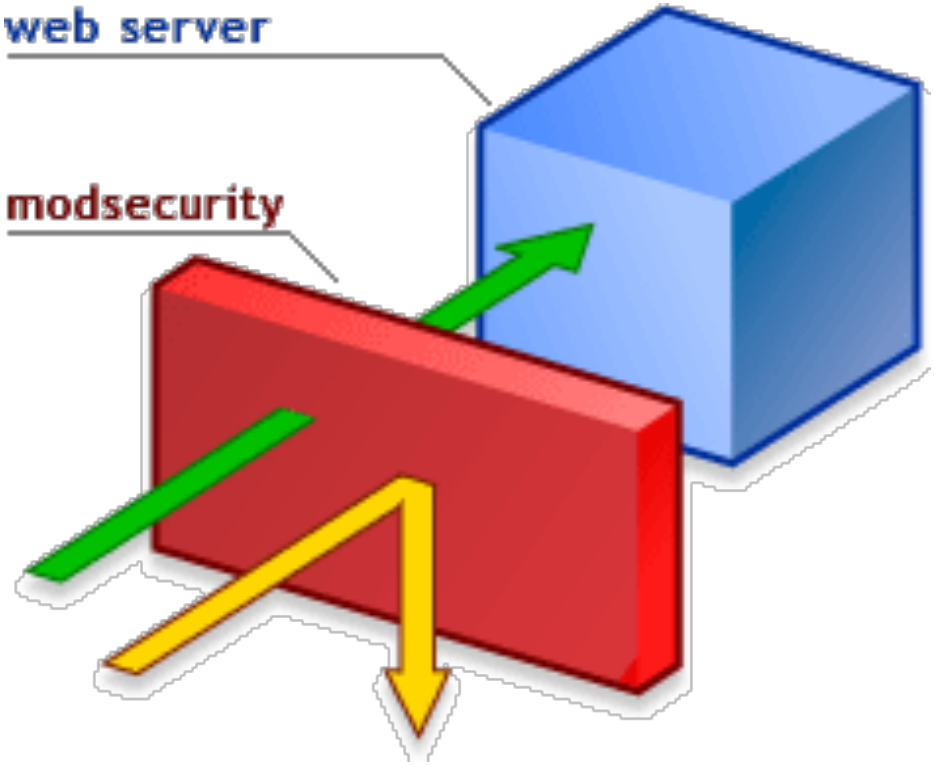
CRSP - OWASP

MOD_SECURITY

- Modulo disponible para:
  - IIS
  - Apache
  - Nginx
- http://modsecurity.org/
- Rulesets
  - ModSecurity Commercial Rules
  - OWASP Core Rule Set Project (CRSP)

CRSP - OWASP

ModSecurity – Arquitectura:



CRSP - OWASP

REQUEST :=

GET /images/NuevosBotones/CONCURSOyLLAMADOS.jpg HTTP/1.1
Host: www.minterior.gub.uy
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.minterior.gub.uy/
Cookie: 61d46af26724acba64f057ba47666ff7=p19d7nsch1uvheg4gapedatdn3
Connection: keep-alive

ACCION := {ACCEPT,DENY,LOGACCEPT,LOGDENY}

FILTRO := IF(METHOD==GET && REFERER==https://www.minterior.gub.uy) => LOGACCEPT

CRSP - OWASP

ModSecurity

Ejemplos de reglas CRSP de OWASP

- base rules
  - SQL Injection
  - crawling
  - XSS
- optional rules
  - Session hijacking
- experimental rules
  - DoS
  - Brute Force

CRSP – OWASP

PERSONALIANDO REGLAS 1/2

- No todos las firmas son aplicables a todas las realidades
- Algunas firmas deben ser modificadas para aplicarlas a los activos específicos.

Ejemplo:

Protección de BruteForce mediante URL:

- **SecAction "id:'900014', phase:1, …**
- **setvar:'tx.brute_force_protected_urls=(login.php login.jsp …)',**
- **setvar:'tx.brute_force_burst_time_slice=60',**
  setvar:'tx.brute_force_counter_threshold=10',
  setvar:'tx.brute_force_block_timeout=600', nolog, pass"
- **…**
- **SecRule TX:FILENAME !@within %{tx.brute_force_protected_urls} pass**

CRSP - OWASP

PERSONALIZANDO REGLAS 2/2

Ejemplo (cont)

- La URL https://correo.agesic.gub.uy/zimbra/home/**usuario@dominio**/Contacts?fmt=cf&t=2&all
- Retorna HTTP 401 (MUST AUTHENTICATE)
- La variable **brute_force_protected_url** no permite almacenar URLS compuestas de expresiones regulares.
- Lo anterior sucede porque el operador **@within** no inspecciona regexp pero si el operador **@rx**

Solución:

- Modificar la regla de control por
- **SecRule REQUEST_FILENAME "!@rx (PATTERN1|...|PATTERNN) ...**

CRSP - OWASP

RESULTADOS OBSERVADOS:

- Multiples intentos de fuerza bruta (desde el medio local)
- Multiples intentos de negación de servicios (desde varios origenes)
- WEB-CRAWLERS

CRSP – OWASP

RESULTADOS OBSERVADOS:

- Cuidado con los falsos positivos

```
GET /index.php?option=com_content&view=article&id=1434 HTTP/1.0
Host: www.minterior.gub.uy
Accept: text/html,text/plain,application/*
From:
User-Agent: agesic-crawler (Enterprise; T3-RG6J9B6XL6A3K;
fabricio.alvarez@agesic.gub.uy,sestevez@hg.com.uy,sperez@hg.com.uy,gbartolomeo@at.com.uy,xavier.verdino@pyxisportal.com,alertasnoc@hg.com.uy)
Accept-Encoding: gzip
If-Modified-Since: Sat, 12 Oct 2013 04:43:20 GMT
```

- ¿Como corroborar la veracidad del request anterior?

CRSP - OWASP

Conclusiones (1/2)

- Las estructuras formales no siempre brindan respuesta a sus clientes
- Si bien existe un modelo formal al tratamiento de incidentes de seguridad, no están claras las responsabilidades de los actores.
  - ¿Quien debe 'garantizar' seguridad?
  - ¿Quien y como 'exige' las garantías minimas?
  - ¿Quien y como responde a dichas exigencias?
- No siempre es posible lograr motivar la dirección sobre el valor que agrega la disciplina de la seguridad informática
  - ¿Ud. cerrraria este proyecto?
- Aprendizaje de nuevas herramientas y distribución del conocimiento
- Detectamos que los ataques sobre la infraestructura son reales y ¿deberian? ser considerados por los 'custodios'. Para esto es necesario **responsabilidades.**

CRSP - OWASP

Conclusiones (2/2)

- URL encryption
- Ofuscar las aplicaciones no es necesariamente malo
- Captchas (siguen siendo una alternativa económica)
- Personalizar paginas de error
- Usar TLS por defecto
- Proteger la última milla como método defensivo
- Bloquear no siempre es suficiente, un ‘tiron de orejas’ a tiempo puede prevenir dolores mayores
- Integración con ISP para obtener información de los atacantes

CRSP - OWASP

gracias por vuestra atención y tiempo