

Web security

2007-11-21

Robert Malmgren
rom@romab.com
+46-708-330378
www.romab.com



Robert Malmgren AB
Trust is good
control is better

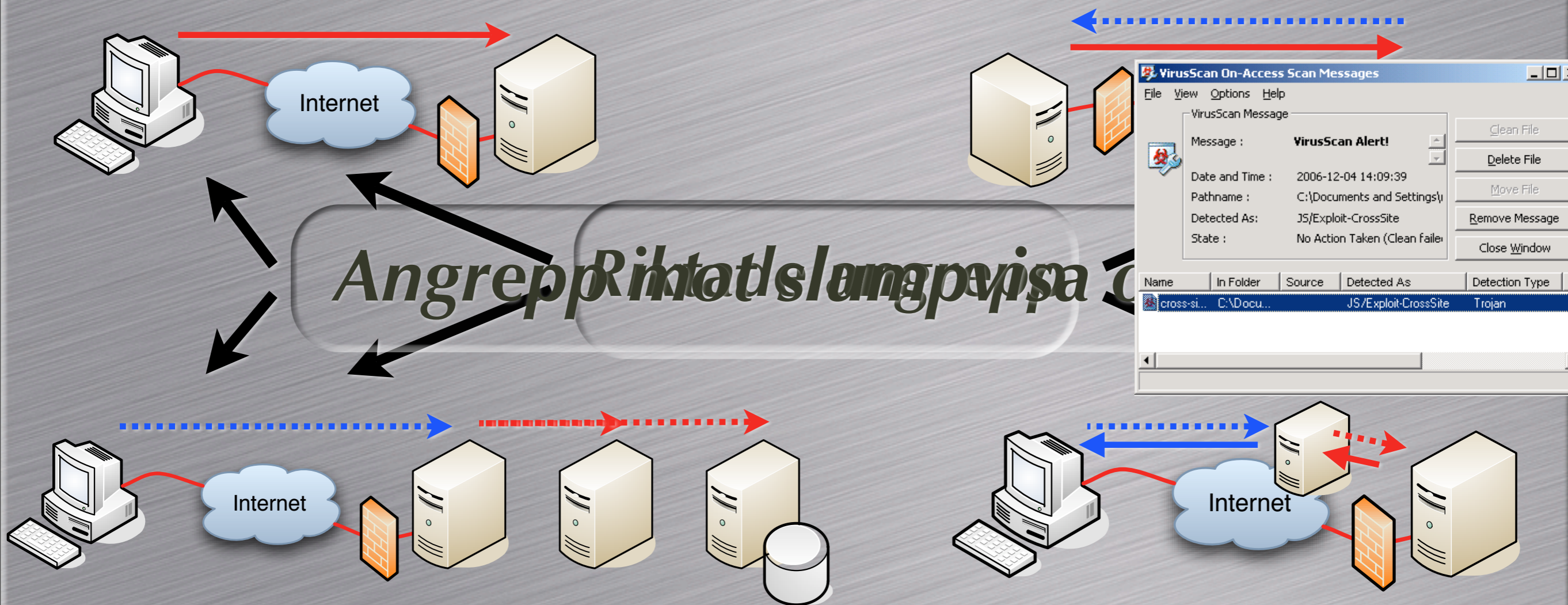
Upplägg

- Angreppsscenarios
- Hot och risker
- Säkerhetsinitiativ
- Något om nya teknologier
- Ofta funna dumheter
- Do's & Don'ts
- Säkerhetskultur & säkerhetsmognad
- Sammanfattning

Angreppsprinciper (1)

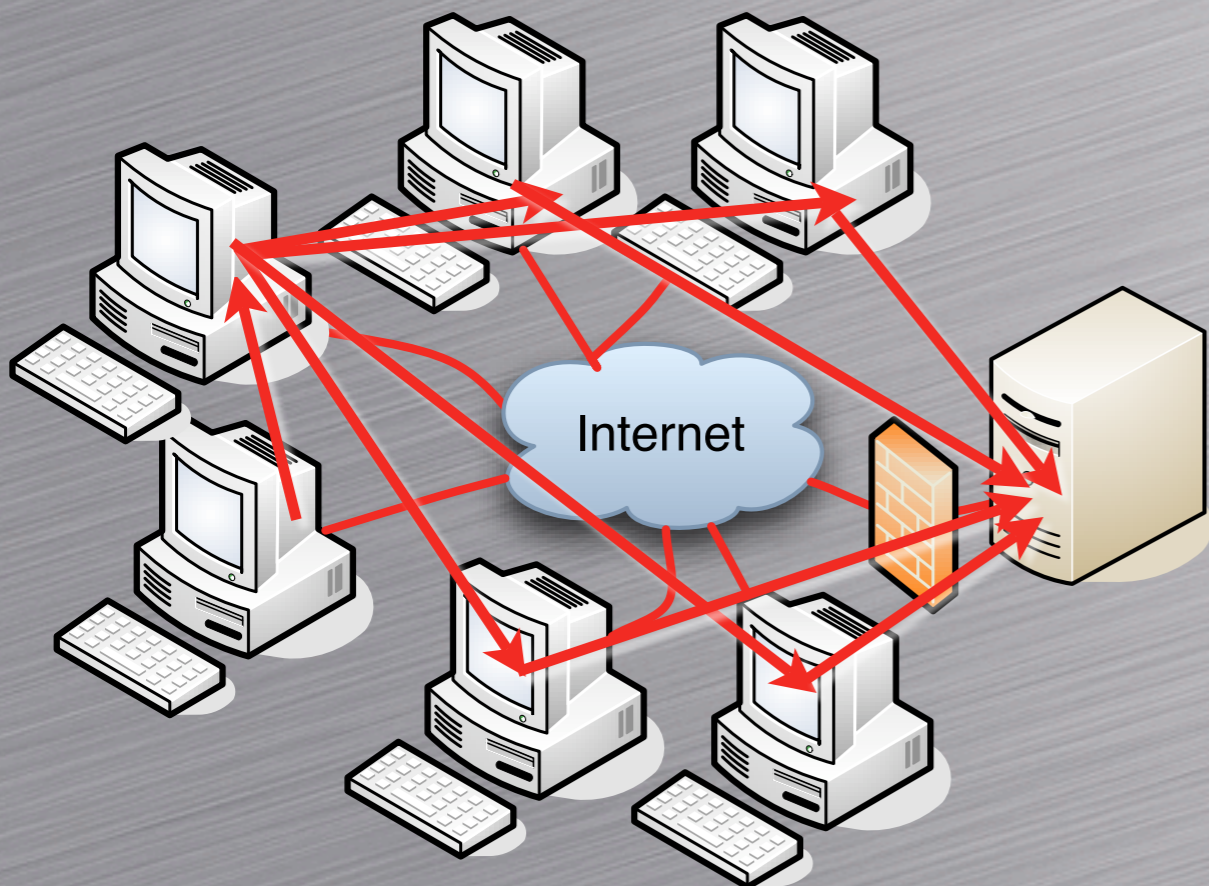
Klient/Server angrepp

Server/Klient angrepp

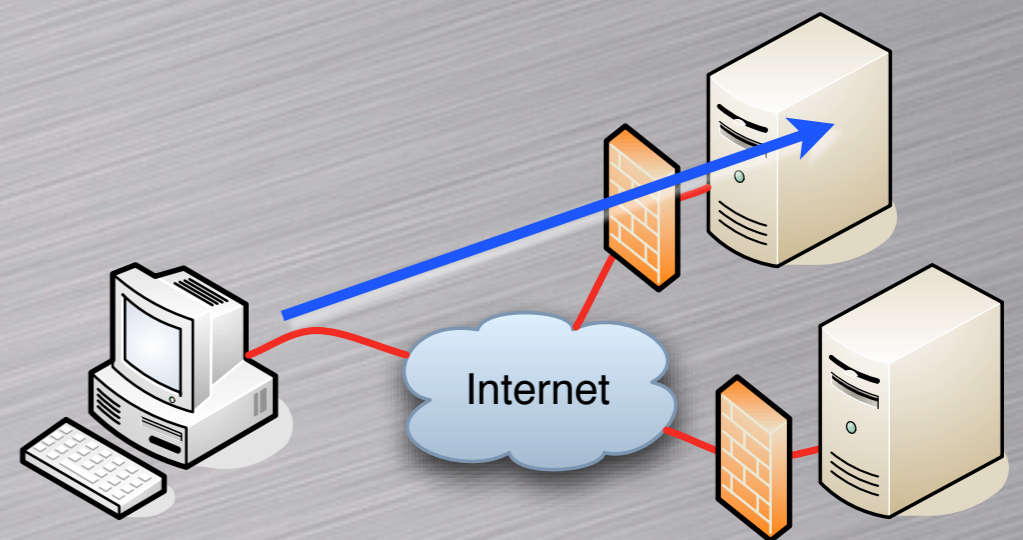


Angreppsprinciper (2)

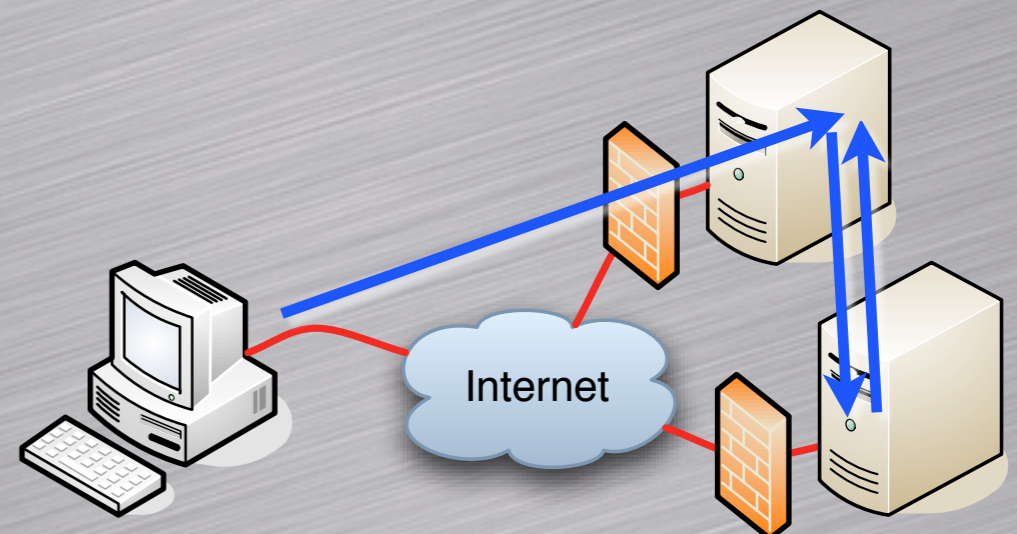
Distribuerad DoS-attack



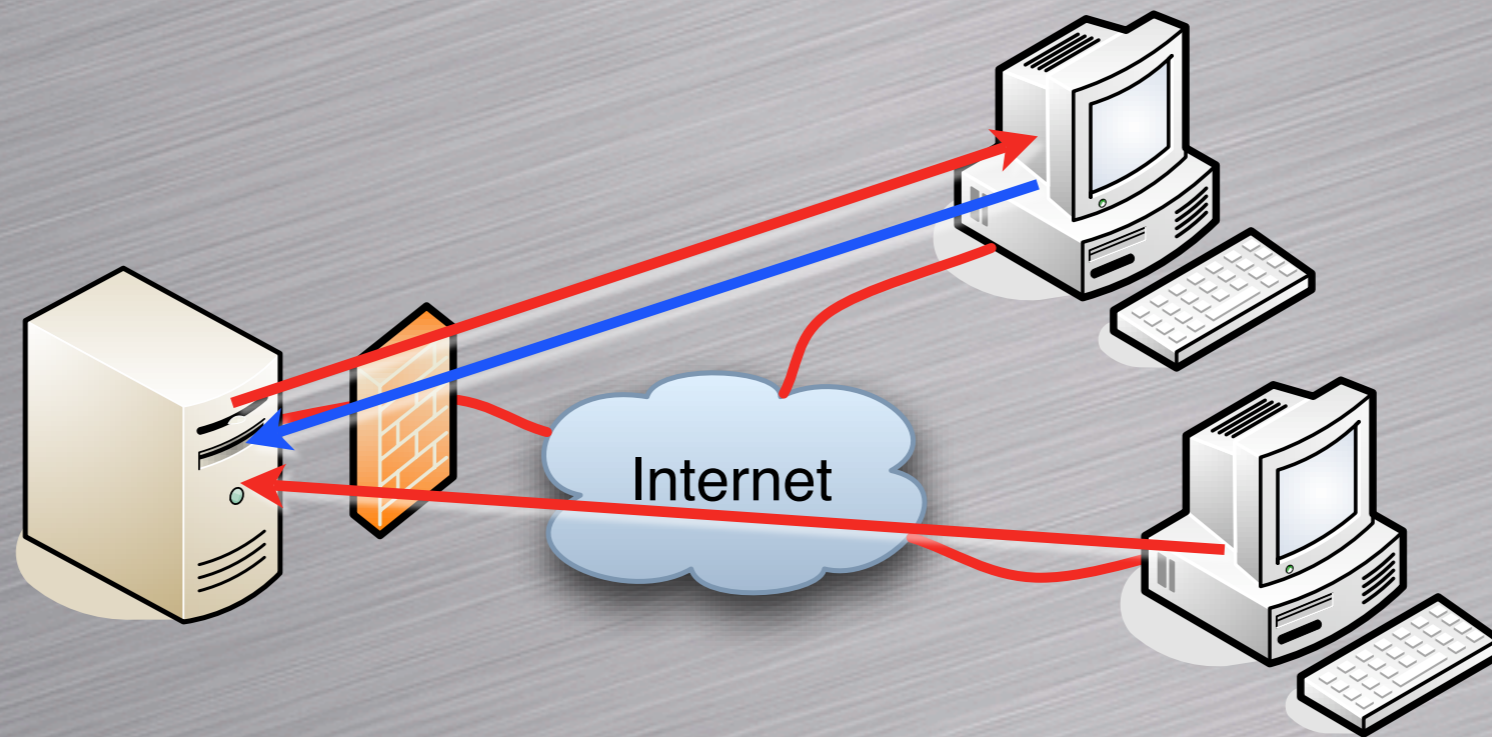
Förtal/parodi/phising-site



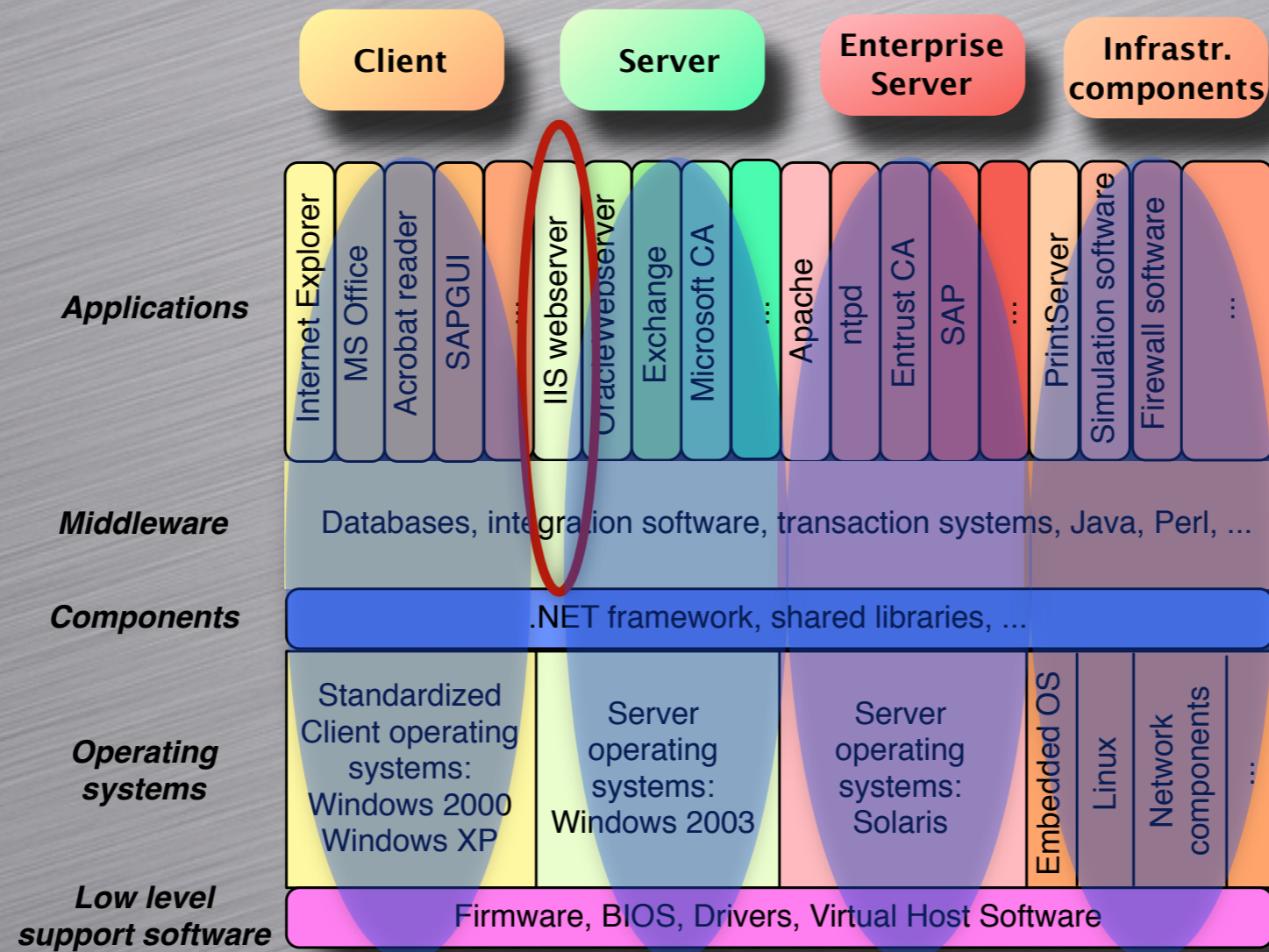
Google hacking



Angreppsprinciper (3)



Komplexitetsfaktorn



Hot

Tekniska hot

Information på webbsidan förvanskas
defacement, *bedrägeri*, *malwarespridning*

Värddatorn övertas och missbrukas av
obehöriga

Omedvetna fel

Opublicerad information
läcker ut (*vid fel tillfälle*)

Interna system innehållande känslig
information blir tillgängliga för angrepp
pgr för hård integration utan kontroll på
risker och typ/nivå av exponering

Legala eller andra hot

Juridiskt ansvar för hur
webbplatsen nyttjas eller
missbrukas: chattar,
kommentarsfält, gästbok

Medveten
handling

Spårdata och annan
dynamisk inte sparad och
går inte att återskapa

Hot: Jakten på säkerhetshål fortsätter

the Month of Apple Bugs

http://projects.info-pull.com/maob/

the Month of Apple Bugs
"Go ahead. Make my day."

bugs faq about press(ure) disclaimer

the Month of PHP Bugs
"formerly known as March"

http://www.php-security.org/

about bugs faq press(ure) disclaimer

BUGS

- MOPB-01-2007
- MOPB-02-2007
- MOPB-03-2007
- MOPB-04-2007
- MOPB-05-2007

LINKS

- Hardened-PHP Project
- Suhosin PHP Protection
- PHP Project

About

This initiative is an effort to improve the security of PHP. However we will not concentrate on problems in the PHP language that might result in insecure PHP applications, but on security vulnerabilities in the PHP core. During March 2007 old and new security vulnerabilities in the Zend Engine, the PHP core and the PHP extensions will be disclosed on a day by day basis. We will also point out necessary changes in the current vulnerability management process used by the PHP Security Response Team.

(Hardened-PHP Project, 2007).

Bugs

#	Title	Description	PoC/Exploit	References
5	PHP unserialize() 64 bit Array Creation Denial of Service Vulnerability	Deserialisation of malformed PHP arrays from within unserialize() might result in a tight endless loop exhausting CPU resources on 64bit systems.	Not needed	CVE-2007-0988
4	PHP 4 unserialize() ZVAL Reference Counter Overflow	During unserialisation of user supplied data that contains a lot of references to a variable the internal 16bit zval reference counter can overflow. This leads to an exploitable double dtor condition.	MOPB-04-2007.php	CVE-NO-NAME MOPB-01-2007
3	PHP Variable Destructor Deep Recursion Stack Overflow	The destruction of deeply nested PHP arrays will exhaust all available stack which leads to remotely triggerable crashes.	Not needed	CVE-NO-NAME
2	PHP Executor Deep Recursion Stack Overflow	A deep recursion of PHP userland code will exhaust all available stack which leads to a sometimes remotely triggerable crash.	Not needed	CVE-2006-1549
1	PHP 4 Userland ZVAL Reference Counter Overflow Vulnerability	In PHP 4 userland code is able to overflow the internal 16bit zval reference counter by creating many references to a variable. This leads to an exploitable double dtor condition.	MOPB-01-2007.php	CVE-NO-NAME

Frequently Asked Questions(FAQ)

The following list of questions and answers provides some information regarding the motives and related facts about the MOPB, such as involved products and disclosure terms. Please check that your question isn't already answered here before attempting to contact us. Any unsolicited e-mail, offensive or non-sense will be ignored.

- Is this an attack, revenge, conspiracy or some kind of evil plot against PHP, the PHP Group, the PHP Developers or the users of PHP?

Not at all. The Hardened-PHP Project has improved the security of PHP for years by reporting serious and sometimes not so serious security holes to the developers and the public. We also killed some vulnerabilities in the PHP CVS snapshot versions before they ever made it into PHP releases. You should consider the Month of PHP Bugs a result report for just another audit we did on PHP. Unfortunately when you disclose security problems in someone else's code or in their bug handling process the developers often feel hurt or attacked.

Browser Fun

feed://browserfun.blogspot.com/feeds/posts/default

25 total

Putting the fun in browser fun hdm 17 aug -06, kl. 16.07

Orphan Objects bug was silently fixed avivra 14 aug -06, kl. 10.24

MS06-044 - Internet Explorer 5.x hdm 8 aug -06, kl. 20.30

AxMan ActiveX Fuzzer hdm 4 aug -06, kl. 00.24

Concluding the Month of E

MoBB #31: Safari KHTMLP

MoBB #30: Orphan Object

MoBB #29: ADODB.Record

MoBB #28: Mozilla Navigat

MoBB #27: NDFXArtEffect

MoBB #26: Opera CSS Bac

MoBB #25: Native Function

MoBB #24: Forms.ListBox

MoBB #23: NMSA.ASFSou

MoBB #22: Internet.HHCtrl

MoBB #21: CEnroll stringT

MoBB #20: OVctl NewDefa

MoBB #18: WebViewFolder

MoBB #15: FolderItem Acc

MoBB #14: Konqueror rep

MoBB #10: DXtFilter Enab

MoBB #9: DirectAnimation

MoBB #8: RDS.DataContro

MoBB #2: Internet.HHCtrl

Welcome to the Browser F

Kernel Fun

http://kernelfun.blogspot.com/

Kernel bugs and madness.

Thursday, November 30, 2006

MOKB-30-11-2006: Apple Airport Extreme Beacon Frame Denial of Service

Apple Airport Extreme driver fails to handle certain beacon frames, leading to an out of bounds memory access, resulting in a so-called kernel panic. Other security implications may exist, although this hasn't been verified and no details can be provided until further research is done. This issue is being coordinated with Apple, and under common agreement it's been decided to keep the details private until a fix has been made available to end-users.

More details

Posted by Unnamed at 5:19 PM 7 comments Links to this post

Labels: [macosx](#), [memory corruption](#), [remote](#), [wireless](#)

Wednesday, November 29, 2006

MOKB-29-11-2006: Linux 2.6.7 - 2.6.18.3 get_fdb_entries() Integer Overflow

Linux 2.6.7 - 2.6.18.3 get_fdb_entries() function is vulnerable to an integer overflow condition. This could be abused to force memory allocation of an attacker controlled size. Successful exploitation could allow arbitrary code execution.

More details and debugging information

Posted by Unnamed at 4:26 PM 0 comments Links to this post

Labels: [integer overflow](#), [linux](#)

Tuesday, November 28, 2006

MOKB-28-11-2006: Mac OS X shared_region_make_private_np() Memory Corruption

About Kernel Fun

Besides hosting the Month of Kernel Bugs, this blog aims to provide information about kernel-land bugs, hacks and tricks. XNU, Linux... you name it.

Best Deal at Ebay Stores

DIE MOSEL PORCELAIN CHINA PLATE PORZELIAN

In Stock, Available via eBay.co.uk at Ebay Stores

Best Deals from Name Brand Merchants

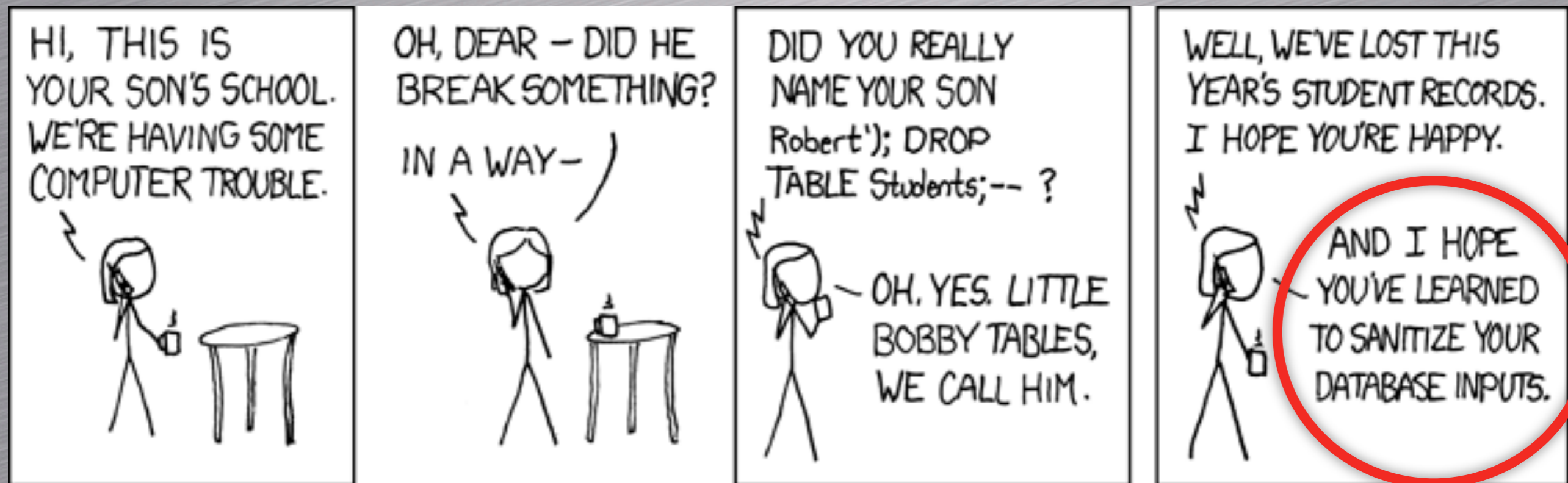
Best Price at: **ebay.co.uk Ebay Stores £4.99**

Featured Store In Stock, Available via eBay.co.uk at Ebay

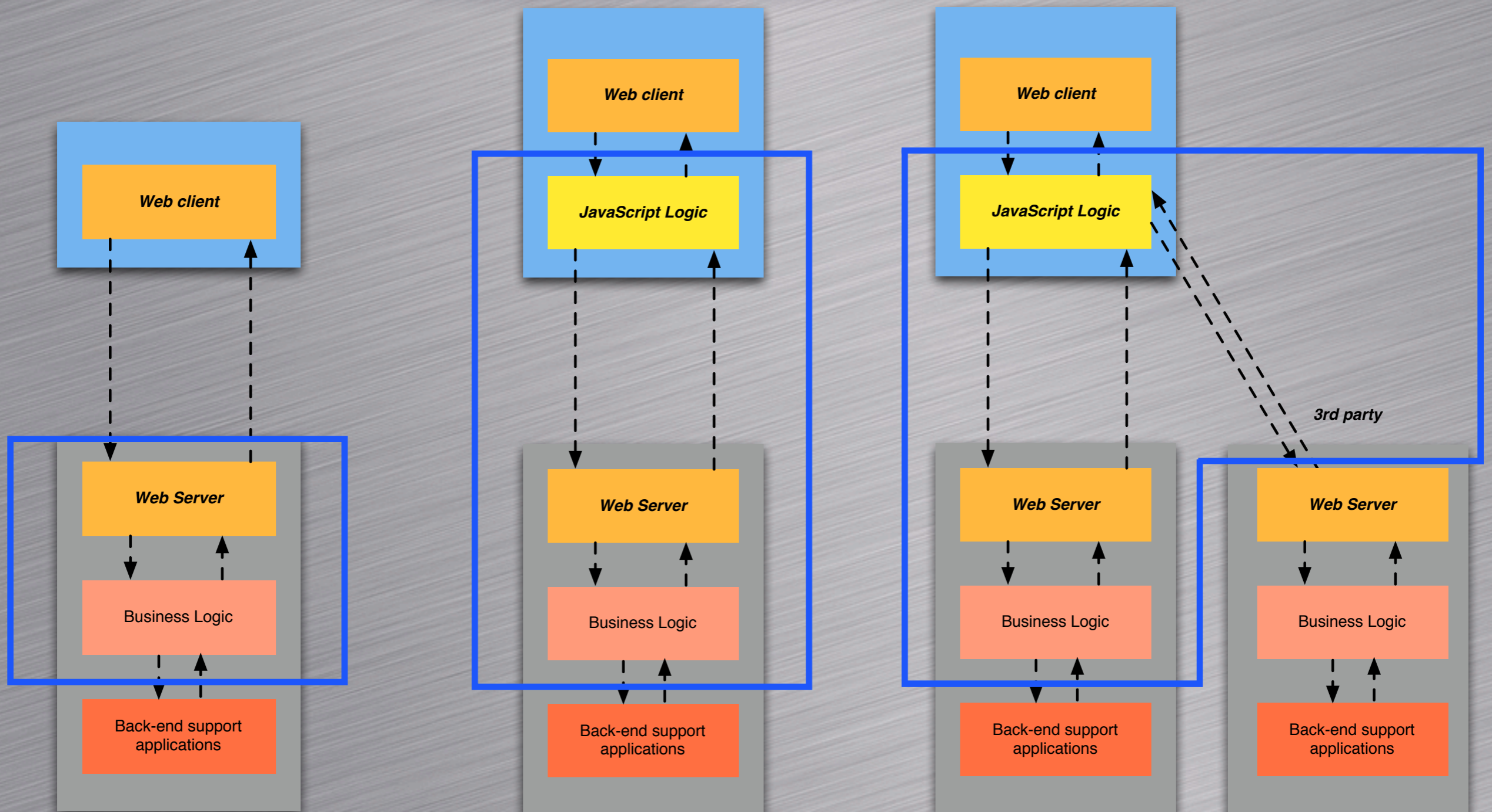
Labels

- [denial_of_service](#) (16)
- [macosx](#) (13)
- [memory corruption](#) (13)
- [sfuzzer](#) (11)
- [linux](#) (11)
- [remote](#) (9)
- [wireless](#) (7)
- [microsoft windows](#) (6)
- [dmg](#) (3)
- [integer overflow](#) (3)
- [mach-o](#) (3)
- [ufs](#) (3)
- [freebsd](#) (2)
- [hfs](#) (2)
- [ISO9660](#) (1)
- [ancient](#) (1)

Säkerhetsproblem i populärkulturen...

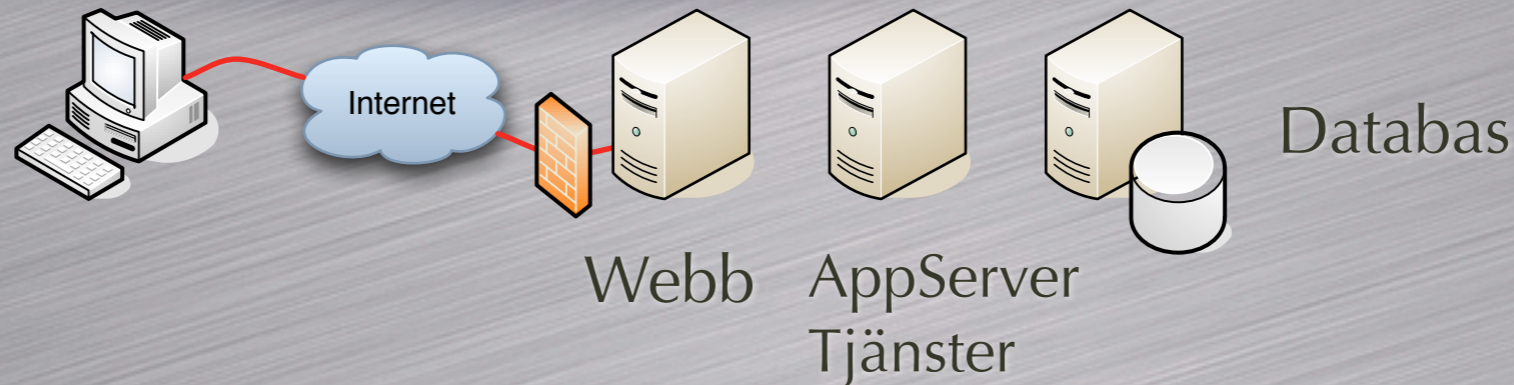


Buzz word bingo: WebServices, Web 2.0



Buzz word bingo: WebServices, Web 2.0

HTML & CSS
JavaScript
XHR-objekt
HTTP



- Web 2.0 öppnar upp för nya typer av angrepp mot, inte minst, klienten
- Skapar en mängd nya säkerhetstjänster
- Brandväggar för WebServices är andra djur än klassiska IP-adressfiltrerande brandväggar
 - Svårt - Tvingar nät/brandväggsadministratörerna att tolka affärslogiken
- Många lösningar nyttjar inte de säkerhetskoncept som finns framtagna:
 - autentisering, signering, etc

Initiativ för webbsäkerhet

Permanent tekniska skydd

Nätbaserad brandvägg Penetrationstester
Hostbaserad brandvägg
Nätbaserad IDS
Hostbaserad IDS
Nätbaserad IPS
Hostbaserad IPS
Reverseproxy
Härdat OS på värddator

Pärlor för svin?

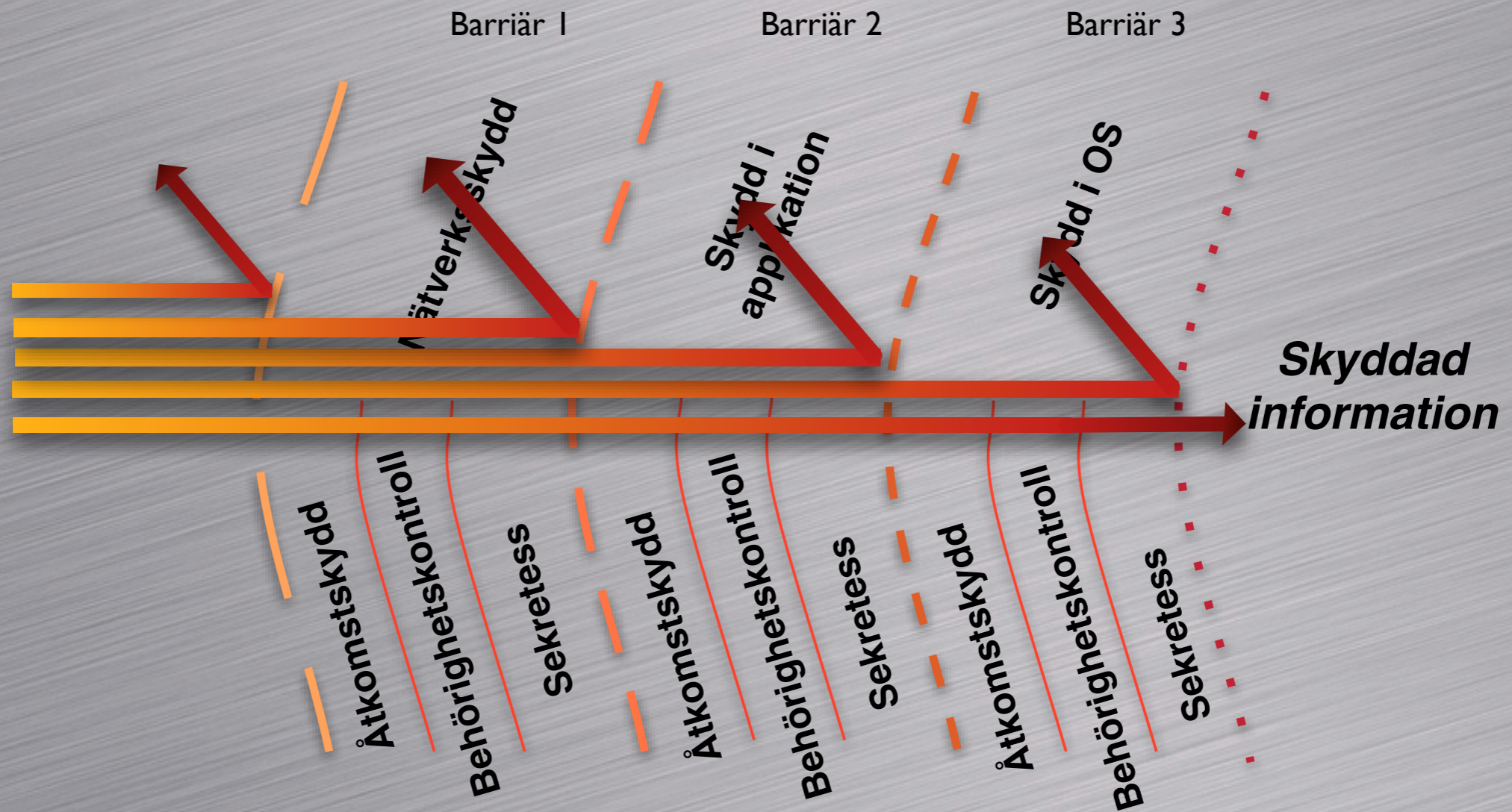
Investeringarna felaktiga alt. ofullständiga om vi **inte förstår tekniska förutsättningar**, vad som skall skyddas eller skyddsvärdet på det som skall skyddas

Permanent organisatoriska skydd

Ha genomtänkt informationsstruktur, test- och produktionssättningsförfarande
Inför en **BRA** utvecklingsmetodik / kräv att leverantörer kan påvisa att de nyttjar sådan

Microsoft Secure Development Lifecycle Model

Försvar i djupled



Seriöst dumma idéer

- Att lita på indata som kommer från något som en användare kan påverka
- Att använda samma dator till alltför många tjänster
- Att hyra en "hacker" för att testa säkerheten
- Att lägga ut information publikt på webben som egentligen skulle ligga på ett *skyddat och begränsat åtkomligt extranet*
- Slita på E-legitimation för än det ena, än det andra - ena dagen privat, andra dagen på jobbet, tredje dagen för en förenings räkning
- PIN-kodsbaserade inloggningar, medför såväl enkla uttömmande sökningar som DoS-attacker
- Hårdkodade lösenord i program och skript
- Att naivt tro att alla är snälla som läser filen robots.txt

Do's & Dont's

Do's

- Förstå riskerna med införande av nya lösningar
- KISS-principen
- Förstå databasen som nyttjas - använd fullt ut de säkerhetsmekanismer som ingår
- Använd mekanismer som finns i programspråk för att undvika säkerhetsproblem:
 - taint, sandlådor, etc
- Försvar på djupet

Don'ts

- Blanda inte utveckling, test och produktionsmiljöer
- Gör inte utveckling i produktionsmiljön
- Lägg inte ut information i tron att den **inte syns** bara för att den saknar länkar från webbsidor
- Undvik att *koda fast* dig i specifika versioner, tex av appservern
- Ge inte ut för mycket information i felmeddelanden

Do's & Dont's

Do's

LOGGAR!!!

Loggar behöver vara korrekta,
kompleta och användbara

Skydda den personliga integriteten

Don'ts

Lägg inte in säkerhetsfunktioner
i enbart klientsidan

Använd inte Pop-up-rutor med text
som ändå ingen läser....

Okontrollerad indexering och sökning
av de egna webbsidorna

Som användare på webben - lämna inte
integritetskänslig information

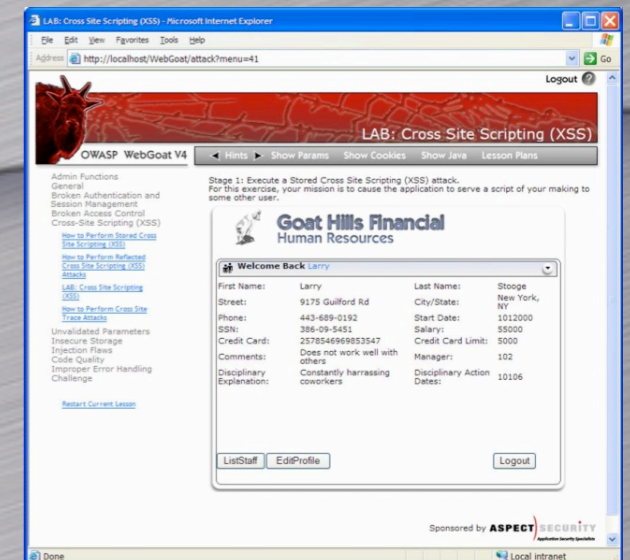
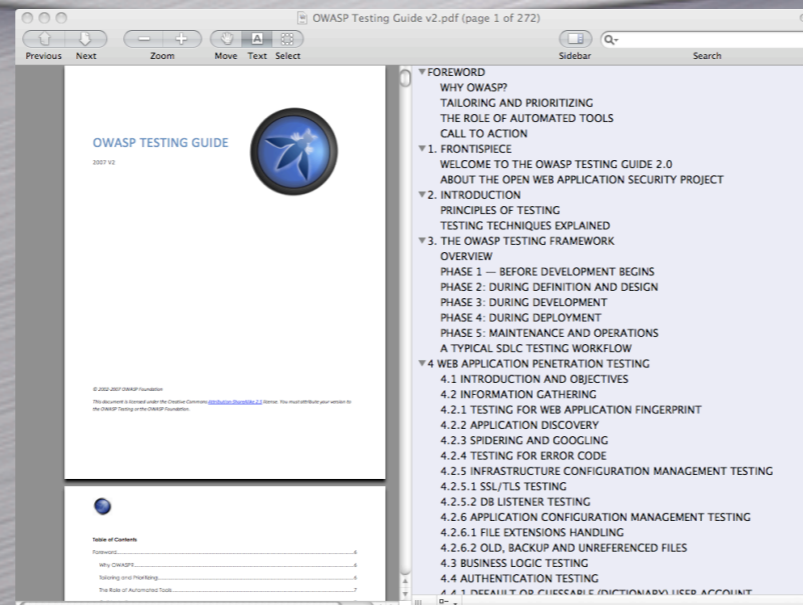
Kom ihåg: Internet glömmer inte

OWASP är din vän!

Open Web Application Security Project

ID	Category	Description
A1	Cross Site Scripting (XSS)	XSS flaws occur whenever an application takes user supplied data and sends it to a web browser without first validating or encoding that content. XSS allows attackers to execute script in the victim's browser which can hijack user sessions, deface web sites, possibly introduce worms, etc.
A2	Injection Flaws	Injection flaws, particularly SQL injection, are common in web applications. Injection occurs when user-supplied data is sent to an interpreter as part of a command or query. The attacker's hostile data tricks the interpreter into executing unintended commands or changing data.
A3	Malicious File Execution	Code vulnerable to remote file inclusion (RFI) allows attackers to include hostile code and data, resulting in devastating attacks, such as total server compromise. Malicious file execution attacks affect PHP, XML and any framework which accepts filenames or files from users.
A4	Insecure Direct Object Reference	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter. Attackers can manipulate those references to access other objects without authorization.
A5	Cross Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim's browser to send a pre-authenticated request to a vulnerable web application, which then forces the victim's browser to perform a hostile action to the benefit of the attacker. CSRF can be as powerful as the web application that it attacks.
A6	Information Leakage and Improper Error Handling	Applications can unintentionally leak information about their configuration, internal workings, or violate privacy through a variety of application problems. Attackers use this weakness to steal sensitive data, or conduct more serious attacks.
A7	Broken Authentication and Session Management	Account credentials and session tokens are often not properly protected. Attackers compromise passwords, keys, or authentication tokens to assume other users' identities.
A8	Insecure Cryptographic Storage	Web applications rarely use cryptographic functions properly to protect data and credentials. Attackers use weakly protected data to conduct identity theft and other crimes, such as credit card fraud.
A9	Insecure Communications	Applications frequently fail to encrypt network traffic when it is necessary to protect sensitive communications.
A10	Failure to Restrict URL Access	Frequently, an application only protects sensitive functionality by preventing the display of links or URLs to unauthorized users. Attackers can use this weakness to access and perform unauthorized operations by accessing those URLs directly.

Table 1: Top 10 Web application vulnerabilities for 2007



- BRA utvecklingsguider
- BRA testguider
- mängder med BRA testverktyg
- Exempel för utbildning
- BRA topplistor över angrepp
- www.owasp.org

WebScarab

File View Tools Help

Summary Message log Proxy Manual Request WebServices Spider Extensions SessionID Analysis Scripted Fragments Fuzzer Compare

Summary

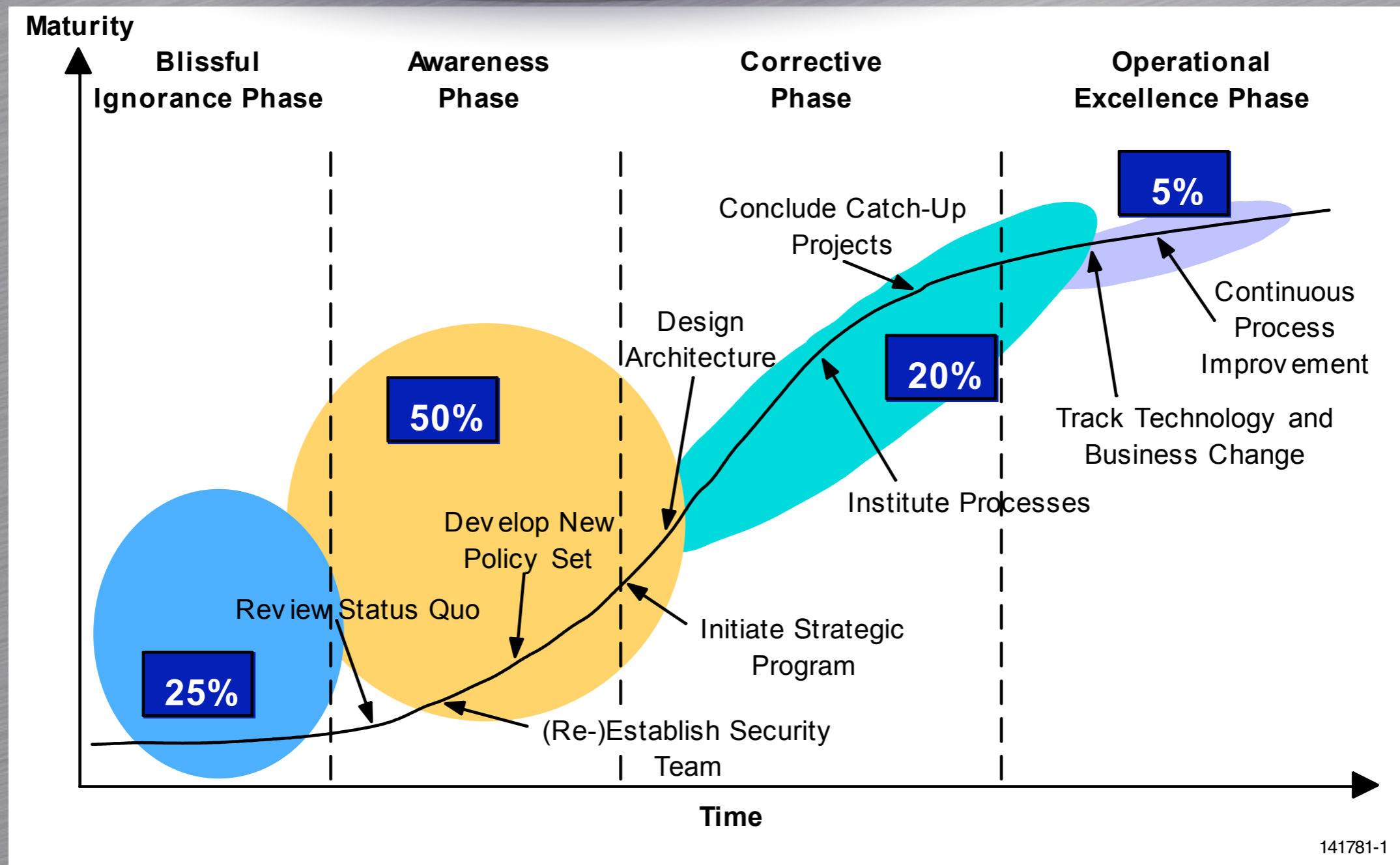
Tree Selection filters conversation list

Url	Methods	Status	Set-Cookie	Comments	Scripts
http://www.owasp.org:80/ banners/	GET	301 Moved ...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
http://www.owasp.org:80/ images/			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
http://www.owasp.org:80/ index.php/ Main_Page	GET	200 OK	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
http://www.owasp.org:80/ skins/			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

ID	Date	Method	Host	Path	Parameters	Status	Origin
5	2006/06/23...	GET	http://www.owasp.org:80	/skins/monobook/main....??		200 OK	Proxy
4	2006/06/23...	GET	http://www.owasp.org:80	/skins/common/IEFixes...		200 OK	Proxy
3	2006/06/23...	GET	http://www.owasp.org:80	/skins/common/commo...		200 OK	Proxy
2	2006/06/23...	GET	http://www.owasp.org:80	/index.php/Main_Page		200 OK	Proxy
1	2006/06/23...	GET	http://www.owasp.org:80	/		301 Moved ...	Proxy

5.27 / 63.56

Utvecklingskurvan för säkerhet

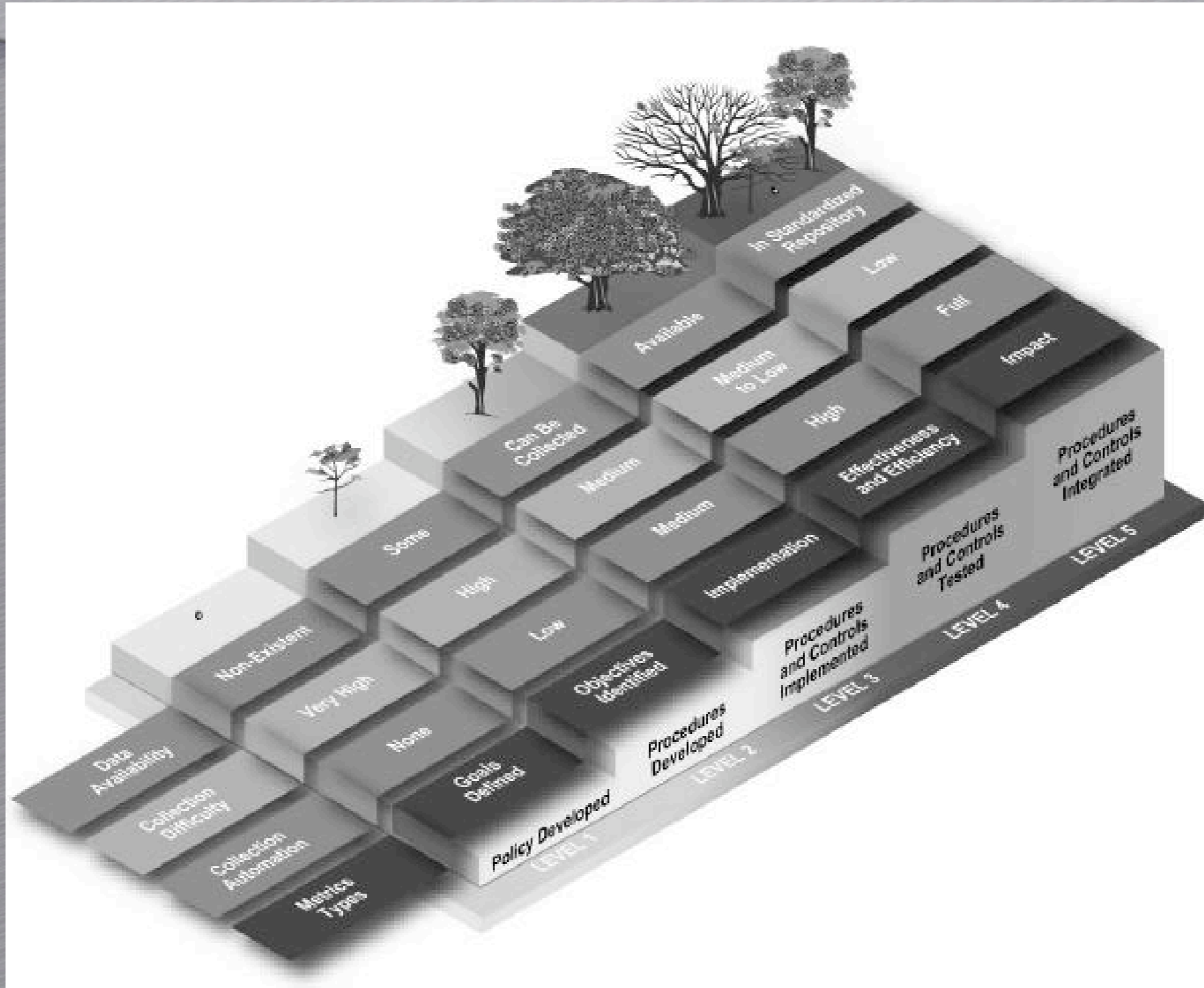


141781-1

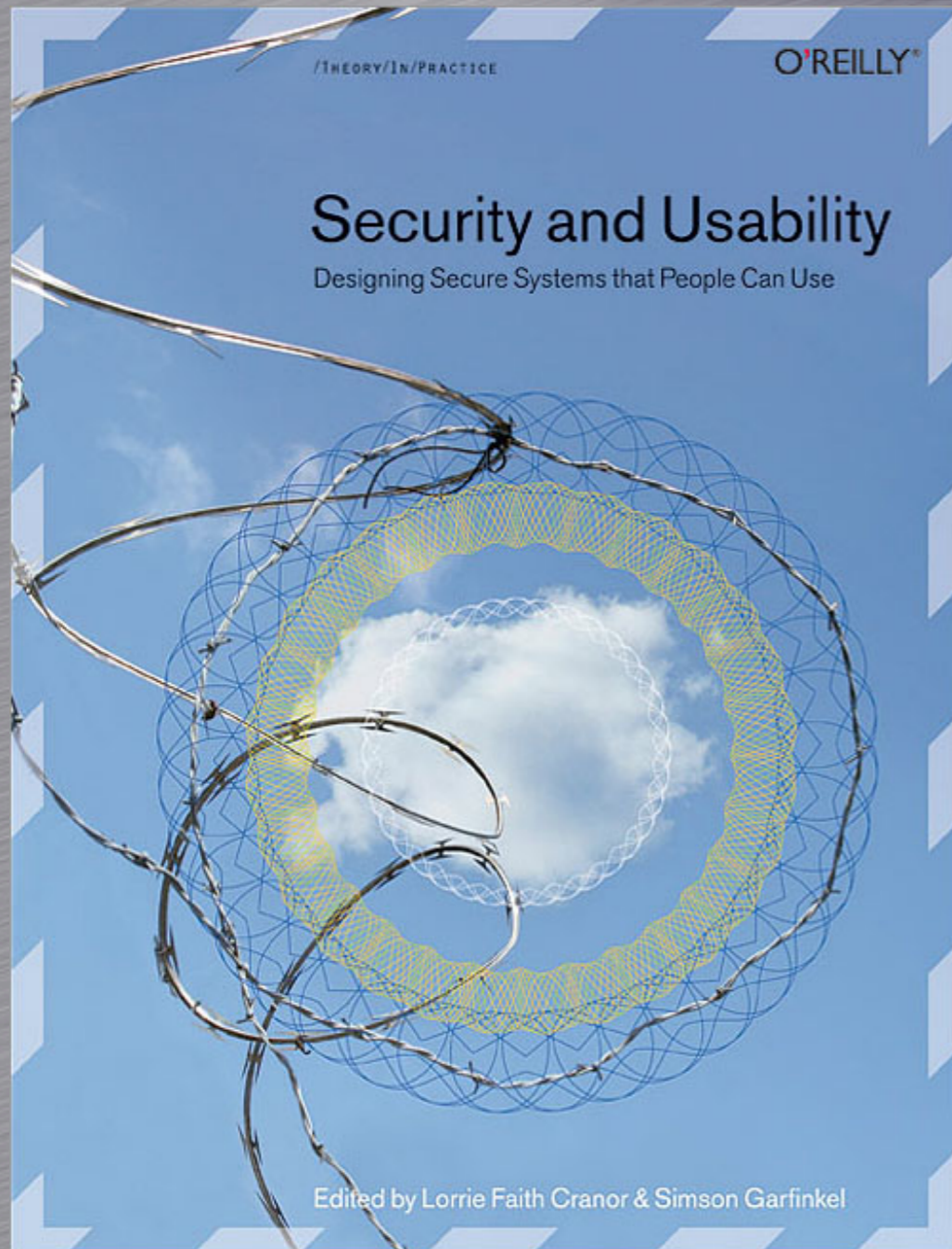
Note: The population distributions represent typical large G2000-type organizations.

Source: Gartner (July 2006)

Utvecklingskurvan för säkerhet 2



Apropå Gulans föredrag...



Psychological Acceptability Revisited

Design for Usability

Designing Systems That People Will Trust

Guidelines and Strategies for Secure Interaction Design

Fighting Phishing at the User Interface

Sanitization and Usability

Security Administration Tools and Practices

Privacy Issues and Human-Computer Interaction

A User-Centric Privacy Space Framework

Five Pitfalls in the Design for Privacy

Firefox and the Worry-Free Web

Users and Trust: A Microsoft Case Study

Apropå Gulans föredrag...

Secure Interaction Design

Ka-Ping Yee

with Norm Hardy, Mark Miller, Chip Morningstar,
Kragen Sitaker, Marc Stiegler, Dean Tribble, and Miriam Walker

Basic Concepts

ACTOR-ABILITY MODEL

At any point in time, the user's model contains a set of **actors** in the system and a set of **potential actions** for each actor.

For a system to be secure, the actual abilities of any actor must never come to exceed the bounds in the user model.

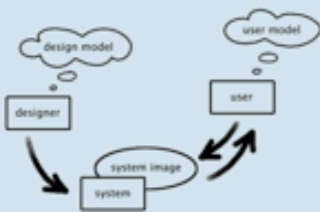
actors $A = \{A_0, A_1, \dots, A_n\}$
perceived abilities $P = \{P_0, P_1, \dots, P_n\}$
real abilities $R = \{R_0, R_1, \dots, R_n\}$

$$P_0 \subseteq R_0$$

$$P_i \supseteq R_i \text{ for } i > 0$$

SYSTEM IMAGE

The actors, actions, and objects in the user's mental model are derived from observing the **system image**, not from knowledge of its internal design.



USERS AND USER AGENTS

The software system intended to serve and protect the interests of the user is the **user agent**. On a stand-alone PC, this is the operating system shell, through which the user interacts with an arena of entities such as files and programs. On a networked PC, a second level of user agent represents the user's interests in a larger arena of interacting computers.

Fundamental Principles

Actor-Ability State

Input and Output

TRUSTED PATH

The interface must provide an unspoofable and faithful communication channel between the user and any entity trusted to manipulate authorities on the user's behalf.

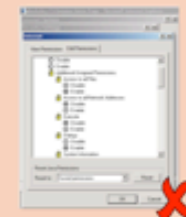


PATH OF LEAST RESISTANCE

The natural way to do any task should also be the secure way.

APPROPRIATE BOUNDARIES

The interface should expose distinctions between objects and between actions along boundaries that matter to the user.



VISIBILITY

The interface should allow the user to easily review any active authority relationships that would affect security-relevant decisions.

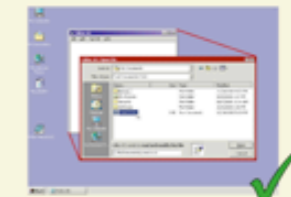


REVOCABILITY

The interface should allow the user to easily revoke authorities that the user has granted, wherever revocation is possible.

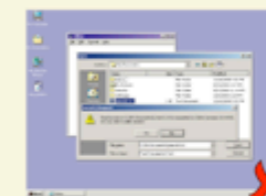
EXPLICIT AUTHORITY

A user's authorities must only be provided to other actors as a result of an explicit action that is understood by the user to imply granting.



EXPECTED ABILITY

The interface must not generate the impression that it is possible to do something that cannot actually be done.

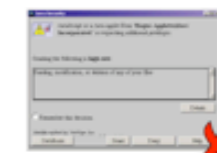


EXPRESSIVENESS

The interface should provide enough expressive power to (a) describe a safe security policy without undue difficulty and (b) allow users to express security policies in terms that fit their goals.

CLARITY

The effect of any security-relevant action must be clearly apparent to the user before the action is taken.



IDENTIFIABILITY

The interface should enforce that distinct objects and distinct actions have unspoofably identifiable and distinguishable representations.

Att vara förberedd...

It-krasch under terrorövning - IDG.se

http://www.studio.idg.se/17.108/2.1085/1.105300

Soekris on O...ng Diskless Positive Ath... Quotations Data Visuali... Approaches nwsmp - tech+bus+world newsmap del.icio.us ...ular treemap

En sajt i IDG.se-nätverket **ONSDAG** 21 november 2007

Ny Internetworld UTE NU! >>

ÖVERSIKT PRENUMERERA KUNDSERVICE ANNONSERA OM IDG

BLI MEDLEM LOGGA IN

NYHETER 360° | TIDNINGAR | TESTER | TIPS & GUIDER | JOBB & KARRIÄR | FRITID & NÖJE | VERKTYG & NYTTA | COMMUNITY | BUTIKEN | EVENTS

Nyheter efter intresseområde » IDG-TV NÄTVERK & TELEKOM PRYLAR MOBIL IT SERVER EXPERTSVAR VOIP VIRTUALISERING IDG-nyheter i RSS-format RSS

KRIS I KRISÖVNINGEN

2007-04-25 10:57 - Computer Sweden:

It-krasch under terrorövning

Av [Daniel Goldberg](#) | ComputerSweden

Samhälle (Uppdaterad) Ett it-haveri har satt käppar i hjulen för krisberedskapsmyndighetens, KBMs, stora terrorövning som pågår i Stockholm. Redan tidigt på morgonen kraschade KBMs webbplats, vilket har skapat stora problem för de inblandade myndigheterna.

ANNONS



FAKTA

Samverkansövning 2007

Fler än 40 myndigheter och upp till 4000 personer deltar i SAMÖ 2007 som sägs vara den största övningen i sitt slag någonsin i Sverige.

Övningen iscensätter en terrorattack mot Stockholms spårtrafik med efterföljande spridning av radioaktiva ämnen. Krisberedskapsmyndighetens roll är att samordna och planera övningen. Syftet är att utvärdera och stärka Sveriges beredskap vid terrorattentat.

Bland de medverkande myndigheterna märks Försvaret, Smittskyddsinstitutet och Räddningsverket. Övningen pågår mellan den 24 och 26 april i Stockholm, men utspelas enbart dagtid.

LÄS MER

▶ [Allt om haverier och systemfel på CS samlingssida](#)

Skriv ut Tipsa Kommentera

Spara till del.icio.us Dela på Facebook

Bloggkommentarer till artikeln [Twingly](#) Twingly bloggsök

2 kommentarer, visar de 2 mest länkade. [Visa alla.](#)

IT-systemet kraschade under beredskapsövning ANMÄL

on 25 apr 2007 - iterationer

Ironi 7 ANMÄL

lö 28 apr 2007 - Till yttermera visso

NYHETS BREV

Heta nyheter till din mejlbox

Få vårt nyhetsbrev med de senaste it-nyheterna, direkt i din inkorg varje morgon! Nu också med dagens Dilbert.

E-postadress

Registrera Avregistrera

ANNONS



Samverkansövningen SAMÖ 2007 är en av de största i sitt slag någonsin i Sverige och iscensätter en terrorattack mot Stockholms spårtrafik.

Men redan tidigt på onsdagsmorgonen kraschade KBMs webbplats, troligen på grund av en kraftig trafikökning.

FRÅN IDG



Slutsatser (1)

- Det finns ingen magisk silverkula
- "JavaScript is the new Shellcode"
- WebServices kan leda till exponering av programåtkomst
- Många lösningar skapar cirkelberoenden mellan komponenter, vilket omöjliggör patchning och uppgradering
- Sårbarheter finns överallt, inte minst i plattformen man nyttjar, tex *XML Core components*, PHP, Java, Ruby, WAS, IAS, Apache, IIS, etc
- Antagonistfaktorn och programmerade hot utelämnas eller misförstås i riskanalyser

Slutsatser (2)

- ❑ Angrepp sker numera ofta från server till klient, efter det att en intern användare "bjudit in".
- ❑ Spårbarhet är en av de grundläggande säkerhetsfunktionerna. Spårbarhet (i form av loggar) är i praktiken ofta **helt oanvändbara** när de behövs som mest...
- ❑ Undvik att bli en tjänst som nyttjas för att skada tredje part: SMS-bombning, phishing,
- ❑ Tänk på det juridiska ansvaret som ändå finns: PuL, Lag om ansvar för elektroniska anslagstavlor, mm, mm
- ❑ **Att ligga i teknikens framkant medför per automatik större risker: oprövad teknik, oupptäckta säkerhetshål, etc**



"Sir, now might be a good time to bring up our security issues. The guards are all paper cutouts, and the swords and spears are made out of rubber."