Tecnologías Emergentes y Tendencias de Seguridad

Una aproximación desde el punto de vista de un atacante

Iván Arce - CTO

Core Security Technologies
Humboldt 1967 2do Piso
Buenos Aires, Argentina
Teléfono: (+54-11) 5556-2673
Email: ivan.arce </at>coresecurity.com

AGENDA

- **Prólogo, auto-bombo y otras excusas para hablar del tema**
- **Como modelar tecnología & seguridad ?**
- **Aplicación del modelo a 6 tecnologías específicas**
  - Seguridad a nivel de aplicaciones, web 2.0 , verticales;
  - Tecnología de virtualización
  - Telefonía IP
  - Infra-estructura de red.
  - Dispositivos móviles
  - SaaS – Cloud computing
- ***Penetration Testing* y la visión del atacante**
- **Futurología: Desafíos y tendencias**

.prolog

Y este que hace aca? De donde salió?!

- **CTO y co-fundador de Core Security Technologies**
  http://www.coresecurity.com
  - Empresa de software y servicios de seguridad informática fundada en Argentina en 1996
- **ex-Director del equipo de consultoría: CORE Security Consulting Services (SCS)**
  http://www.coresecurity.com/content/services-overview-core-security-consulting-services
  - Focalizado en servicios de Penetration Testing y auditoría de seguridad de software
- **Lidero la creación del equipo de CORE IMPACT**
  http://www.coresecurity.com/content/core-impact-overview
  - El primer software comercial de penetration testing a nivel mundial
  - Lanzamiento de v1.0 en Abril del 2002
  - Lanzamiento de v8.0 en Dic. 2008
  - Más de 800 clientes en 40+ paises alreadedor del mundo.
- **Editor asociado de IEEE Security & Privacy magazine**
  http://www.computer.org/portal/security
  - Co-editor de la sección New Vulnerabilities and Attack Trends (2003-2007)

Cúal es el propósito de esta charla?

LA CONSTRUCCION DE UN LENGUAJE COMUN

- **Presentar un marco de referencia para hablar de:**
  - Tecnologías emergentes
  - Tendencias de la industria y mercado
  - Tendencias y visibilidad de ataques
  - Estrategias para manejar riesgo y priorizar inversión en seguridad

- **Describir nuestras experiencias abordando el problema desde el punto de vista del atacante**

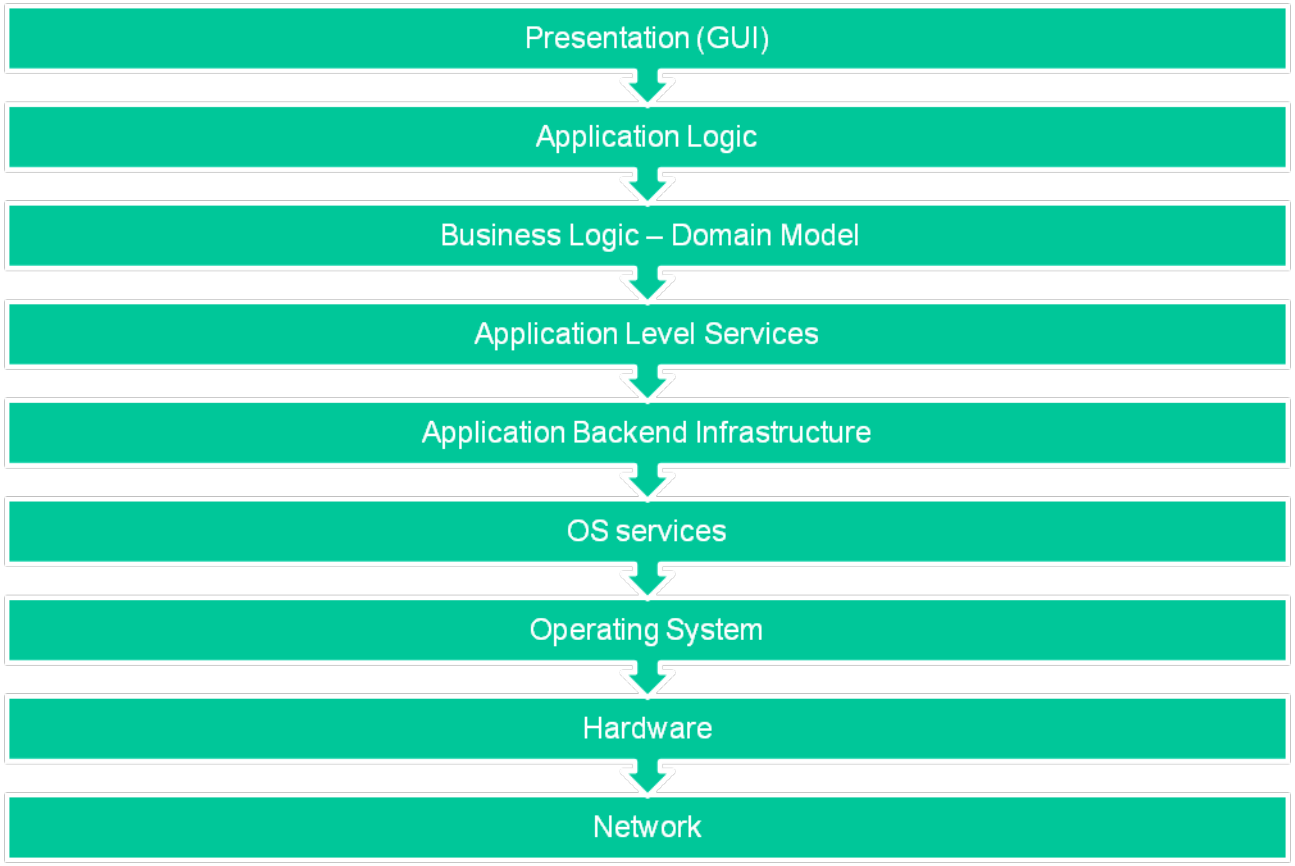
- **Motivar la discusión y reflexión sobre la naturaleza de nuestra disciplina**

- **Describir algunos desafíos técnicos y científicos venideros**

COMO MODELAR TECNOLOGIA PARA EMPRESAS?

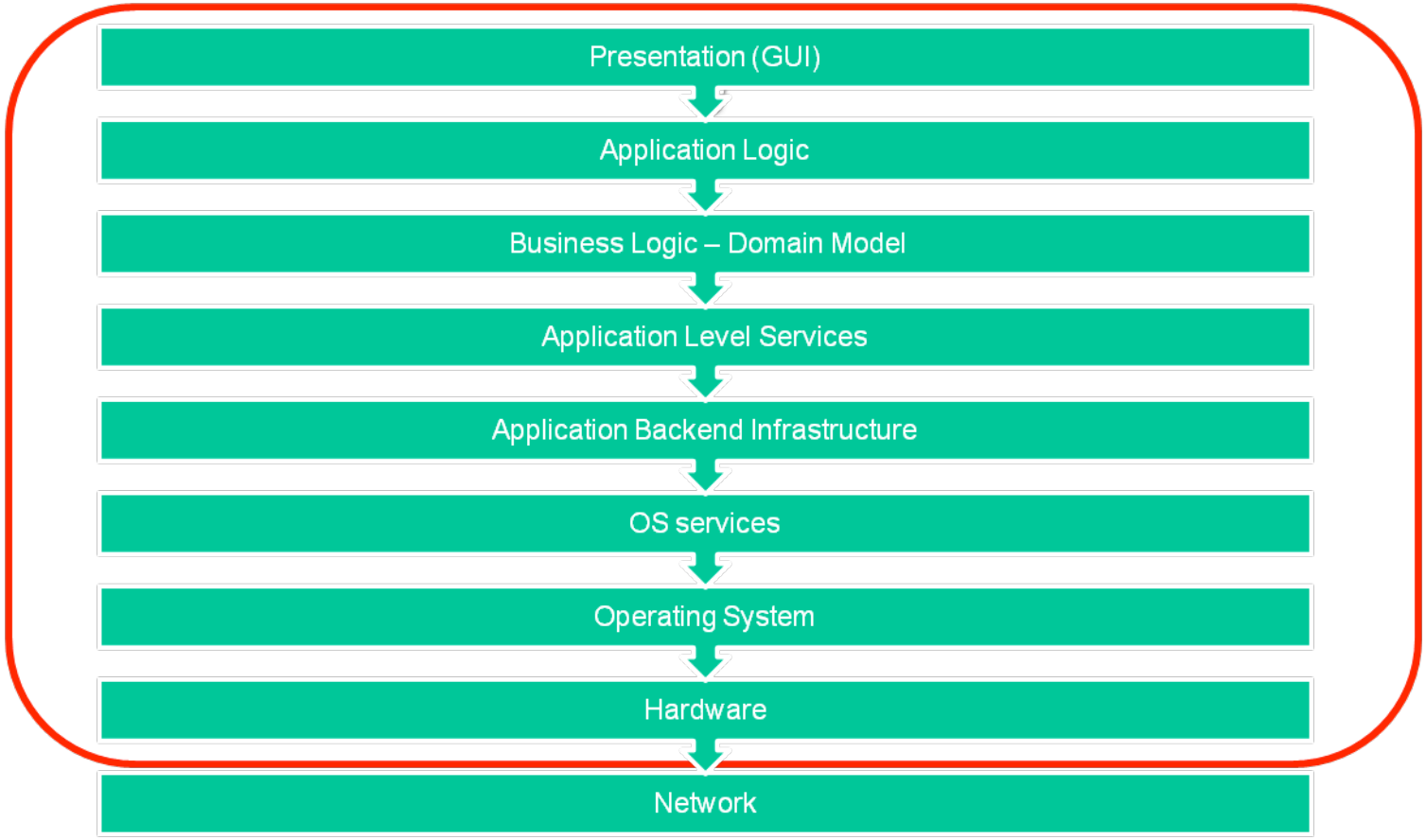
Modelo pseudo-OSI amigable para gente de negocios

Visión de tecnología en capas



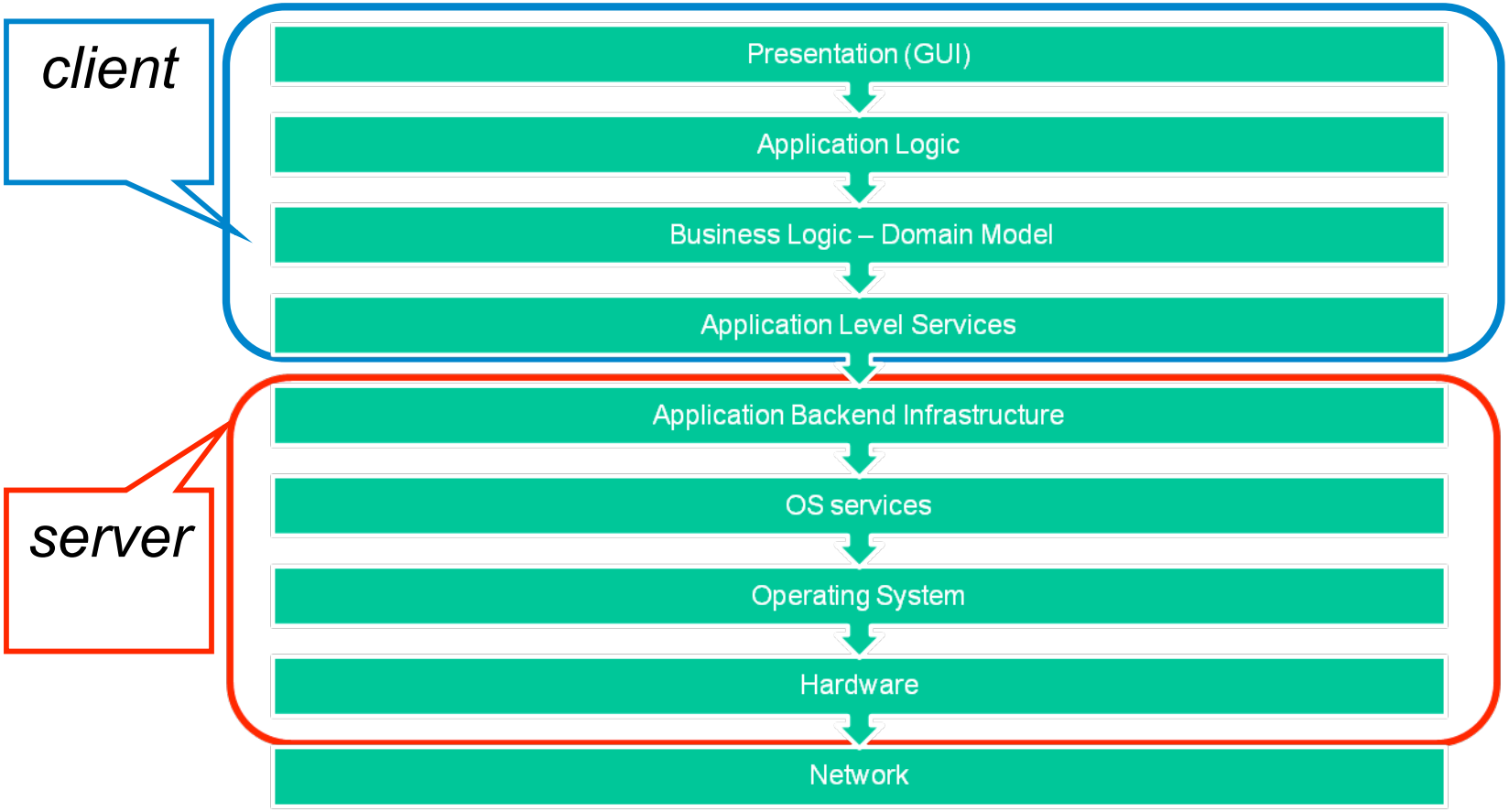
1960-1980: El paradigma Mainframe

El stack completo vive dentro de la maquina



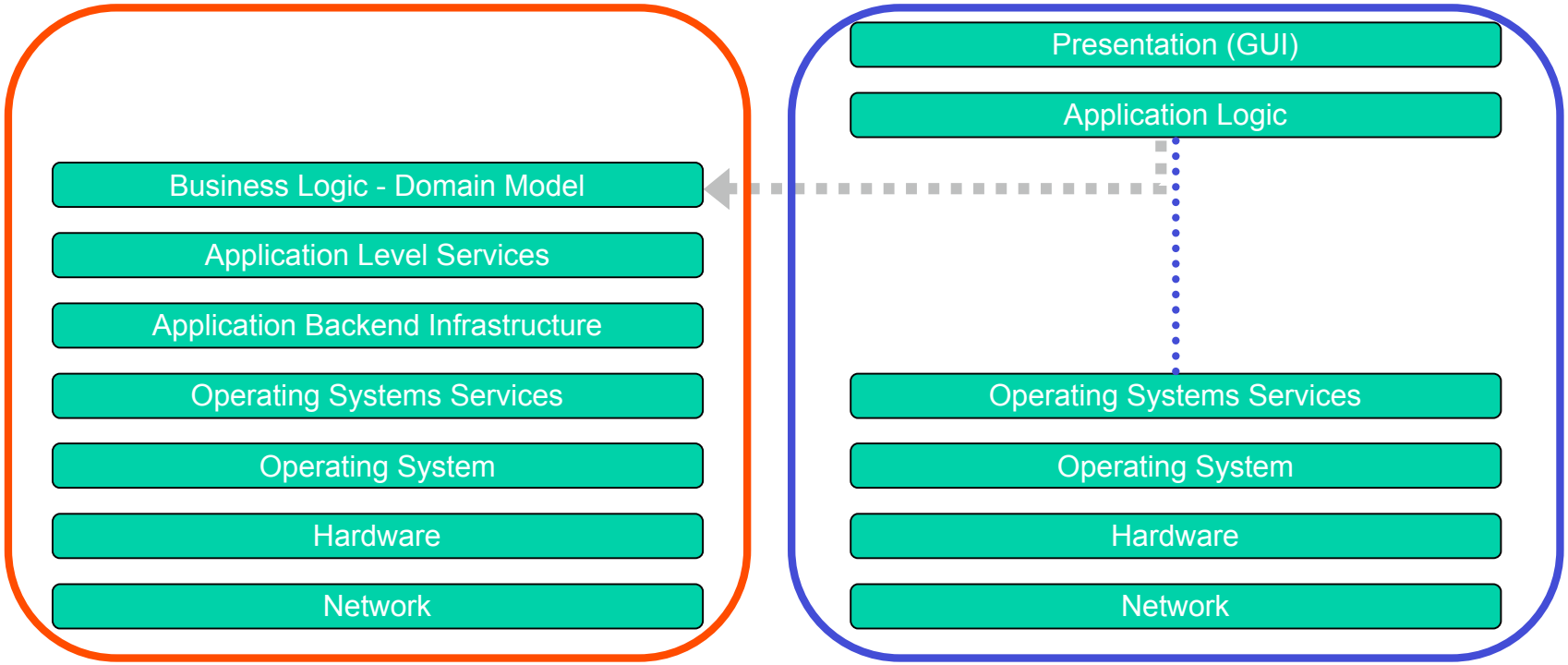
1980-1995: El paradigma Cliente-Servidor

Separemos las aplicaciones entre dos (o más) actores



1980-1995: Arquitectura cliente-servidor con 2 *tiers*

Esta es la arquitectura más común para la mayoría de empresas

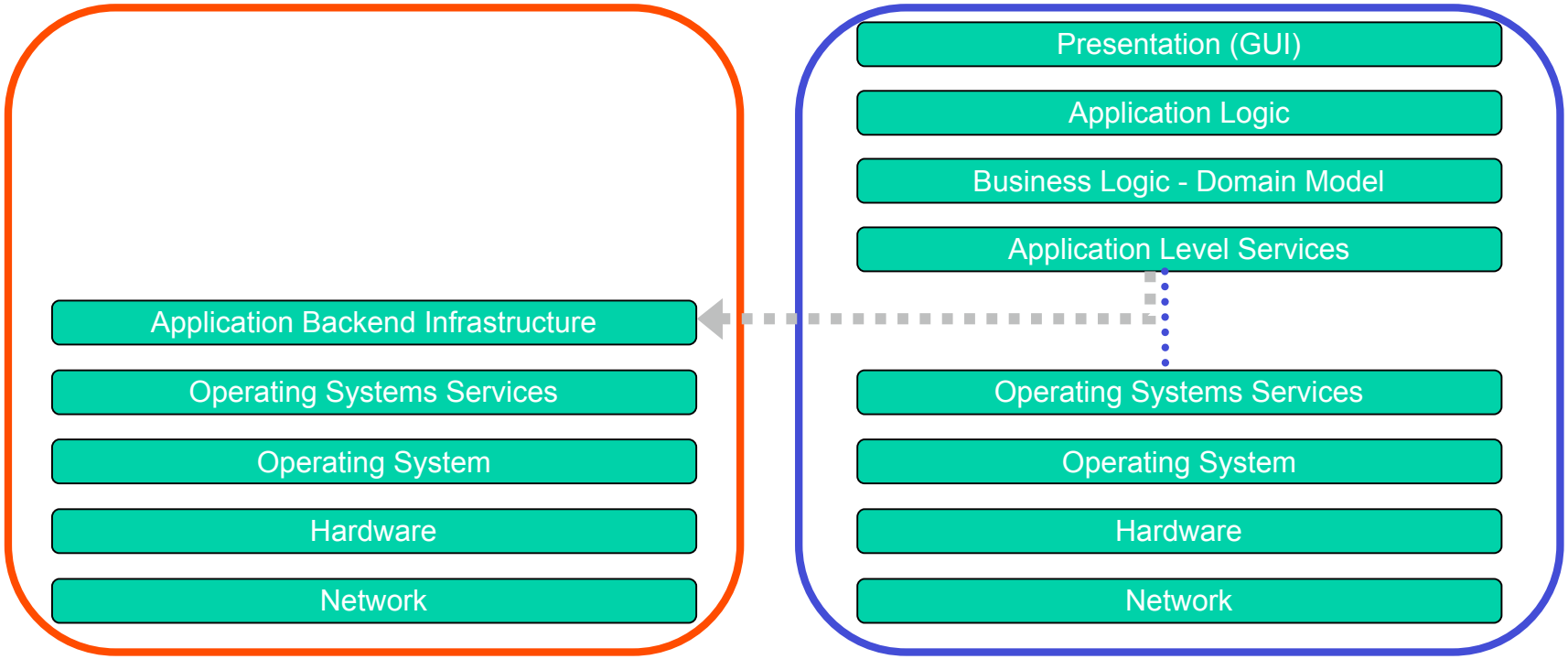


Servidores centralizan la lógica de negocios y el accesso a recursos críticos de la empresa y proveeen servicio a multiples clientes

Las aplicaciones clientes corriendo en computadoras d escritorio se las arreglan con el usuario. Integración via red TCP/IP (y otras)

1980-1995: 2-tier Cliente/Servidor con cliente gordo (“Fat Client”)

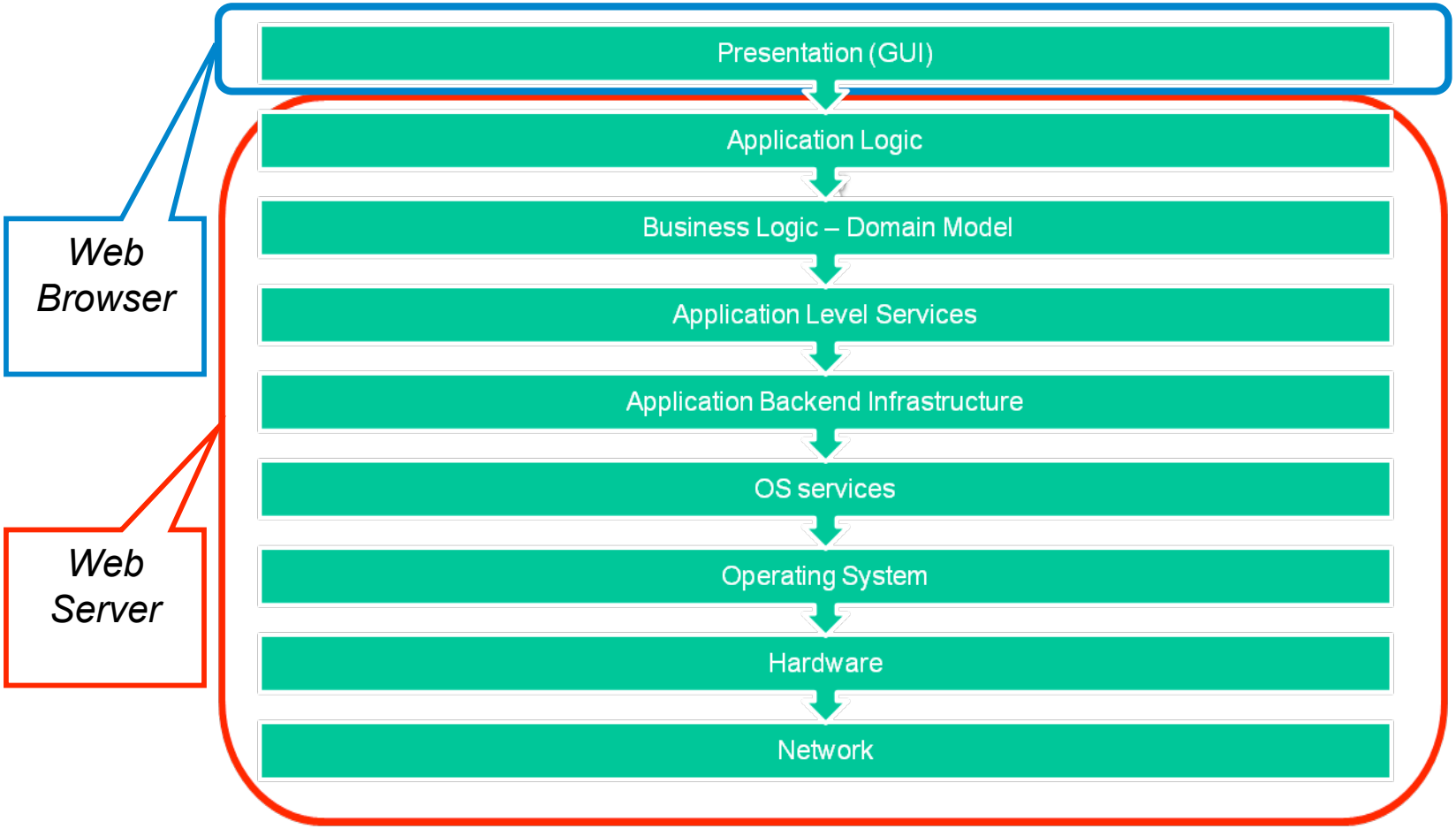
Se corre mucho más código en el cliente



La aplicaciones cliente hace mayor y mejor uso de los recursos de las PCs y reducen la carga de los servers… pero al mismo tiempo transfieren logica de negocios a un ambiente de ejecución que no controlan!

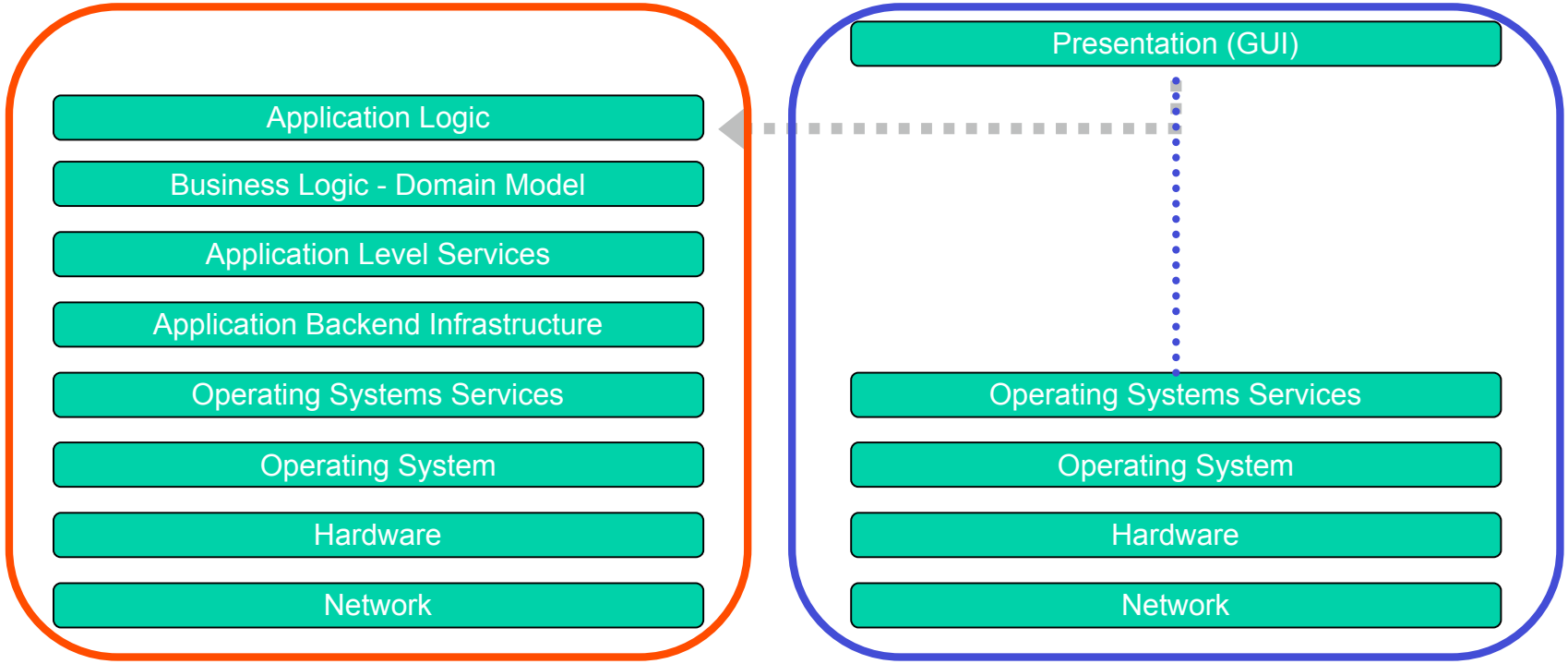
1995-2002: El paradigma Web (1.0)

Alguien se acuerda de "*the network is the computer*" ?



1995-2002: El paradigma Web (1.0)

La segunda forma mas común de tecnología en empresas



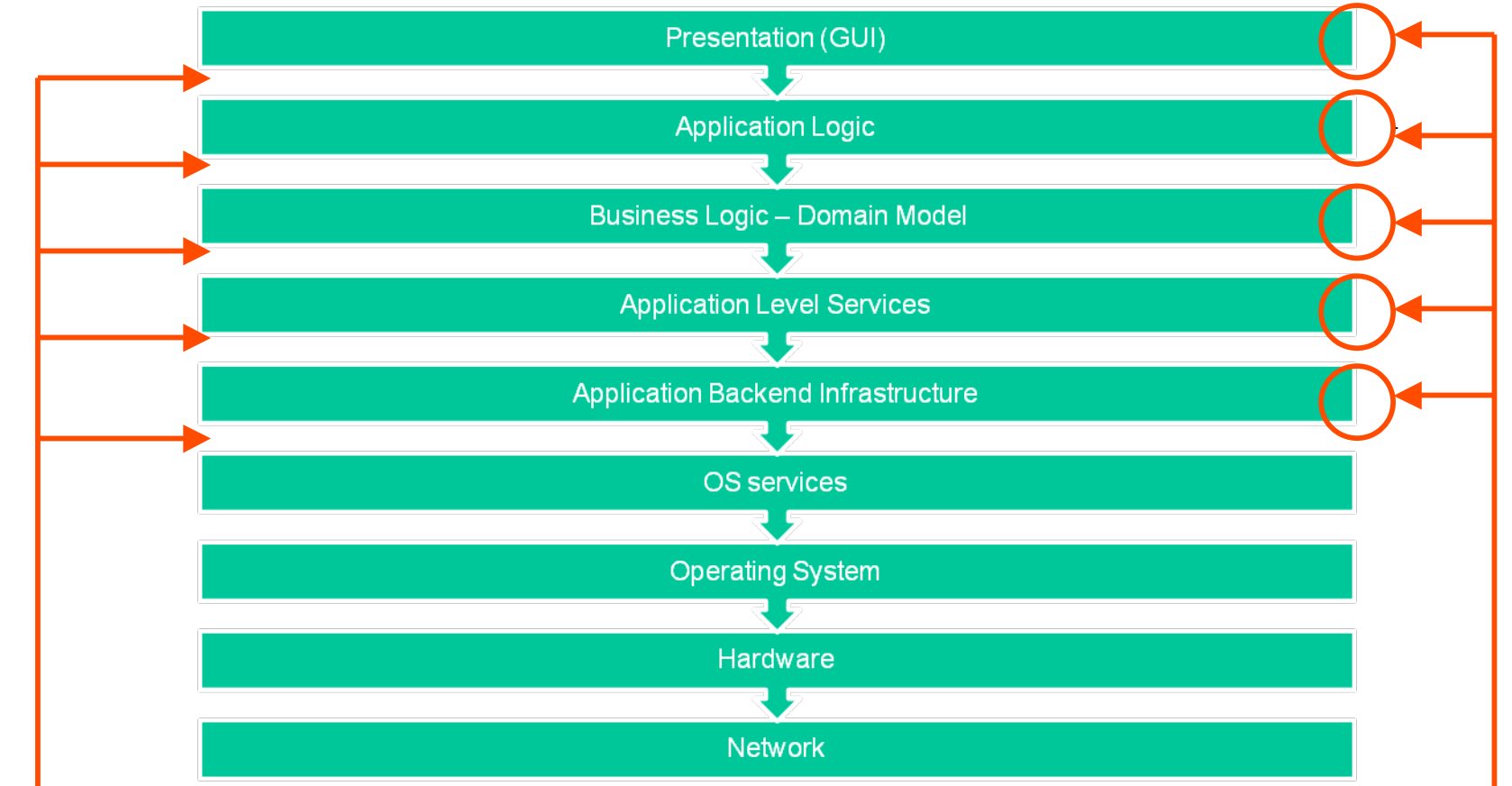
Servidores centralizan la lógica de negocios y el accesso a recursos críticos de la empresa y proveeen servicio a multiples clientes

Cualquier web browser sirve de interfaz con los usuarios (GUI).
…Claro que se necesita procesar y presentar el documento HTML.
Integración sobre HTTP en una red TCP/IP

QUE ES SEGURIDAD INFORMATICA?

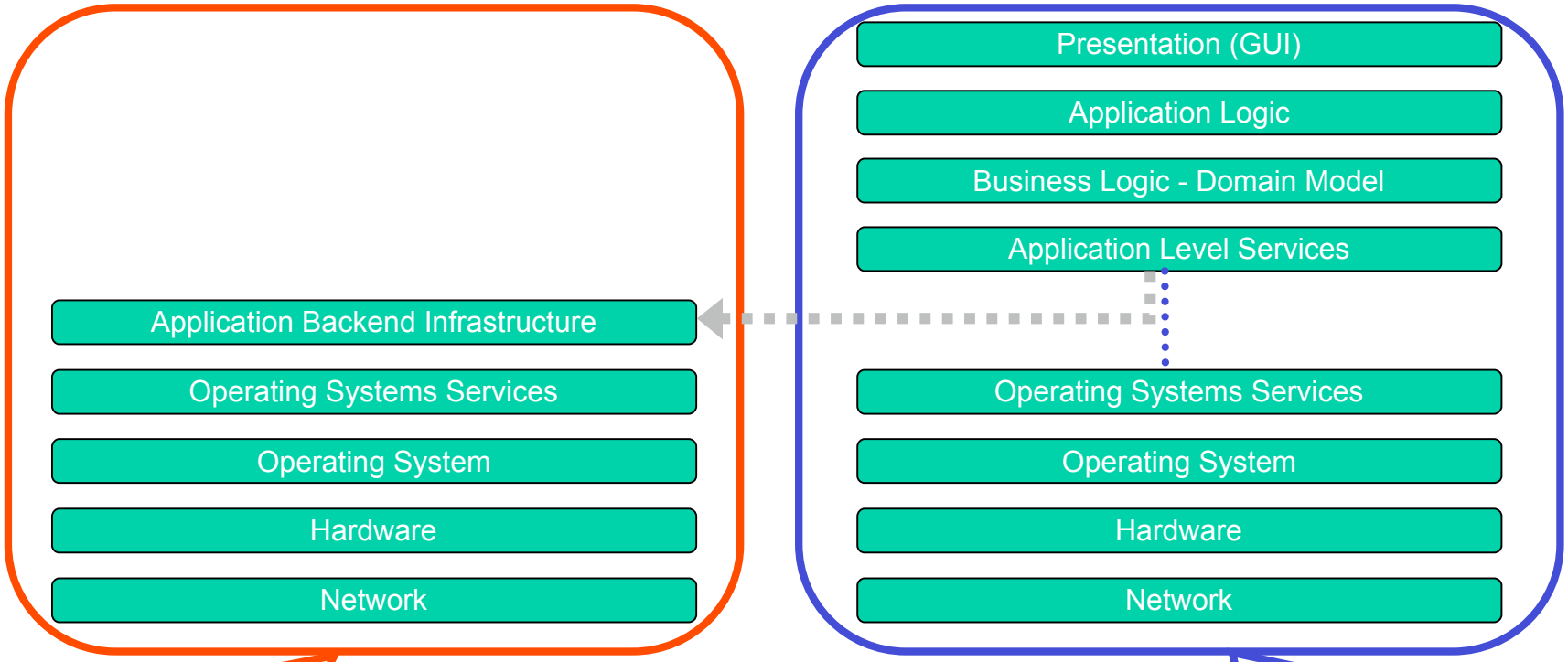
En definitiva , de que se trata la seguridad?

Superficies de ataque y fronteras de confianza



Problemas típicos del modelo Cliente/Servidor

Pocos servidores pero muy críticos, muchos clientes con MAYOR criticidad

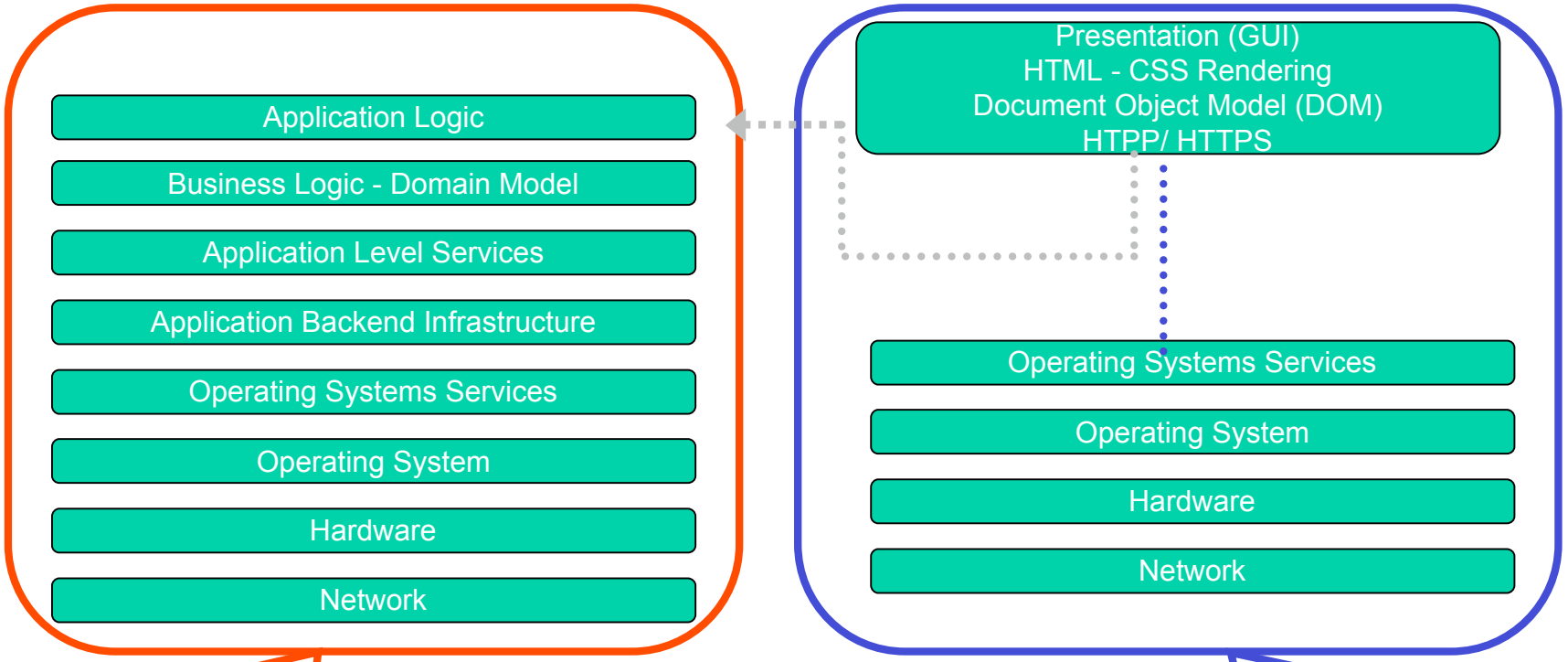


Inyección y ejecución remota de código
Filtración de información confidencial
Transitividad en la cadena de confianza y clientes "malos'"
Fallas de autenticación
Fallas de autorización

*Inyección y ejecución de código en el cliente (**Client-side**)*
Subversión de funcionalidad de la aplicación
Subversión de autenticación / autorización
Transitividad en la cadena de confianza (control del HW)
Server spoofing y MITM

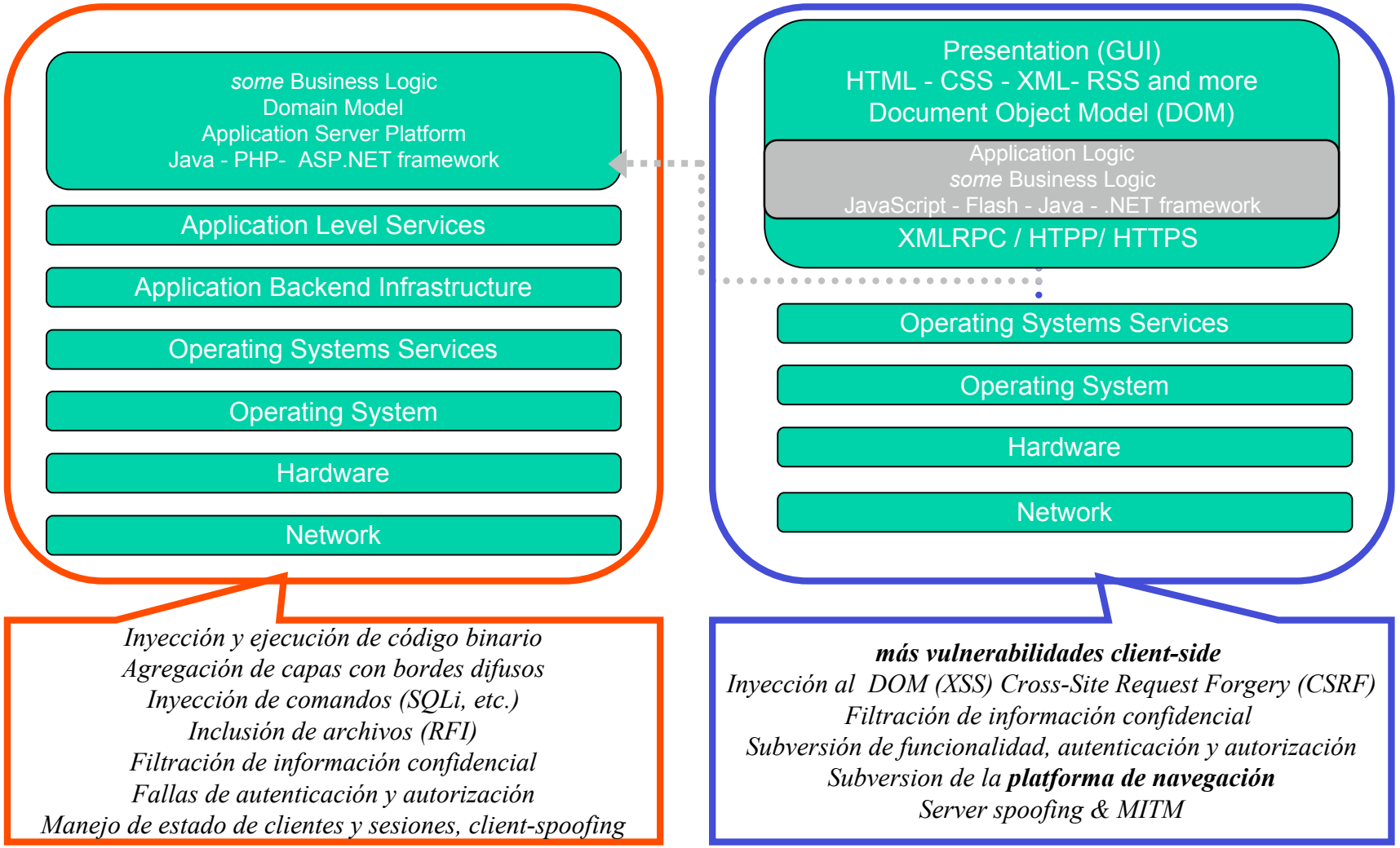
Bueno, y si hacemos que el cliente sea menos “pesado”? ... por ejemplo, si fuera “simplemente” un navegador Web?

El navegador es bastante mas complicado de lo que uno piensa



Que significa “Web 2.0” en este contexto?

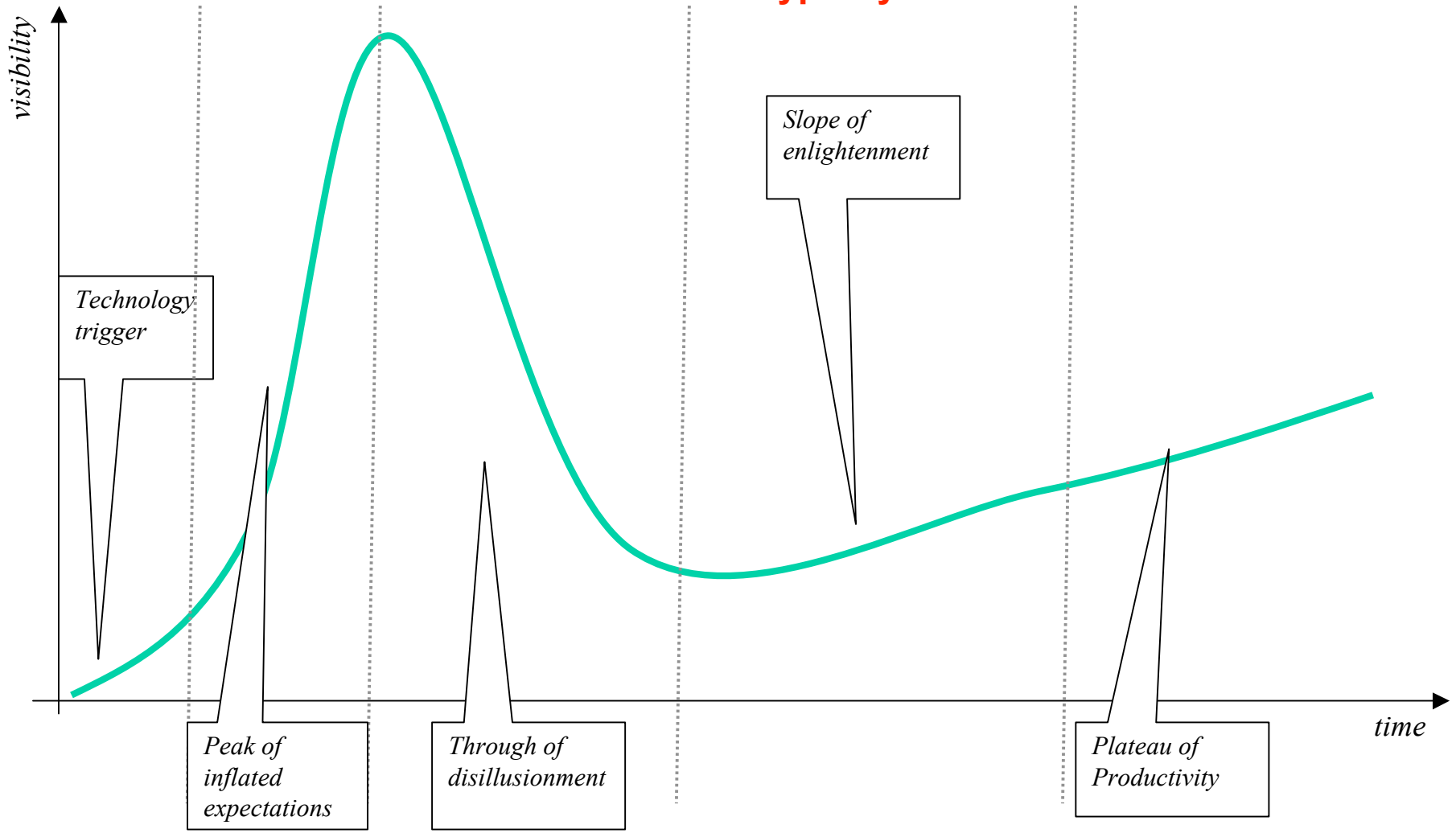
El “paradigma” Web 2.0 aumenta la superficie de ataque del browser



COMO GESTIONAR TODAS ESTAS AMENAZAS SIMULTANEAMENTE?

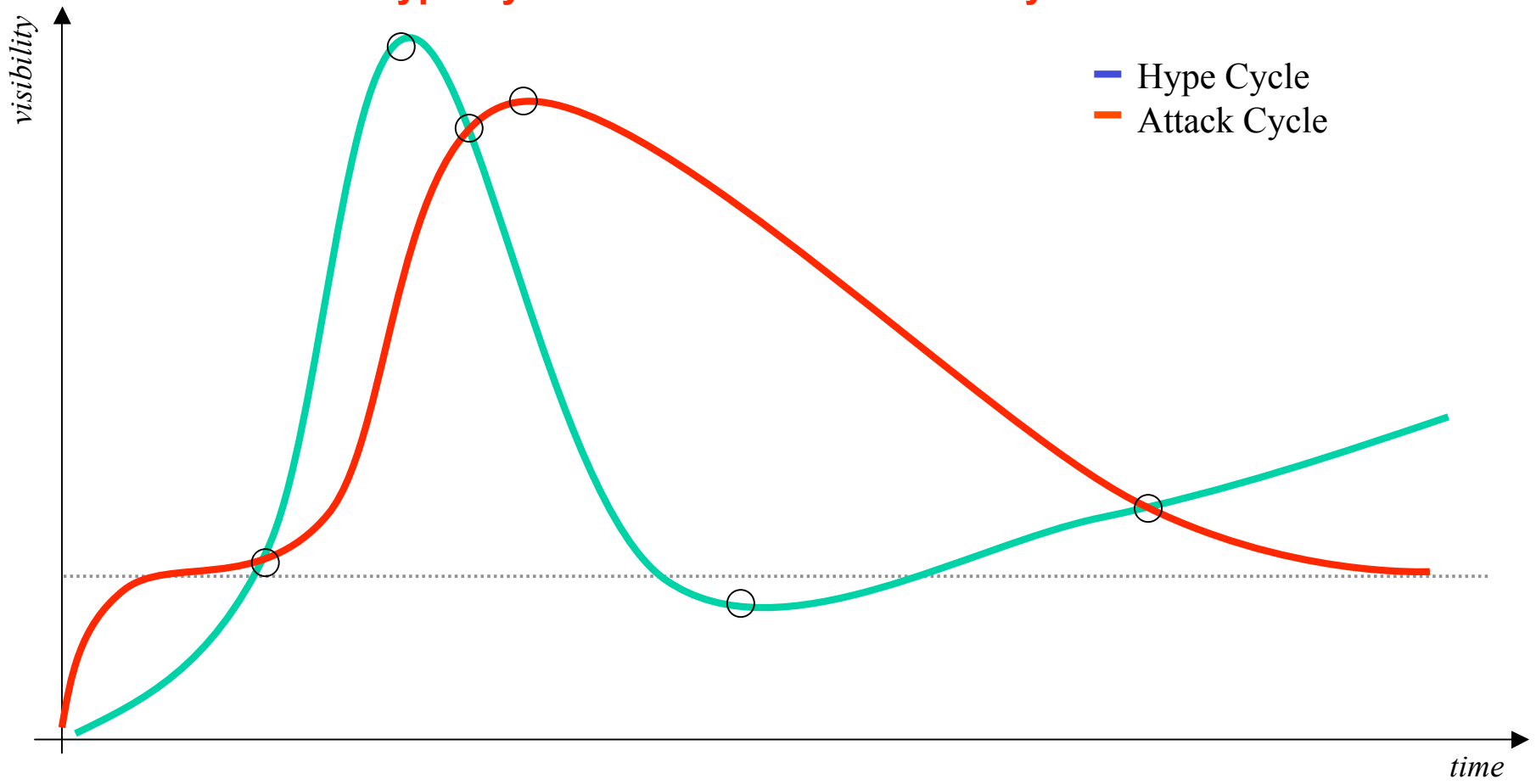
Es valido usar el “ruido del mercado” como un estimador del nivel de amenaza asociado?

Gartner’s Hype Cycle



Y si nos inventamos un modelo propio?

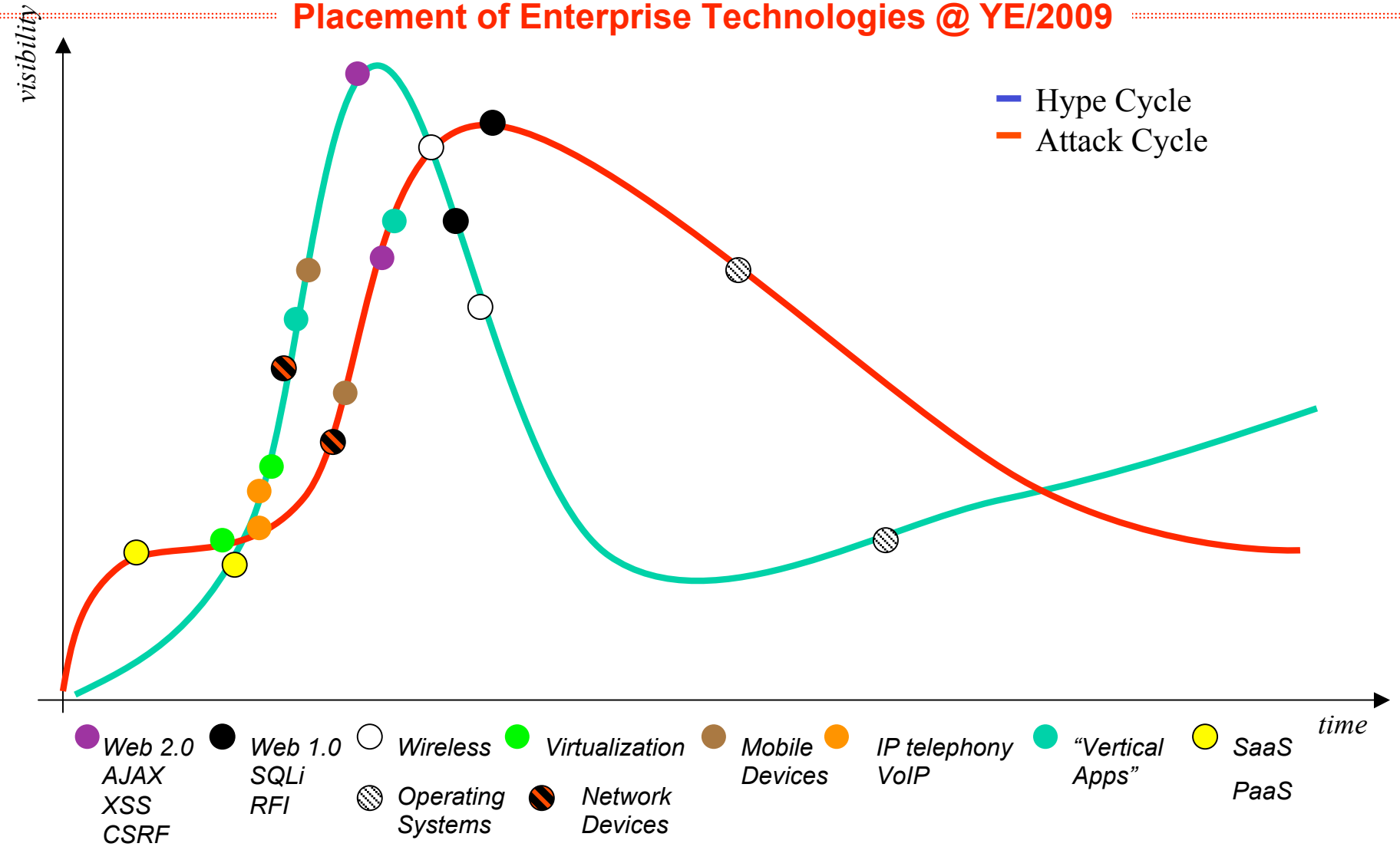
Hype Cycle de Gartner vs. Attack Cycle de Core



Pronostico para fin del año 2009

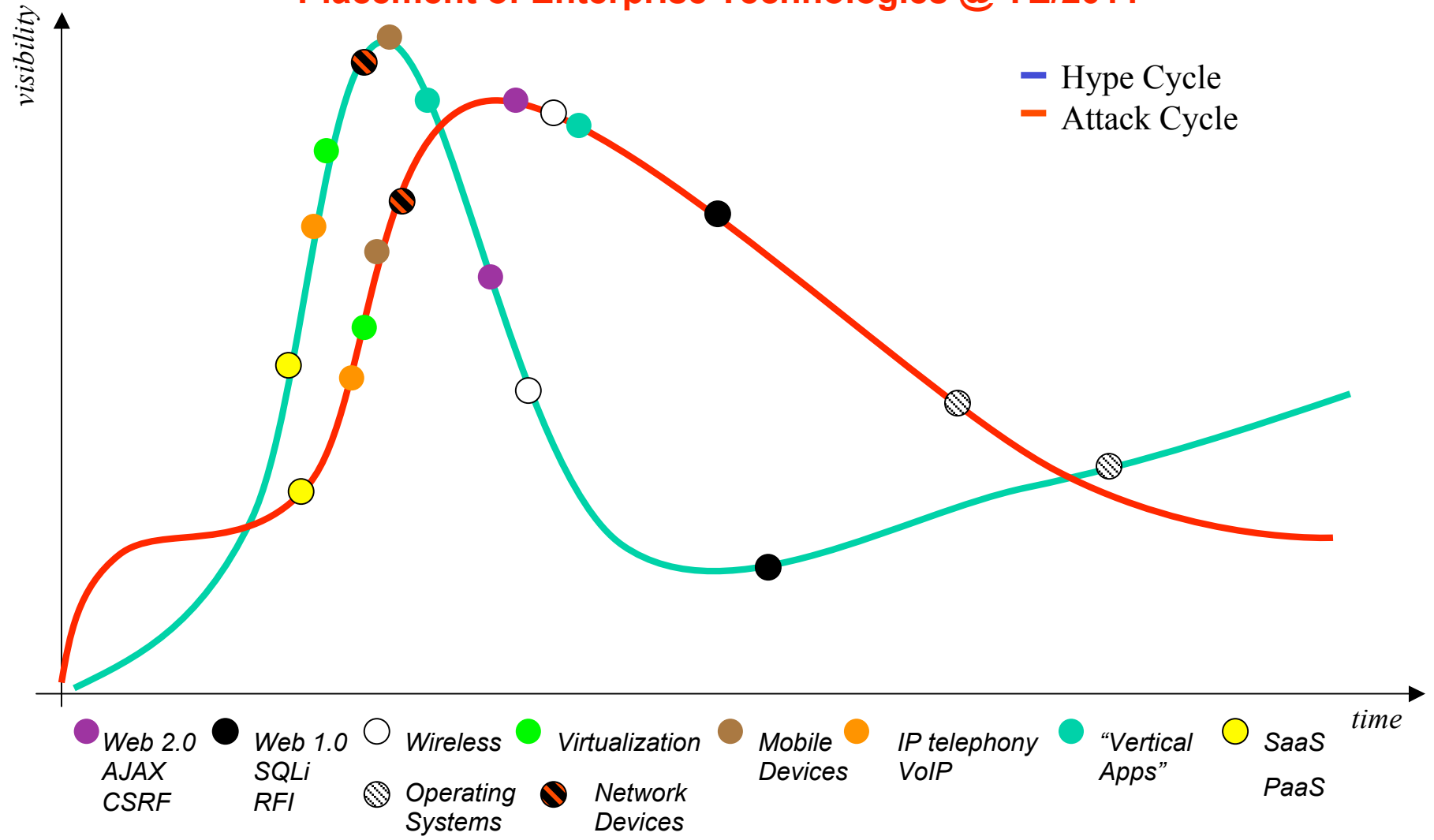
Estimado en Enero 2009

Placement of Enterprise Technologies @ YE/2009



Pronostico para fin del 2011

Placement of Enterprise Technologies @ YE/2011



PENETRATION TESTING

Mi definición de “Penetration Testing”

“Un intento acotado y localizado en el tiempo de vulnerar la arquitectura de seguridad de una organización utilizando las técnicas de los atacantes”

Por qué es desable hacer Penetration Testing?

- **Modelar y gestionar riesgo con una estrategia puramente defensiva induce a error**
- **La percepción de valor y de riesgo de dos sujetos puede ser distinta**
- **Permite contrastar y corroborar hipótesis sobre la postura de seguridad**

Algunas cosas implícitas en nuestra definición

Se implica…

- **Una definición de alcance en el espacio-tiempo**
- **La existencia de un marco de referencia para modelar atacantes**
- **La existencia de algun tipo de "arquitectura de seguridad" a probar**
- **La aceptación del potencial disruptivo de vulneración de dicha arquitectura**
- **La existencia de una *técnica (o…modus operandi)***
  http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=t%E9cnica
  - 1. adj. Perteneciente o relativo a las aplicaciones de las ciencias y las artes.
  - 3. m. y f. Persona que posee los conocimientos especiales de una ciencia o arte.
  - 5. f. Conjunto de procedimientos y recursos de que se sirve una ciencia o un arte.
  - 6. f. Pericia o habilidad para usar de esos procedimientos y recursos.
  - 7. f. Habilidad para ejecutar cualquier cosa, o para conseguir algo.
- **Una sútil distinción entre "como un atacante" y "usando técnicas del atacante"**

=> Se da a entender que hay un propósito u objetivo para el ejecicio

UN POCO DE HISTORIA

El “Penetration Testing” no es algo nuevo

1972-1990

- Comienza cómo una práctica interna de organizaciones militares de EEUU, aplicando conceptos de seguridad física y ejercicios de guerra al campo de la informática. Implementaciones aisladas sobre instalaciones militares y centros de investigación académica relacionados con organismos de defensa.
- Se concentra en definir un marco metodológico y probar que es aplicable
- Utilizado para validar mecanismos de seguridad y/o corroborar hipótesis de falla en sistemas de tiempo compartido
- Se practica tanto sobre ambientes de laboratorio como operativos (“en “producción”)

- **1973:** C. Weissman, *System Security Analysis/Certification Methodology and Results, S*P-3728, System Development Corp., Santa Monica, Calif., October 1973

- **1974:** Paul Karger and Roger Schell, *Multics security evaluation: Vulnerability Analysis* **http://csrc.nist.gov/publications/history/karg74.pdf**

- **1976:** C.R. Attanasio, P.W. Markstein and R.J. Phillips, *Penetrating an Operating System: A Study of VM/370 Integrity* **http://www.research.ibm.com/journal/sj/151/ibmsj1501H.pdf**

Transformación en práctica comercial

1990-2000

- El worm de Internet (RTM)-> Corrupción de memoria, Inyección de código y explotación binaria (Smashing the Stack for Fun & Profit) -> Proliferación de herramientas de ataque implementadas ad-hoc y sobre la marcha
- El foco se pone en buffer overflows, reconocimiento de la red, cracking de claves y explotación de configuraciones inseguras
- Penetration Testing ingresa a la industria de seguridad como una oferta de servicios
- La práctica es adoptada por las grandes consultoras (“big 5”) y consultoras especializadas en seguridad informática
- Metodología no documentada, propietaria y poco formalizada. Sustentada principalmente por la experiencia , conocimientos y capacidades técnicas de los consultores.
- Jerarquía de expertos técnicos
- Aparición y adopción masiva de software de scanning de vulnerabilidades como una forma benigna y menos riesgosa de mejorar la identificación y gestión de riesgo

- **1993:** Dan Farmer y Wietse Venema,*Improving the security of your site by breaking into it* **http://www.fish2.com/security/admin-guide-to-cracking.html**

- **1995:** IBM’s Global Security Analysis Lab liderado por Charles Palmer **http://domino.research.ibm.com/comm/research_projects.nsf/pages/gsal.index.html**

Masividad de ataques y formalización de la práctica

2000-2005

- Code Red-Nimda-Slapper-Slammer worms
- Proliferación en el uso de TCP/IP, invasividad de la Internet -> ataques *Client-Side*
- Proliferación de aplicaciones Web *carenciadas*-> Inyección de SQL , Remote File Inclusion
- Software Development Lifecycle vs. Operational Risk Management
- Mayor sofisticación y mejor organización entre los atacantes
- Especialización técnica y motivaciones de negocios
- Aparición de la oferta de software para PenetrationTesting
  - 2002: CORE IMPACT v1.0 (CoT$)
  - 2003: Metasploit Framework v1.0 (Open Source)
- Adopción de penetration testing como práctica de seguridad
- Ataques distribuidos, botnets, zombies y spam

- **2001:** Ivan Arce y Max Caceres, *Automating Penetration Testing: A new challenge for the IS industry?*
  http://www.coresecurity.com/files/attachments/ArceCaceres_2001-blackhat.pdf

- **2004:** HD Moore y “spoonm”**,** *Hacking like in the movies*
  http://www.metasploit.org/data/confs/blackhat2004/defcon.pdf

El estado actual

2005-2009

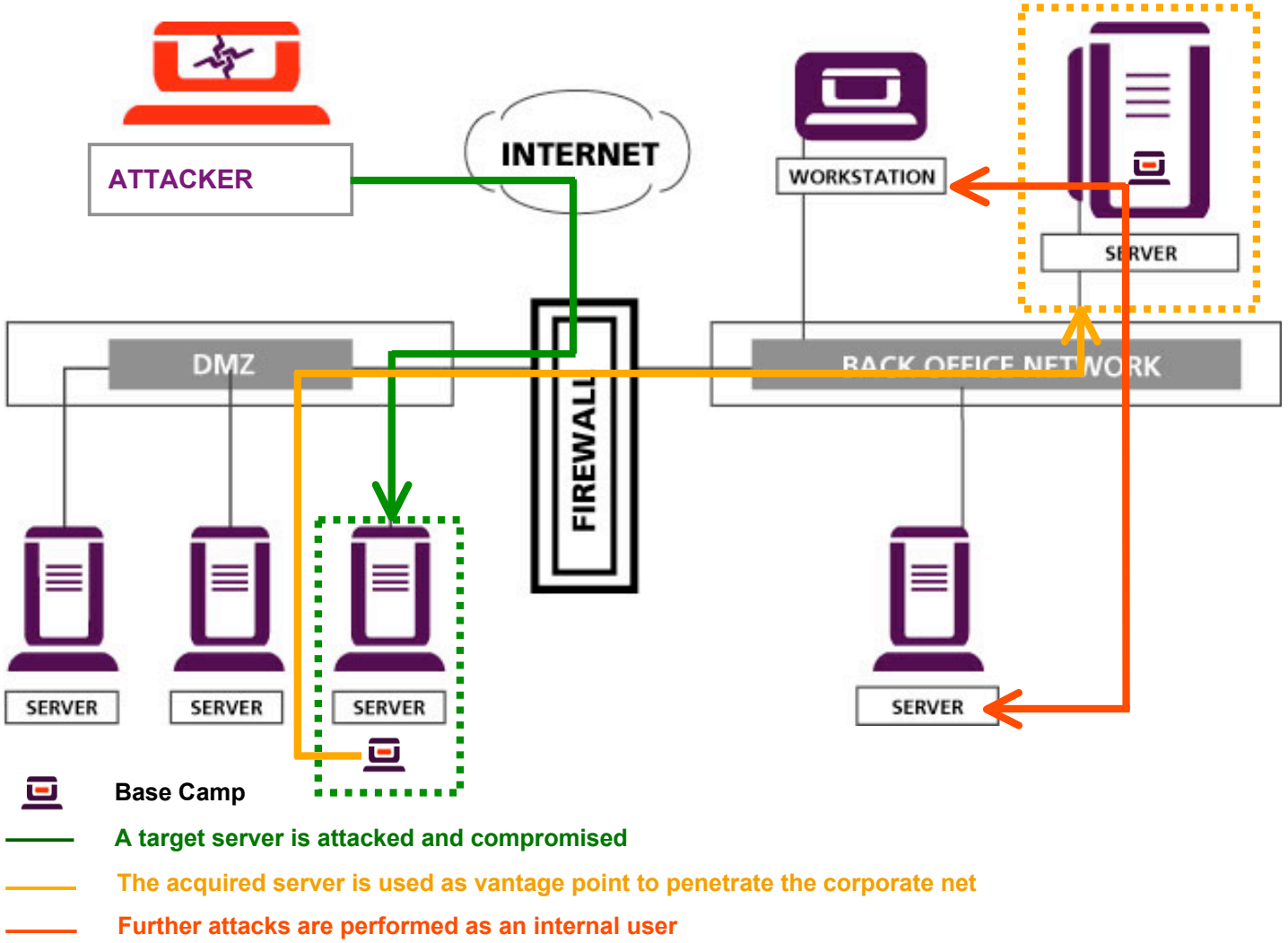
- Proliferación de aplicaciones Web **2.0** “carenciadas” -> **más** SQL injection, RFI & **XSS**
- La euforia de WiFi, Web Services, SaaS, VoIP y los Smartphones
- Ecosistema estable de atacantes especializados
- Multiplicidad de vectores de ataques y herramientas de software específicas
- Productos de detección y análisis de vulnerabilidades de código fuente
- El debate académico: Análisis estático vs análisis dinámico
- El universo paralelo de regulaciones y estándares de seguridad adopta Penetration testing

- **2008:** NIST SP 800-53A, *Guide for Assessing the Security Controls in Federal Information Systems*
  http://csrc.nist.gov/publications/nistpubs/800-53A/SP800-53A-final-sz.pdf

- **2008:** PCI Data Security Standard, *Information Supplement: Requirement 11.3 Penetration Testing*
  https://www.pcisecuritystandards.org/pdfs/infosupp_11_3_penetration_testing.pdf

DESAFIOS PRACTICOS Y SOLUCIONES

Anatomía del ataque clásico de los 90s (1990-2001)



Cómo modelar el trabajo de un Penetration Tester?

- **Extracción y formalización de la metodología:**

```
while (timeleft>MIN_REPORT_TIME {
   information_gathering();
   information_analysis_and_planning();
   vulnerability_detection();
   if(attack_and_penetration())
     privilege_escalation();
}
final_analysis_and_report_generating()
clean_up()
```

- **Uso de "domain-driven design" para diseñar una arquitectura que permita implementarla con software genérico**
- **Debe contemplar todos los aspectos de la práctica y solucionar sus limitaciones**
  - Atribución, repetitividad y reproducción, consistencia, eficiencia y eficacia
  - Capaz de adquirir y transferir conocimiento del dominio
  - Minimizar la disrupción de servicio,
  - Extendible, customizable, mantenible

Implementación en software

Un modelo simplificado y orientado a objetos del dominio

- **Módulo: Cada tarea específica que se realiza durante el penetration test, utilizada como bloque constructivo de acciones para tareas complejas**

  `Implementadas como programas escritos en un lenguaje de scripting de alto nivel (Python)`

  `icmp_network_discovery, tcp_portscan, os_fingerprinter, IIS_remote_exploit, chroot_breaker, atjobs_exploit, vulnerbaility_report`

- **Agente: ambiente de ejecución para acciones**

  `LocalAgent:` Ambiente de ejecución en el sistema del penetration tester, encapsulado en una máquina virtual de alto nivel (Python) execution

- **Host: Un objeto descripto parcialmente sobre el que se aplican las acciones**

  Descripto en función de un conjunto de tuplas (propiedad, valor) como por ejemplo:

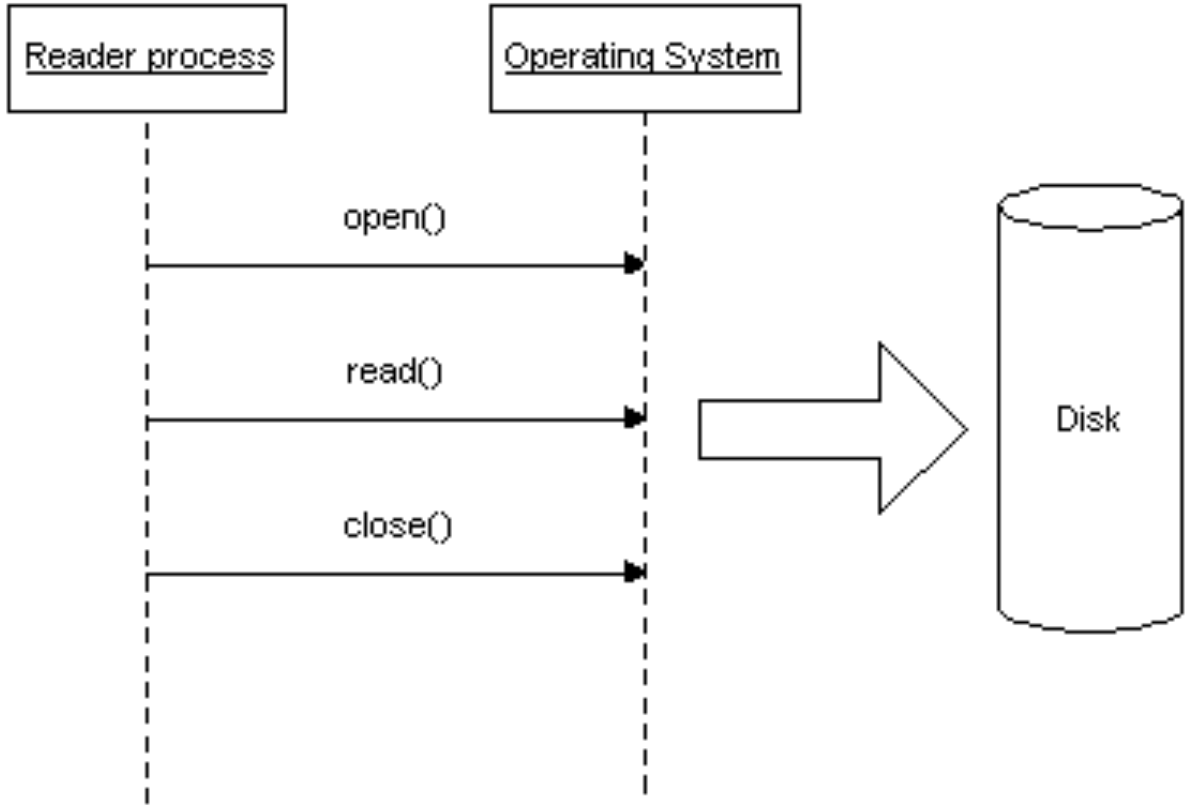
  ```
  { ("os_name",NULL),("tcp_ports",{22,80}) }
  ```

- **Modelo de objectos, Base de Datos para persistencia, GUI, y extensiones**

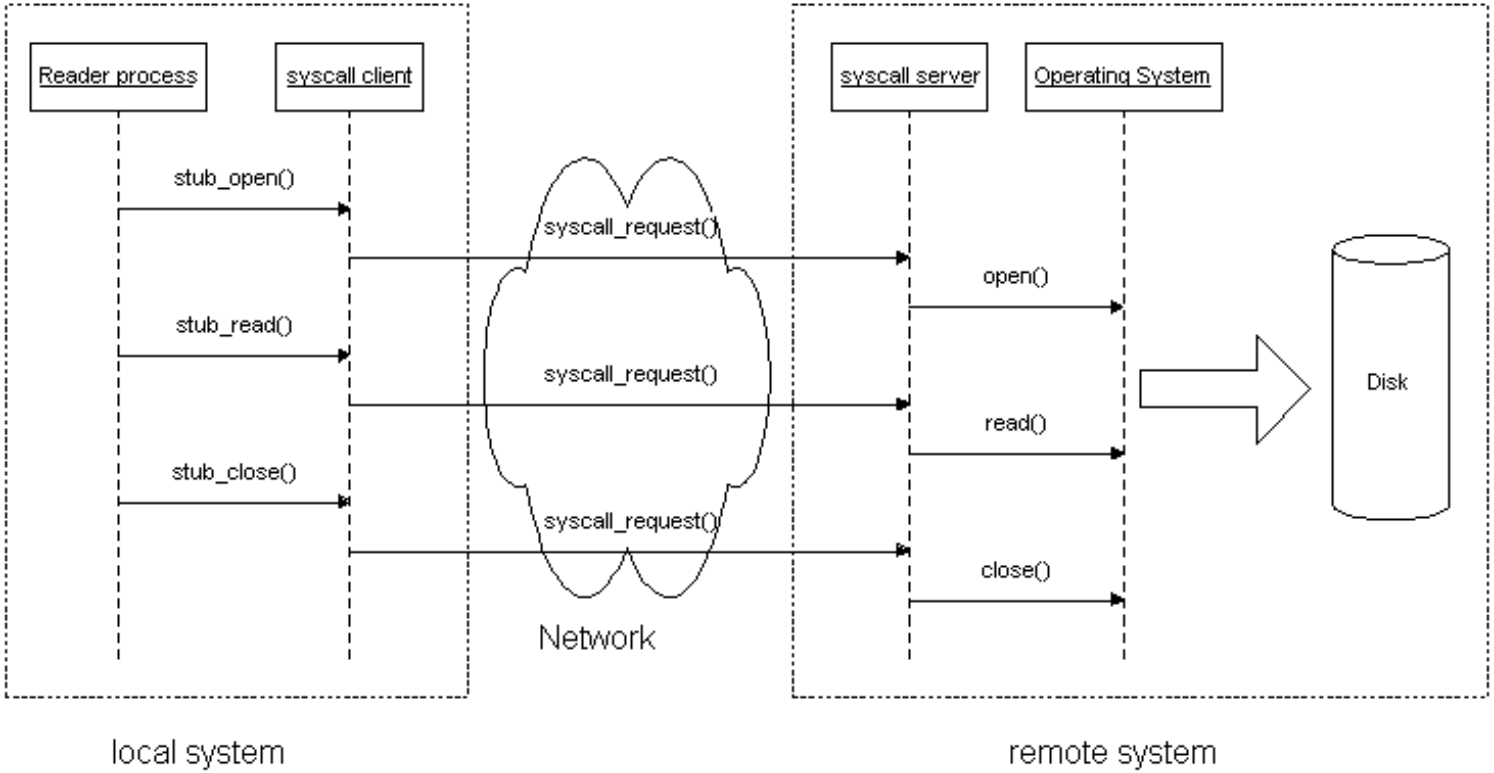
- **Syscall proxying**
  - Pivoteo transparente
  - Minimiza el potencial disruptivo
  - Simplifica el proceso de roll-back

  2002: Max Caceres, *Syscall Proxying: Simulating Remote Execution*
  http://www.coresecurity.com/files/attachments/SyscallProxying.pdf

Como funciona syscall proxing?
Un proceso que lee datos del disco



Con syscall proxing

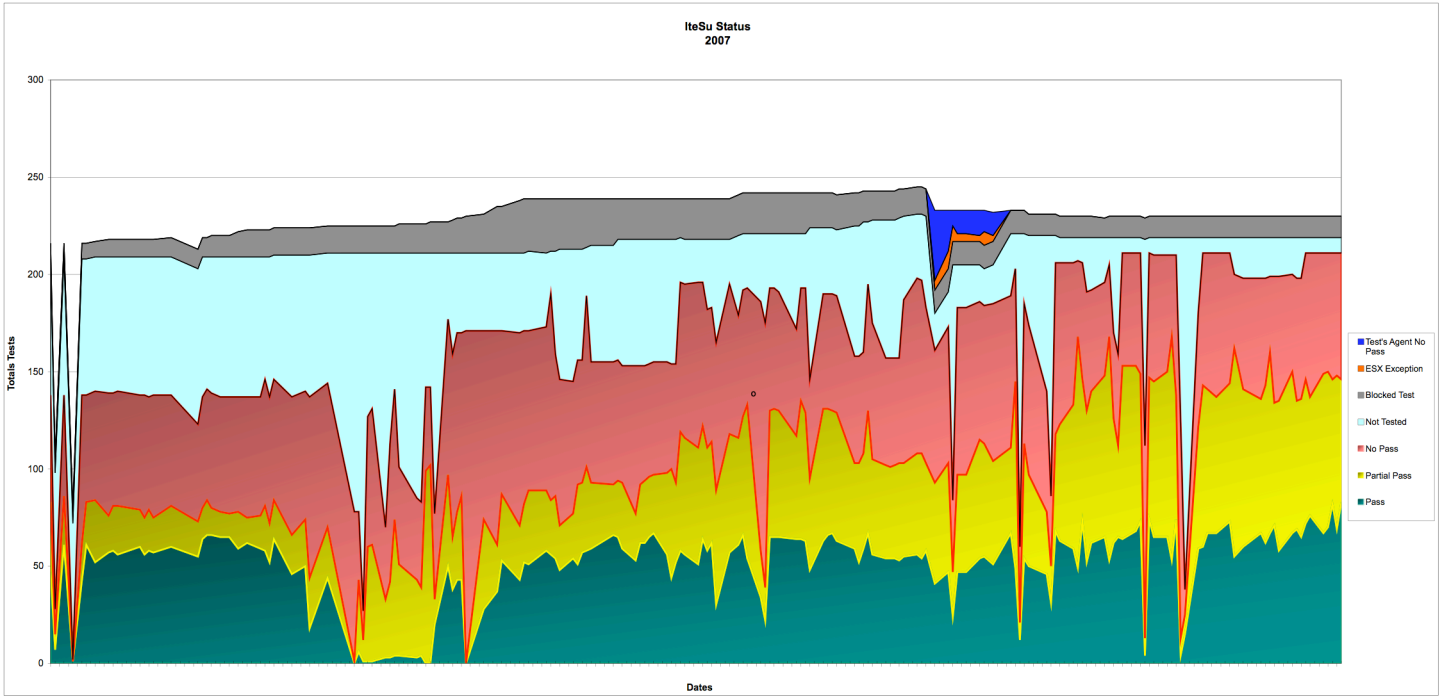


Que pasó cuando usamos ese modelo para hacer un software comercial de penetration testing?

- **La efectividad de las técnicas de Information Gathering depende fuertemente de consideraciones topológicas y ambientales de la red**
  - Existe una multiplicidad de acciones, todas ellas imprecisas, que generan el mismo tipo de información con calidad y confiabilidad variable.
  - Los resultados solo son válidos para algun nivel de confianza y durante un intervalo de tiempo
  - Procesamiento y analisis programático de informacion parcial
- **Modulos de explotación de vulnerabilidades de corrupción remota de memoria**
  - Desarrollo de exploits con grado de calidad requerido para un producto comercial de software
  - Equivalente a la produción de componentes de software que utilizan una ABI de terceros no documentada, no soportada, cambiante… **cuya existencia no estaba prevista!**
- **Explosión combinatoria de blancos potenciales:**
  - OS Name: {Windows 2000}
  - OS Edition: {Professional, Server, Advanced Server, Datacenter}
  - Service Pack level: {SP0, SP1, SP2, SP3, SP4, SP4+hotfixes}
  - Localization & Configuration: {Eng, Fr, Sp, Gr}
  - Opciones de compilacion y linking
  - Variabilidad de las versiones a nivel: VeryBreakable v8, 9i, 10g, 10gR2, 11i
- **A la fecha (Agosto 2009) :**
  - 1200+ exploits
  - 5000+ targets

Pruebas automaticas de regresión

- **Se corren todas las noches usando un conjunto de servers de virtualización.**
  - 450+ VM images, 1000+ exploits, 5000+ targets
  - Multiple attack vectors y mecanismos de conexion
  - Cada exploit se corre contra cada uno de los targets soportados con cada uno de los mecanismo de comunicación posible



El método de fuerza bruta no escala!
Y ahora… que hacemos?

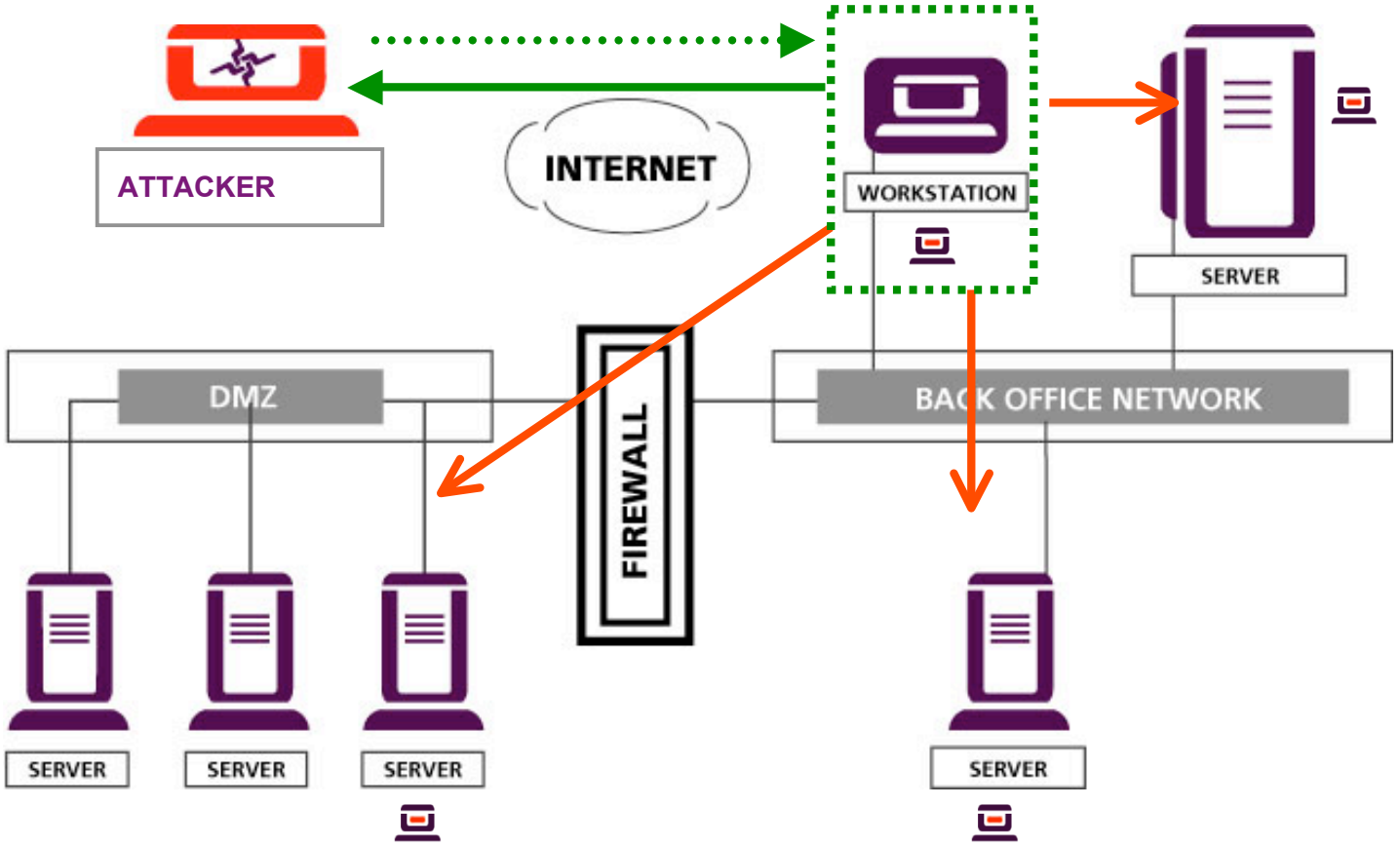
- **El tiempo total de una pruba de regresion completa supera las 8 horas**

- **Por otro lado, no es realista suponer que un usuario (Penetration Tester) puede determinar y seleccionar manualmente que exploit es apropiado para usar de un conjunto de mas de 1000 posibles:**
  - No tiene sentido usar uno para IIS contra un servidor web corriendo Apache
  - Existen dependencias y pre-requisitos para los módulos (y en particular para los exploits)
  - Hay correlacion entre ejecucion de ciertos modulos y fallas en la disponibilidad de los servicios
  - La ejecución de algunas pruebas se puede distribuir y paralelizar
  - Las pruebas y sus resultados no son deteminísticos! El “ruido ambiente” altera los resultados y la medición
- **Un primer problema de optimización:**
  - Minimizar el tiempo total de una prueba completa de regresion
  - Minimizar la probabilidd de romper algun servicio
  - No vale usar soluciones “de jueguete”
- **Automatización de ataques**
  - Armado de planes de ejecucion para sequencias semi’fijas de acciones
  - Implementación vía software de metodologías de penetration testing conocidas (aperturas)
- **Attack Planning**
  - Planning adaptativo online. Control estocástico dinámico
  - Tiene que escalar hasta el orden de 2^{12} modulosy 2^{24} targets (hosts, direcciones de email, paginas web)

2008: Carlos Sarraute and Alejandro Weil, *Advances in Automated Attack Planning*
http://www.coresecurity.com/content/advances-in-automated-attack-planning

Ataques Client-Side (2001+)

CURRENT ATTACK TREND



Base Camp

A target workstations are attacked and compromised

Further attacks are performed as an internal user

Introducción de ataques de Client-Side y aplicaciones web al modelo

- **Los blancos son usuarios y sus estacioens de trabajo:**
  - Phishing, spam, trojanos,

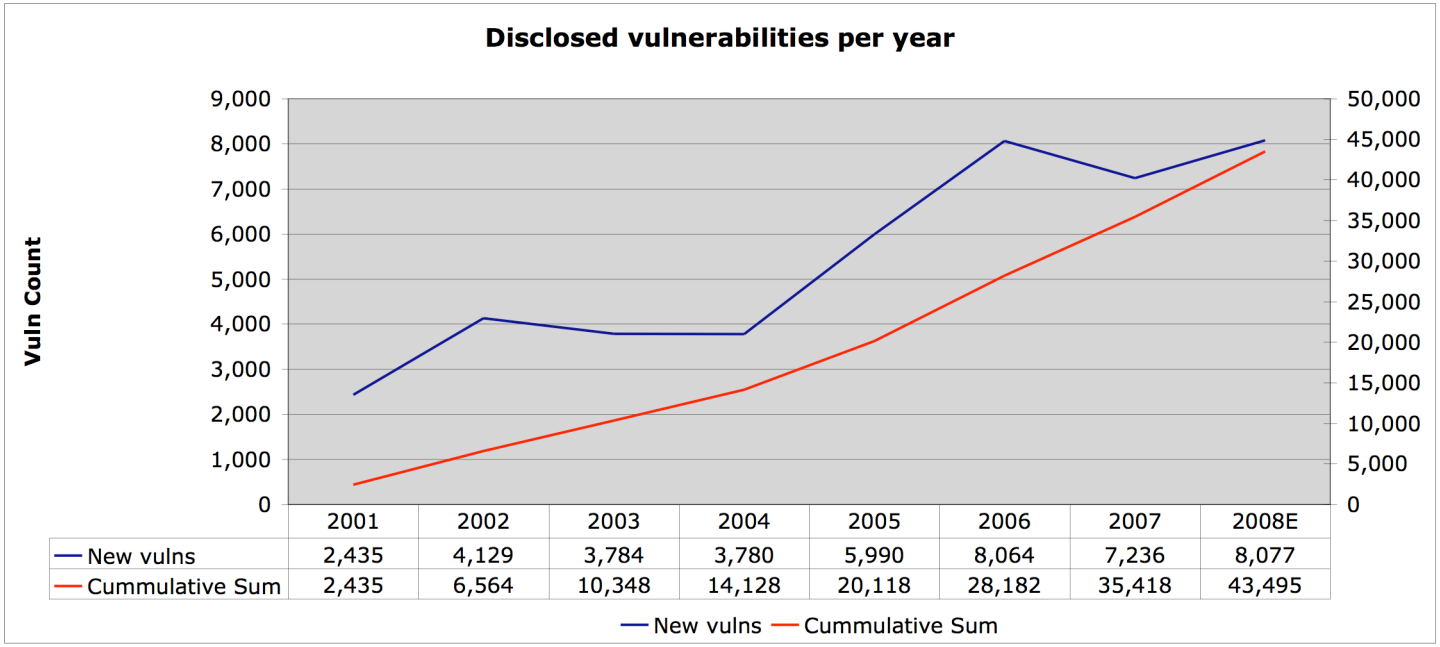
- **Forzo un refactoring del modelo**
  - Ahora el Penetration Tester tiene que proveer infrastructura para los componentes del lado del servidor
  - Los ataques son asimetricos y se comportan de forma no-asincrónica
  - Efectos de la topología no'visible de la red (NAT, Firewalls, routers, etc.)
  - Técnicas tradicionales de Information Gathering no son aplicables
  - Explotación indirecta o "a ciegas"

- **Nuevo refactoring fue necesario para incorporar penetrationt esting de applicaciones web en el modelo**
  - Los ataques de inyección de comandos SQL extraen datos, no inyectan código
  - Los ataques de XSS tienen tuplas de blancos (pagina web, identidad de usuario) y resultan en la obtención de control sobre un navegador asociado a un dominio (no la maquina sobre la que corre)

2007: Fernando Russ and Diego Tiscornia,
Zombie 2.0
http://www.coresecurity.com/content/Zombie-2-0
Agent oriented SQL abuse
http://www.coresecurity.com/content/Agent-Oriented-SQL-Abuse

Cantidad de vulnerabilidades reportadas por año

- **Más de 34,000 vulnerabilidades únicas (CVEs) en la National Vulnerability Database**
- **Más de 44,000 vulnerabilidades procesadas por CERT/CC desde 1995**

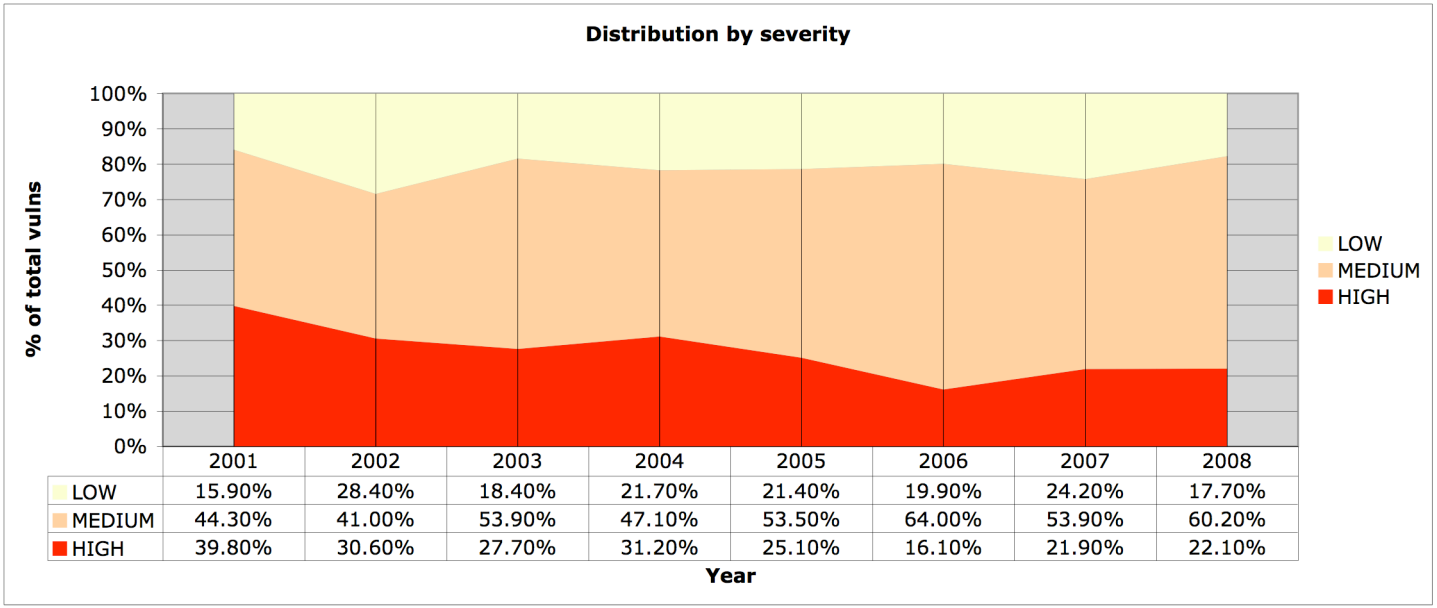


	2001	2002	2003	2004	2005	2006	2007	2008E
New vulns	2,435	4,129	3,784	3,780	5,990	8,064	7,236	8,077
Cummulative Sum	2,435	6,564	10,348	14,128	20,118	28,182	35,418	43,495

Fuente: CERT/CC 2008 estimado usando los datos reales vistos entre Q1-Q3 2008 (6,058)

Si bueno, pero cuantas son realmente relevantes?

- Un estimado conservador: 20% (de 8,000 anuales)



	2001	2002	2003	2004	2005	2006	2007	2008
LOW	15.90%	28.40%	18.40%	21.70%	21.40%	19.90%	24.20%	17.70%
MEDIUM	44.30%	41.00%	53.90%	47.10%	53.50%	64.00%	53.90%	60.20%
HIGH	39.80%	30.60%	27.70%	31.20%	25.10%	16.10%	21.90%	22.10%

Source: IBM/ISS X-Force 2008 Mid-Year Trend Statistics

Mas desafíos…

- **Dada la cantidad creciente de vulnerabilidades que un atacante potencial podría explotar**
  - Como determinar cuales implementar dado un conjunto finito de recursos?
  - Decaimiento del valor del exploit en funcion del tiempo
  - Decaimiento del valor total del conjunto de exploits en funcion del tiempo
  - Es posible generar exploits programaticamente?
  - Como medir la calidad de un exploit?

- **Attack Paths & Attack Graphs**
  - Attack graph: Grafo dirigido (multiple y con loops). Targets y módulos.
  - Attack paths: Conjunto de caminos del attack graph
  - Determinar el conjunto mínimo de nodos a sacar para hacer el grafo no-conexo
  - Determinar el conjunto mínimo de aristas a sacar para hacer el grafo no-conexo
  - Determinar el conjunto mínimo de nodos a sacar que remuevan todos los caminos
  - Representación gráfica de grafos con miles de nodos y aristas

- **La multiplicidad de nuevos vectores de ataque tensiona el modelo**

- **Como adquirir y trasferir conocimiento de un conjunto de especialistas expertos?**

The future…

- **Crecimiento exponencial de la cantidad de nodos en las redes**
- **La ley de Moore tambien se aplica a la capacidad de los atacantes**
  - Poder de cómputo
  - Procesamiento distribuido
  - Cloud computing
  - Costo de adquisición y mantenimiento de botnets
- **Y entonces como modelamos y emulamos efectivamente a un atacante?**
- **Agregación de niveles de abstración**
  - Operativo
  - Táctico
  - Estratégico
- **Penetration Testing y Manejo de Riesgo**
  - Value at Risk (VaR), Annual Loss Expentancy (ALE)
  - Ausencia de datos sobre incidentes, valoración subjetiva de activos, activos intangibles
  - Concepcion constructivista/reduccionista del riesgo
  - Riesgo como carácterística emergente
- **Economía, Ciencias Sociales, Biología, Física, Epistemología.**
  - Teoría de Juegos, Dinamica Evolutiva, sistemas adapativos complejos, Inteligencia Artificial, Aprendizaje

.epilog

Y al final.. Para que me sirvió escuchar todo esto??

- **Un marco de referencia para pensar en seguridad y tecnología**
- **Visión actualizada de problemas prácticos de seguridad**
- **Visión del atacante como componente necesario en la estrategia de seguridad**
- **Discusión sobre seguridad informática:**
  - Es una disciplina técnica?
  - Es una disciplina científica?
  - Es una disciplina empirica?
  - Todas las anteriores?
  - Sólo todas las anteriores?
- **Que puede aportar el mundo académico?**
- **Que puede aportar la industria?**
- **Que puede aportar la comunidad de practicantes y profesionales?**

GRACIAS!