

# Tehnologii Web



reddit.com/r/ProgrammerHumor

Username : admin

Password : admin

**securitatea aplicațiilor Web**  
o prezentare generală

„Experiența este acel minunat lucru  
care îți dă voie să recunoști o greșeală  
pe care ai mai făcut-o.”

**F.P. Jones**

# Ce înseamnă securitatea datelor?

# securitatea datelor

**Securitatea** este procesul de **menținere** a unui nivel acceptabil de risc perceptibil

# securitatea datelor

**Securitatea** este procesul de **menținere** a unui nivel acceptabil de risc perceptibil

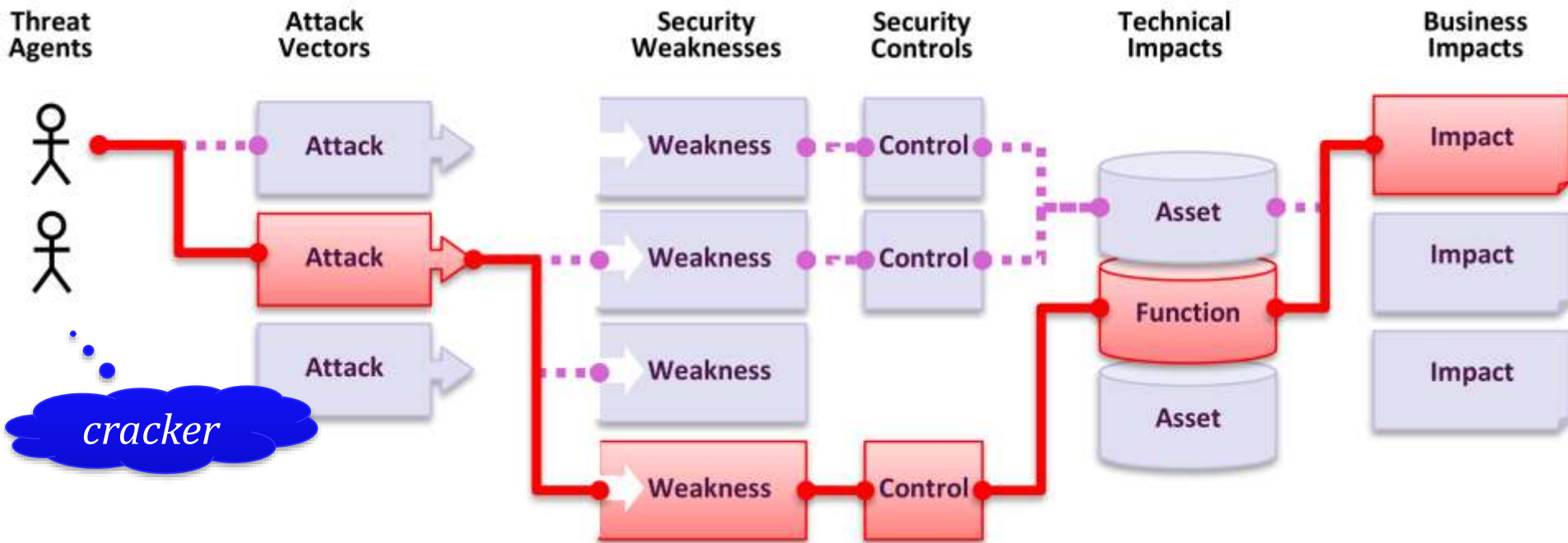
*“Security is a process, not an end state.”*

Mitch Kabay

*An Introduction to Information Security* (NIST, 2017)

[www.nist.gov/publications/introduction-information-security](http://www.nist.gov/publications/introduction-information-security)

# securitatea datelor



riscuri de securitate (*Web application security risks*)  
conform **OWASP** – *Open Web Application Security Project*  
[www.owasp.org](http://www.owasp.org)

# securitatea datelor

Confidențialitatea

Autentificarea

Autorizarea

Integritatea

Nerepudierea

Intimitatea (*privacy*)

Disponibilitatea

# securitatea datelor

## Confidențialitatea

imposibilitatea unei terțe entități să aibă acces  
la datele vehiculate între doi receptori



# securitatea datelor

## Confidențialitatea

soluție:

conexiuni private între cele 2 puncte terminale  
ale canalului de comunicație

datele circulă printr-un tunel oferit de o rețea privată  
virtuală (VPN – *Virtual Private Network*)

de studiat [www.ssh.com/ssh/tunneling/](http://www.ssh.com/ssh/tunneling/)

# securitatea datelor

## Confidențialitatea

### HTTPS (*HyperText Transfer Protocol Secure*)

scop: criptare bidirecțională + autentificare „sigură”,  
prevenind atacuri de tip *man-in-the-middle* și  
interceptare/alterare de date (*eavesdropping, tampering*)

RFC 7230

# securitatea datelor

## Confidențialitatea

**HTTPS** (*HyperText Transfer Protocol Secure*)

*HTTP over TLS (Transport Layer Security)*

URL-urile folosesc schema **https** – port standard: **443**

studiu de caz: *HTTPS on Stack Overflow* (2017)

[nickcraver.com/blog/2017/05/22/https-on-stack-overflow/](https://nickcraver.com/blog/2017/05/22/https-on-stack-overflow/)

# securitatea datelor

## Confidențialitatea

soluție:

criptarea datelor via diverse abordări (algoritmi)  
o introducere practică la [www.crypto101.io](http://www.crypto101.io)

cadrul general: **Web Cryptography API**  
(recomandare a Consorțiului Web, 2017)  
[www.w3.org/TR/WebCryptoAPI/](http://www.w3.org/TR/WebCryptoAPI/)

# securitatea datelor

Exemplificări de soluții criptografice – biblioteci specializate și/sau oferite de mediile de dezvoltare Web:

OpenSSL (bibliotecă C; numeroase portări) – [openssl.org](https://openssl.org)

Java Cryptography Architecture

Forge (JavaScript) – [github.com/digitalbazaar/forge](https://github.com/digitalbazaar/forge)

System.Security.Cryptography (.NET)

crypto (Node.js) – [www.npmjs.com/package/crypto-js](https://www.npmjs.com/package/crypto-js)

Mcrypt, phpseclib, Zend Framework Encryption (PHP)

Cryptography Toolkit (Python) – [www.pycrypto.org](https://www.pycrypto.org)

mai multe la [github.com/sobolevn/awesome-cryptography](https://github.com/sobolevn/awesome-cryptography)

# securitatea datelor

## Confidențialitatea

atenție: exploatarea vulnerabilităților bibliotecilor

exemplu (2014): **heartbleed**

slăbiciune majoră a bibliotecii *open-source* OpenSSL

[heartbleed.com](http://heartbleed.com)

exemplificare (2015): **FREAK**

se baza pe vulnerabilități TLS ale *browser*-ului

[censys.io/blog/freak](http://censys.io/blog/freak)

# securitatea datelor

## Autentificarea

mecanism ce permite utilizatorilor să acceseze  
un serviciu după verificarea identității  
utilizatorului – uzual, pe bază de nume + parolă

# securitatea datelor

## Autentificarea

soluție:

serverul Web oferă suport pentru  
autentificări de bază (*basic authentication*)  
sau bazate pe algoritmi de tip *digest* (*hash*)  
– e.g., SHA-2 (SHA-256, SHA-512 etc.), SHA-3  
[csrc.nist.gov/projects/hash-functions](https://csrc.nist.gov/projects/hash-functions)



# securitatea datelor

## Autentificarea

exemplificări:

`mod_auth_basic`, `mod_auth_digest`, `mod_authn_dbd`, ...  
(module Apache)

[httpd.apache.org/docs/howto/auth.html](http://httpd.apache.org/docs/howto/auth.html)

`ngx_http_auth_basic_module`, `ngx_http_auth_request_module`  
(module Nginx)


pentru alte soluții, de vizitat [wiki.nginx.org/Modules](http://wiki.nginx.org/Modules)

# securitatea datelor

## Autentificarea

soluție:

folosirea/implementarea unor servicii de autentificare  
de exemplu, **OpenID Connect**



vezi unul dintre  
cursurile anterioare

pentru utilizatori umani, de recurs la autentificare  
multi-factor – e.g., *Two Factor Auth* (2FA)

# securitatea datelor

## Autorizarea

specifică acțiunile (rolurile) pe care un utilizator  
ori o aplicație a utilizatorului  
le poate realiza într-un anumit context

# securitatea datelor

## Autorizarea

specifică acțiunile (rolurile) pe care un utilizator  
ori o aplicație a utilizatorului  
le poate realiza într-un anumit context

asociată autentificării

permite definirea politicilor de control al accesului  
la servicii (funcționalități)

# securitatea datelor

## Autorizarea

soluții:

drepturi de acces (permisiuni)

+

liste de control al accesului (*ACL – Access Control List*)

context: autorizarea accesului la datele disponibile

în cadrul unei aplicații Web – *e.g.*, via OAuth

[www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2](http://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2)

# securitatea datelor

## Autorizarea

soluții:

controlul accesului bazat pe roluri  
(RBAC – *Role-Based Access Control*)

exemplu:

un utilizator obișnuit cu rol de administrator  
într-o situație specifică

# securitatea datelor

## Integritatea

în acest context, implică detectarea încercărilor  
de modificare neautorizată (*tampering*)  
a datelor transmise

# securitatea datelor

## Integritatea

soluții:

algoritmi de tip *digest*

semnături digitale

(stocate, eventual, în format XML – *XML Signature*)  
pot fi vehiculate și via mesaje SOAP



# securitatea datelor

## Nerepudierea

asigură faptul că expeditorul unui mesaj  
nu poate afirma că nu l-a trimis

# securitatea datelor

Nerepudierea

soluție:

certificate digitale

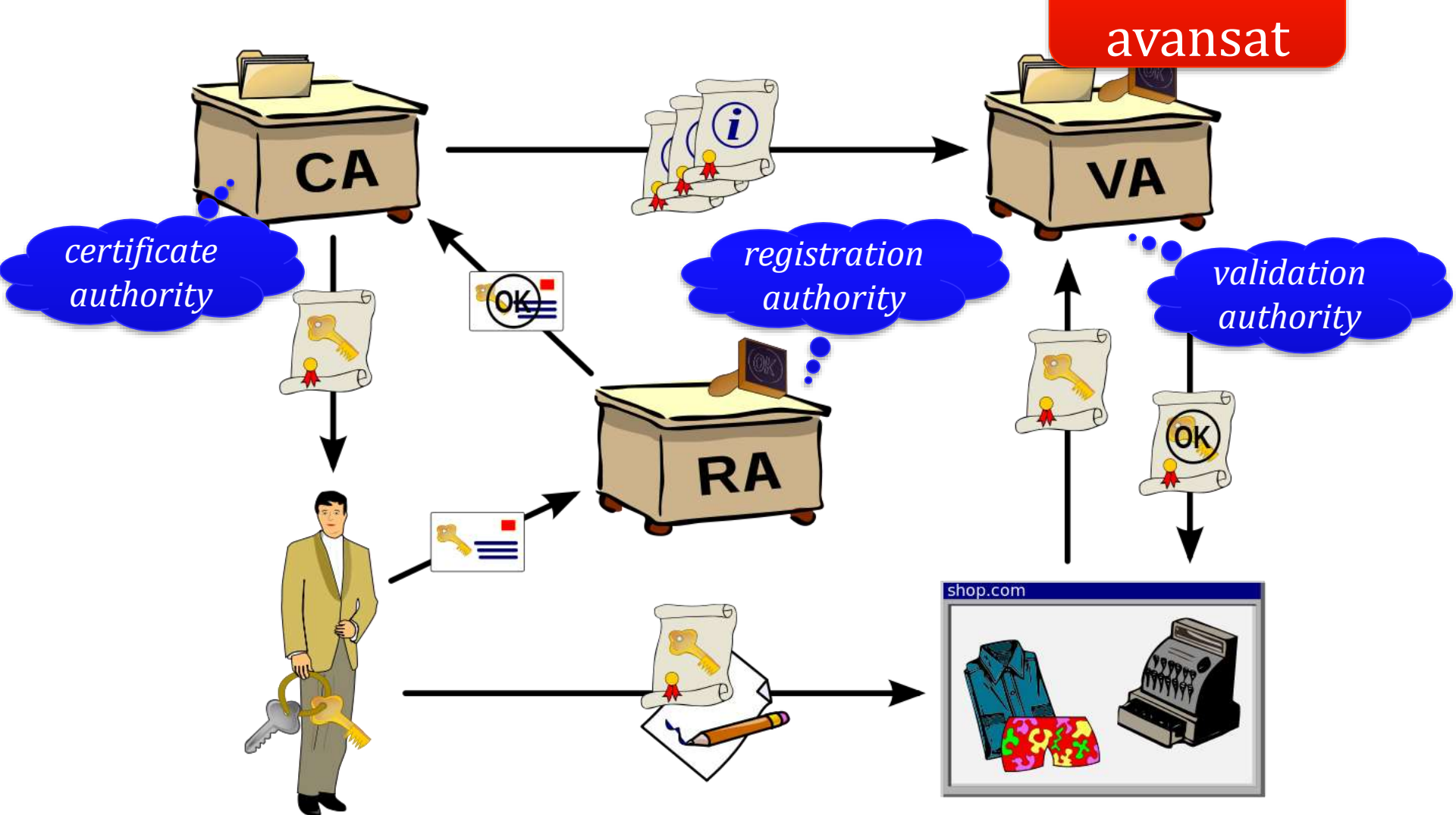
stochează date privind identitatea  
unei entități deținătoare a unui secret:  
parolă, serie a cărții de credit, certificat digital,...

# securitatea datelor

PKI (*Public Key Infrastructure*)  
infrastructura bazată de chei publice

set de resurse hardware, software, umane + politici și  
proceduri pentru managementul certificatelor digitale  
(creare, distribuție, utilizare, stocare, revocare)

la nivel de Web, de considerat specificația  
*Web Authentication: An API for accessing  
Public Key Credentials* (*W3C Recommendation*, 2019)  
[www.w3.org/TR/webauthn/](http://www.w3.org/TR/webauthn/)



[www.herongyang.com/PKI/](http://www.herongyang.com/PKI/)

PKI permite utilizatorilor să comunice „sigur” într-o rețea publică nesigură, inclusiv verificând identitatea unui utilizator via certificate digitale emise de o autoritate

# securitatea datelor

## Disponibilitatea

necesitatea ca o anumită resursă  
să poată fi accesată la momentul oportun

# securitatea datelor












## Disponibilitatea

necesitatea ca o anumită resursă  
să poată fi accesată la momentul oportun

aspect de interes: **calitatea unui serviciu**  
stipulată via **SLA** (*Service-Level Agreement*)

*uptime, average speed to answer, turn-around time,  
abandonment rate, mean time to recover,...*

**37.7%** (213/565) endpoints are **available**

	SPARQL Endpoint ▲ ▼	Uptime Last 24h ▲	Uptime Last 7 days
	Kidney and Urinary Pathway Knowledge Base	100%	24.4%
	OntoBeef	100%	78.11%
	demografiaataun	100%	94.05%
	Bio2RDF::Mesh	100%	96.45%
	Bio2RDF::Genage	100%	97.02%
	Bio2RDF::Interpro	100%	97.04%
	Bio2RDF::Pharmgkb	100%	97.63%
	DBpedia-Live	100%	98.22%
	Lista de Encabezamientos de Materia as Linked Open Data	100%	98.22%
	Spanish Linguistic Datasets	100%	98.22%
	Terminesp Linked Data	100%	98.22%
	AEMET metereological dataset	100%	98.82%
	Bio2RDF::Pubmed	100%	98.82%
	CRTM	100%	98.82%
	CulturaLinkedData	100%	98.82%

gradul de disponibilitate a unor servicii Web

# securitatea datelor

## Disponibilitatea

cauze ale indisponibilității:

atacuri de refuz al serviciilor DoS (*Denial of Service*)

atacuri distribuite de tip DDoS (*Distributed DoS*)

implementare precară



# securitatea datelor

## Intimitatea

vizează drepturile ce trebuie respectate privind  
caracterul (subiectul) datelor vehiculate

confundată, deseori, cu confidențialitatea

EPIC (*Electronic Privacy Information Center*) – [www.epic.org](http://www.epic.org)

EU GDPR (*General Data Protection Regulation*) – [eugdpr.org](http://eugdpr.org)

# securitatea datelor

## Intimitatea

breșe:

stocarea necorespunzătoare a datelor  
la nivel de server – *information disclosure*

atacuri de tip XSS (*Cross-Site Scripting*)

atacuri de tip *phishing* – [www.honeynet.org/papers/phishing/](http://www.honeynet.org/papers/phishing/)

configurarea neadecvată a sistemelor

# securitatea datelor

Securitatea Web trebuie să ia în considerație:

## clientul

interacțiunea cu utilizatorul

date personale stocate: *cookie*-uri, date *off-line*, *cache*,...

transferurile asincrone – Ajax/WebSockets

rularea (neautorizată) a programelor JavaScript

existența *plugin*-urilor/extensiilor suspecte

...

# securitatea datelor

Securitatea Web trebuie să ia în considerație:

## datele aflate în tranzit

securitatea rețelei (cu/fără fir)

schimbul sigur de mesaje între diverse entități

nerepudiarea datelor

...

# securitatea datelor

Securitatea Web trebuie să ia în considerație:

## serverul

securitatea serverului/serverelor Web

securitatea aplicațiilor, *framework*-urilor, bibliotecilor,...

disponibilitatea serviciilor oferite

# securitatea datelor

Securitatea Web trebuie să ia în considerație:

**clientul**

**datele aflate în tranzit**

**serverul**

**atacurile pot viza oricare din cele 3 aspecte!**

# securitatea datelor

## Vulnerabilități

slăbiciuni ale unui sistem hardware/software ce permit utilizatorilor neautorizați să aibă acces asupra lui

pot apărea și datorită unei administrări precare

# securitatea datelor

## Vulnerabilități

**niciun sistem nu este 100% sigur**



Cum are loc un atac privind securitatea?

# atacuri

## Examinarea mediului

identificarea porturilor/serviciilor publice

descoperirea tipurilor + versiunilor aplicațiilor

generarea de erori + examinarea mesajelor obținute

găsirea de informații sensibile:

cod-sursă, comentarii, câmpuri ascunse ale formularelor,...

i	<a href="#">Office 365 Mail</a> Business Email Hosting	Aug 2016	May 2019
i	<a href="#">MailChimp SPF</a> Campaign Management	inspectarea tehnologiilor folosite de o aplicație Web: <b>BuiltWith</b>	
SSL Certificates			
i	<a href="#">Comodo PositiveSSL</a>	May 2015	May 2019
i	<a href="#">Comodo PositiveSSL Wildcard</a> Wildcard	May 2015	May 2019
i	<a href="#">Comodo SSL</a> Root Authority	May 2015	May 2019
i	<a href="#">SSL by Default</a>	Jun 2017	May 2019
Web Servers			
i	<a href="#">nginx</a>	Jul 2012	May 2019
i	<a href="#">Varnish</a> Caching Proxy	Jul 2012	Apr 2019

# atacuri

Stabilirea țintei atacului

mecanismul de autentificare (*login*)

câmpurile formularelor Web

managementul sesiunilor

infrastructura folosită – serverele de stocare a datelor,  
serviciile adiționale (*e.g., proxy*),...

# atacuri

## La nivel de HTTP

analizarea pachetelor de date (*network sniffing*):  
funcționează pentru fluxuri de date HTTP necriptate

o soluție de prevenire: HTTPS

# atacuri

## La nivel de HTTP

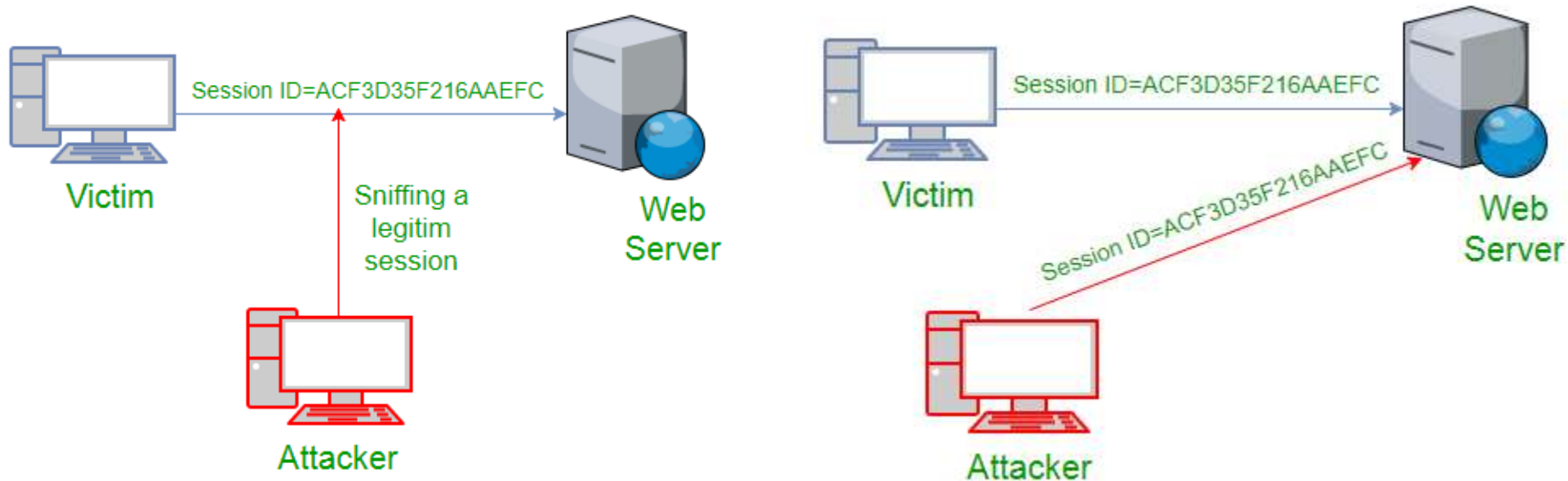
deturnarea sesiunilor (*session hijacking*):

atacatorul – *man-in-the-middle* – determină SID-ul utilizatorului și îl folosește în scop propriu

exemplu: analizarea câmpului **Referer**

**Referer:** [https://www.ebank.info/view/account?id=98151  
&jssid=BAC13606AC22B81E5137F45F95EE7573](https://www.ebank.info/view/account?id=98151&jssid=BAC13606AC22B81E5137F45F95EE7573)

caz real: reutilizarea sesiunii editând *cookie*-ul asociat  
Trello (23 mai 2018) – [hackerone.com/reports/352732](https://hackerone.com/reports/352732)



tehnici de preluare a informațiilor despre sesiunile Web:

- interceptarea mesajelor HTTP (*packet sniffing*)
- atacuri de tip XSS (*cross-site scripting*)
- ghicirea identificatorului de sesiune – SID (*blind attack*)

# atacuri

## La nivel de HTTP

deturnarea sesiunilor (*session hijacking*)

soluții clasice de prevenire:  
eliminarea SID-ului din URL

stocarea SID-ului în câmpul **User-Agent**

utilizarea unui algoritm de generare a unui SID/*token*  
având o valoare impredictibilă

utilizarea unui SID variabil



# atacuri

## La nivel de HTTP

folosirea codului de stare HTTP pentru a expune date

detalii în Mike Cardwell, *Abusing HTTP Status Codes to Expose Private Information* (2011)

[www.grepular.com/Abusing\\_HTTP\\_Status\\_Codes\\_to\\_Expose\\_Private\\_Information](http://www.grepular.com/Abusing_HTTP_Status_Codes_to_Expose_Private_Information)

# atacuri

## *Server Side Request Forgery (SSRF)*

abuz asupra funcționalității unui server Web  
pentru a accesa sau altera resurse interne

pe baza unui URL, atacatorul poate modifica parametri  
utilizați de o aplicație pentru a crea cereri malițioase

*modus operandi* + soluții de contracarare:

[www.acunetix.com/blog/articles/server-side-request-forgery-vulnerability/](http://www.acunetix.com/blog/articles/server-side-request-forgery-vulnerability/)

[www.netsparker.com/blog/web-security/server-side-request-forgery-vulnerability-ssrf/](http://www.netsparker.com/blog/web-security/server-side-request-forgery-vulnerability-ssrf/)

# atacuri

## *SQL injection*

presupune scrierea unor interogări SQL care permit afișarea, alterarea, ștergerea de date din baze de date via formulare Web ori direct, folosind URL-uri

pentru detalii, a se consulta *Testing for SQL Injection*:  
[www.owasp.org/index.php/Testing\\_for\\_SQL\\_Injection\\_%28OTG-INPVAL-005%29](http://www.owasp.org/index.php/Testing_for_SQL_Injection_%28OTG-INPVAL-005%29)  
cazuri reale: [laurent22.github.io/so-injections/](https://laurent22.github.io/so-injections/)

# atacuri

*SQL injection* – exemplu:

**select \* from customers where name=\$name and pass=\$pass**

cu **\$name** preluat din formular având valoarea " **or 1=1 --**

# atacuri

*SQL injection* – exemplu:

[http://e-banking.org/access\\_client.php?client=3](http://e-banking.org/access_client.php?client=3)

în *script*: **select credit\_card from clients where client=\$client**

# atacuri

*SQL injection* – exemplu:

[http://e-banking.org/access\\_client.php?client=3](http://e-banking.org/access_client.php?client=3)

în *script*: **select credit\_card from clients where client=\$client**

ce se întâmplă dacă URL-ul este

[http://www.sit.org/access\\_client.php?client=client](http://www.sit.org/access_client.php?client=client) ?

dar dacă în loc de **select** apărea comanda **delete** ?

# atacuri

## *SQL injection*

variații:

crearea de interogări SQL incorecte  
pentru a avea acces la mesaje de eroare „interesante”

# atacuri

*SQL injection* – exemplu:  
[www.web.info/search?id=1+OR+xy=1](http://www.web.info/search?id=1+OR+xy=1)

se poate obține un mesaj precum:

```
[Microsoft][ODBC SQL Server Driver] [SQL Server] Invalid column name 'xy'.  
SELECT group_id, securityName, maxSalesCharge, price,  
security_id, trade_date FROM funds  
WHERE group_id = 1 OR xy=1 ORDER BY price DESC
```



# atacuri

*SQL injection* – exemplu:

[www.web.info/search?id=1+OR+xy=1](http://www.web.info/search?id=1+OR+xy=1)

se poate obține un mesaj precum:

```
[Microsoft][ODBC SQL Server Driver] [SQL Server] Invalid column name 'xy'.  
SELECT group_id, securityName, maxSalesCharge, price,  
security_id, trade_date FROM funds  
WHERE group_id = 1 OR xy=1 ORDER BY price DESC
```

atacatorul poate continua – de pildă – cu:

[www.web.info/search?id=1;DELETE+FROM+funds+--](http://www.web.info/search?id=1;DELETE+FROM+funds+--)

# atacuri

## *SQL injection*

soluții de prevenire:

„neutralizarea” meta-caracterelor SQL,  
*prepared statements*, utilizarea de *framework-uri* ORM  
(*Object-Relational Mapping*), proceduri stocate,...

incorect

```
$sql = "select * from users  
where user = " . $user . "";
```

corect

```
$rezultat = $db.query  
("select * from users  
where user = ?", $user);
```

# atacuri

## *SQL injection*

soluții de testare a vulnerabilităților (*penetration tools*):

Safe3 SQL Injector – [sourceforge.net/projects/safe3si/](https://sourceforge.net/projects/safe3si/)

sqlmap – [sqlmap.org](https://sqlmap.org)

SQL Ninja – [sqlninja.sourceforge.net](https://sqlninja.sourceforge.net)

detalii la [www.owasp.org/index.php/Blind\\_SQL\\_Injection](https://www.owasp.org/index.php/Blind_SQL_Injection)

# atacuri

## *NoSQL injection*

exploatarea limbajului de programare disponibil în cadrul serverului NoSQL, inclusiv slăbiciunile API-ului oferit și/sau formatul de transfer al datelor (JSON, XML)

exemplificare: *Hacking Node.js and MongoDB* (2014)  
[blog.websecurify.com/2014/08/hacking-nodejs-and-mongodb.html](http://blog.websecurify.com/2014/08/hacking-nodejs-and-mongodb.html)

pentru amănunte, a se parcurge  
[www.owasp.org/index.php/Testing\\_for\\_NoSQL\\_injection](http://www.owasp.org/index.php/Testing_for_NoSQL_injection)

# atacuri

## *Shell command injection*

rularea de comenzi externe via *script*-uri CGI sau din cadrul serverelor de aplicații Web (PHP, Python, Ruby)

soluție de prevenire:  
inhibarea folosirii funcțiilor `system ()`, `exec ()` etc.

# atacuri

## *SQL injection + command injection*

utilizarea SQL pentru execuția la nivel de *shell* de comenzi din cadrul serverului de baze de date

exemplu:

```
SELECT * FROM users WHERE name = 'tuxy' AND  
pass = ' '; xp_cmdshell 'taskkill /F /IM sqlservr.exe' --'
```

# atacuri

## *XPath injection*

recurgerea la expresii XPath pentru acces la date  
într-un document XML sau pentru a realiza  
diverse acțiuni via funcții XPath

consecințe și asupra transformărilor XSLT  
considerate maligne ► pot cauza, de exemplu, DoS  
detalii la [www.agarri.fr/blog/](http://www.agarri.fr/blog/)

# atacuri

## *Path traversal*

posibilitatea de accesare a unor zone nepermise ale sistemului de fișiere – *i.e.*, în afara directoarelor în care rezidă aplicația Web

exemplificare:

**[e-photos.info/photos/list.jsp?dir=../../../../](#)**

exemplu real: Cisco ASA (20 mai 2019): **[hackerone.com/reports/378698](#)**

alt caz: Node.js modules (1 apr. 2019): **[hackerone.com/reports/510043](#)**



# atacuri

## *Path traversal*

posibilitatea de accesare a unor zone nepermise ale sistemului de fișiere – *i.e.*, în afara directoarelor în care rezidă aplicația Web

exemplu în contextul XML (**XXE** – *XML External Entity*):  
[cwe.mitre.org/data/definitions/611.html](https://cwe.mitre.org/data/definitions/611.html)

```
<!DOCTYPE doc [ <!ENTITY xxe SYSTEM "file:///tmp/sessions/..."> ]>
```

# atacuri

Exemplificare reală – atac asupra PostgreSQL

conectare cu privilegii reduse

preluare global/pg\_auth prin XXE

suprascrierea acestui fișier via XSLT

re-conectare cu privilegii de administrator

restaurare global/pg\_auth via XSLT

lansare postgres\_payload.rb – resursă oferită de proiectul

**Metasploit:** [www.metasploit.com](http://www.metasploit.com)

# atacuri

## *Poisonous null-byte attack*

folosirea caracterului NULL pentru plasarea de *script*-uri pe server ce ulterior pot fi executate

exemplu:

*upload*-ul unei „imagini” – [img.php%00.jpg](#)

*“Thank you! See your picture at [img.php](#)”*

# atacuri

## *Cross-Site Scripting (XSS)*

permite „injectarea” în cadrul sistemului,  
pentru execuția direct în *browser*,  
a programelor JavaScript

funcționează mai ales în cadrul siturilor Web interactive  
(*e.g.*, forumuri, *blog*-uri, *wiki*-uri)

tutorial: [www.hacker101.com/sessions/xss](http://www.hacker101.com/sessions/xss)

# atacuri

## *Stored XSS*

atacatorul injectează un program JS (denumit și *payload*) care e stocat permanent în cadrul aplicației țintă

*e.g.*, în baza de date a aplicației Web de tip CMS (*Content Management System*)

exemplificări:

Steam Community (25 mai 2018): [hackerone.com/reports/351171](https://hackerone.com/reports/351171)

Starbucks (23 mai 2018): [hackerone.com/reports/227486](https://hackerone.com/reports/227486)

# atacuri

## *Reflected XSS*

*script*-ul de tip *payload* este transmis de serverul Web al atacatorului ca parte a unui mesaj de răspuns HTTP (codul malițios e livrat de la distanță fiecărei victime)

utilizatorul e persuadat să viziteze un URL special via tehnici de *social engineering* (*e-mail*, rețele sociale,...)

*Reflected XSS explained* (2018): [blog.sqreen.io/reflected-xss/](https://blog.sqreen.io/reflected-xss/)  
exemplu: OLX (decembrie 2018): [hackerone.com/reports/429647](https://hackerone.com/reports/429647)

# atacuri

## *DOM-based XSS*

*payload*-ul este stocat – în urma unei manipulări ilegale (referențiere și utilizare) a codului JS – în arborele DOM disponibil la nivel de *browser*

detalii în articolul lui Ferruh Mavituna (2017)

[www.netsparker.com/blog/web-security/dom-based-cross-site-scripting-vulnerability/](http://www.netsparker.com/blog/web-security/dom-based-cross-site-scripting-vulnerability/)

caz real: ZEIT (16 mai 2019): [hackerone.com/reports/545121](https://hackerone.com/reports/545121)

# atacuri

XSS – exemple tipice:

``

redirecționează utilizatorul spre alt URL,  
preia valori de *cookie*-uri ori blochează *browser*-ul


includerea de cod malițios (*malware*)  
spre a fi executat la nivel de *browser*  
via elemente precum `<iframe>`, `<img>` sau `<video>`



# atacuri

XSS – alte acțiuni malefice:

```
<script type="text/javascript">  
setInterval (function () {  
  var w = window.open ();  
  w.document.write (document.documentElement.outerHTML ||  
    document.documentElement.innerHTML);  
  }, 33);  
</script>
```



crearea recursivă  
de ferestre via DOM  
(à la *fork bomb*)

# atacuri

**XSS** – alte acțiuni malefice:

plasarea de programe *malware*  
în cadrul unei aplicații Web – *e.g.*, cod jQuery fals

studii de caz – cel mai recent descris pe 20 mai 2019:  
[blog.sucuri.net/category/website-malware-infections](http://blog.sucuri.net/category/website-malware-infections)

# atacuri

## *Cross-Site Request Forgery (CSRF)*

forțează utilizatorul autentificat în cadrul unei aplicații să execute acțiuni nedorite – *e.g.*, alterarea datelor

# *Cross-Site Request Forgery (CSRF)*

cazuri concrete:

modificarea adresei poștale + închirierea de filme  
de către persoanele având cont la Netflix (2006)

vulnerabilitate OAuth în Periscope Producer API (2017)  
[blog.innerht.ml/testing-new-features/](https://blog.innerht.ml/testing-new-features/)

acces cu GraphQL la un cont Facebook via Oculus (2018)  
[www.josipfranjkoVIC.com/blog/hacking-facebook-oculus-integration-csrf](https://www.josipfranjkoVIC.com/blog/hacking-facebook-oculus-integration-csrf)

control asupra contului la Khan Academy (17 mai 2019)  
[hackerone.com/reports/442901](https://hackerone.com/reports/442901)

# atacuri

## *Cross-Site Request Forgery (CSRF)*

poate conduce și la furtul identității (*phishing*)  
sau la plasarea de cod *malware* la client

[www.owasp.org/index.php/Cross-Site\\_Request\\_Forgery\\_%28CSRF%29](http://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29)

o soluție de contracarare:

biblioteca **CSRFGuard**

[github.com/aramrami/OWASP-CSRFGuard](https://github.com/aramrami/OWASP-CSRFGuard)

# atacuri

## *Cross Site History Manipulation (CSHM)*

breșă de securitate eludând *Same Origin Policy*,  
ce permite manipularea istoricului navigării  
de către un program malițios – *e.g.*, detectarea stării de  
autentificare a utilizatorului pe un sit, *user tracking*,  
acces la parametrii asociați unui URL,...

[tinyurl.com/qyurynm](http://tinyurl.com/qyurynm)

# atacuri

## Alte atacuri Web de tip *phishing*

folosirea de cod JavaScript pentru a modifica textul redat de navigatorul Web utilizatorului sau pentru a manipula utilizatorul să viziteze legături ascunse



[jeremiahgrossman.blogspot.com/2008/09/cancelled-clickjacking-owasp-appsec.html](http://jeremiahgrossman.blogspot.com/2008/09/cancelled-clickjacking-owasp-appsec.html)

*modus operandi* (2018): [blog.innerht.ml/google-yolo/](http://blog.innerht.ml/google-yolo/)  
caz real (10 mai 2019): [hackerone.com/reports/530008](http://hackerone.com/reports/530008)

# atacuri

## Alte atacuri Web de tip *phishing*

folosirea de cod JavaScript pentru a genera într-un *tab* al navigatorului o replică a unui formular de autentificare în cadrul unei aplicații – *e.g.*, Facebook, GMail



*tabnabbing*

[www.azarask.in/blog/post/a-new-type-of-phishing-attack/](http://www.azarask.in/blog/post/a-new-type-of-phishing-attack/)



# atacuri

## Alte atacuri Web de tip *phishing*

adoptarea de **tehnici de *social engineering***

*“any act that influences a person to take an action that may or may not be in their best interest”*

manipularea utilizatorilor – *e.g.*, furtul de parole –  
prin intimidare, șantaj, autoritate, flatare,  
substituție de persoană, vanitate etc.

[www.social-engineer.org](http://www.social-engineer.org)

# atacuri

## Exemple reale:

*Email spam campaign impersonating Google Docs (2017)*  
[reddit.com/r/google/comments/692cr4/new\\_google\\_docs\\_phishing\\_scam\\_almost\\_undetectable/](https://reddit.com/r/google/comments/692cr4/new_google_docs_phishing_scam_almost_undetectable/)

*Anatomy of an Amazon Phishing Attack (2017)*  
[shkspr.mobi/blog/2017/01/anatomy-of-an-amazon-phishing-attack/](https://shkspr.mobi/blog/2017/01/anatomy-of-an-amazon-phishing-attack/)

*Open Redirect Vulnerability @ Rockstar Games (2019)*  
[hackerone.com/reports/380760](https://hackerone.com/reports/380760)

*PayPal Android: Remote Theft of User Session (2019)*  
[hackerone.com/reports/424443](https://hackerone.com/reports/424443)

# atacuri

Soluții de contracarare:

inhibarea folosirii marcajelor HTML

HTML *escaping* via o bibliotecă specializată

filtrarea marcatorilor

separarea prezentării datelor de procesarea efectivă

etc.

# atacuri

## Probleme cauzate de URI/IRI-uri

inducerea în eroare a utilizatorului  
asupra domeniului Internet a sitului Web  
exemplu: <http://www.reddit.com@63.241.3.69/>

+

codificarea defectuoasă a codurilor hexa  
► vulnerabilități în cadrul unor servere Web

# atacuri

Probleme cauzate de **URI/IRI**-uri

includerea caracterelor Unicode  
probleme la decodificarea URL-urilor considerate „sigure”

siturile având domenii internaționale  
(**IDN** – *International Domain Names*)

► atacuri bazate pe homografie

detalii la [www.unicode.org/reports/tr36/](http://www.unicode.org/reports/tr36/)

exemplu: [www.xudongz.com/blog/2017/idn-phishing/](http://www.xudongz.com/blog/2017/idn-phishing/)

# atacuri

Probleme privind folosirea parolelor

majoritatea proceselor de autentificare utilizează parole

# atacuri

## Probleme privind folosirea parolelor

cu cât utilizatorul trebuie să rețină mai multe parole,  
cu atât sistemul de autentificare via parole e predispus  
la breșe de securitate:

alegerea unor parole slabe, folosite timp îndelungat

partajarea parolelor în grupuri de prieteni/colegi

scrierea parolelor pe hârtie – eventual, la vedere

recurgerea la aceeași parolă pentru aplicații Web multiple

# atacuri

Probleme privind folosirea parolelor

exemplu de atac:

pe baza unui dicționar sau *brute-force* asupra Twitter

► descoperirea parolei “*happiness*”

asociată unui cont cu drepturi de administrare

[blog.codinghorror.com/dictionary-attacks-101/](http://blog.codinghorror.com/dictionary-attacks-101/)

soluție tipică de prevenire:

conturi de administrare separate de conturile normale



# atacuri

## Troienii Web

situri/aplicații Web aparent folositoare,  
la care utilizatorul poate ajunge  
eventual via redirectare automată

suplimentar, pot recurge la XSS/CSRF  
sau la tehnici de tip *social engineering*

# atacuri

## Troienii Web

exemple: antiviruși falși, achiziții online de produse farmaceutice, software modificat de căutare pe Web

reclame abuzive (*large-scale abusive advertising*)

+

escrocare via plăți electronice (card de credit ori Bitcoin)

[cseweb.ucsd.edu/~savage/papers/CCS12Priceless.pdf](http://cseweb.ucsd.edu/~savage/papers/CCS12Priceless.pdf)

# atacuri

Exemple:

injectarea de biblioteci JS măsluite  
în cadrul CMS-urilor – *e.g.*, Joomla, Wordpress,...

*Fake jQuery Scripts in Nulled WordPress Plugins*

[blog.sucuri.net/2015/05/fake-jquery-scripts-in-nulled-wordpress-pugins.html](http://blog.sucuri.net/2015/05/fake-jquery-scripts-in-nulled-wordpress-pugins.html)

*jQuery.min.php Malware Affects Thousands of Websites*

[blog.sucuri.net/2015/11/jquery-min-php-malware-affects-thousands-of-websites.html](http://blog.sucuri.net/2015/11/jquery-min-php-malware-affects-thousands-of-websites.html)

injectarea de *plug-in-uri* false

de exemplu, bbPress la WordPress

[blog.sucuri.net/2017/01/fake-bb\\_press-plugin.html](http://blog.sucuri.net/2017/01/fake-bb_press-plugin.html)

Situație concretă:  
injectarea în *browser*  
(inclusiv pe platforme  
mobile) a unor  
programe JS – *script-uri*  
la distanță sau cod  
*ransomware* inclus în  
imagini – pretinzând  
existența unor  
actualizări pentru a fi  
inițiate atacuri

EXE

14 / 72

File name

browser.jpg

File size

2.12 MB

Last analysis










2019-02-25 13:55:47 UTC

Detection

Details

Relations

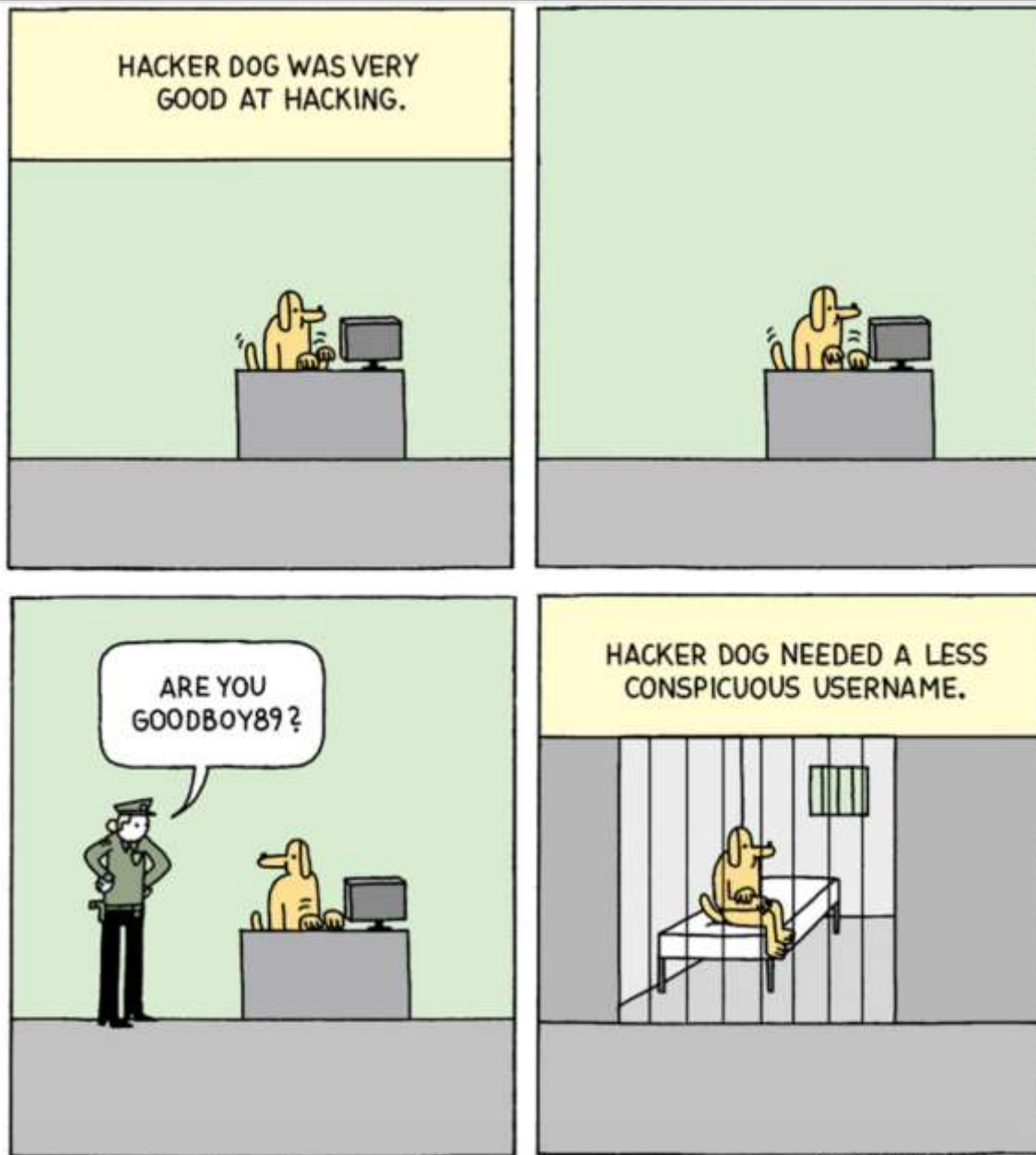
Community

Acronis	 suspicious
CrowdStrike Falcon	 malicious_confidence_100%
Cylance	 Unsafe
Endgame	 malicious (high confidence)
ESET-NOD32	 a variant of Win32/Kryptik.GP
Fortinet	 W32/Kryptik.GOUT!tr.ransom
Ikarus	 Trojan-Ransom.Crypted007
Qihoo-360	 HEUR/QVM20.1.7721.Malware
Rising	 Ransom.Cerber!8.3058 (TFE:2

D. Sinegubko, *Fake Browser Updates Push Ransomware  
and Bank Malware* (februarie 2019)

[blog.sucuri.net/2019/02/fake-browser-updates-push-ransomware-and-bank-malware.html](http://blog.sucuri.net/2019/02/fake-browser-updates-push-ransomware-and-bank-malware.html)

# (în loc de) pauză



# atacuri

## Refuz de servicii (*denial of service*)

exploatarea unor componente ale aplicației astfel încât funcționalitățile să nu poată fi oferite clienților reali

uzual, inițierea de procesări recursive  
(eventual, via programe care se autoreproduc)

M. Abliz, *Internet Denial of Service Attacks and Defense Mechanisms* (2011)  
[people.cs.pitt.edu/~mehmud/docs/abliz11-TR-11-178.pdf](http://people.cs.pitt.edu/~mehmud/docs/abliz11-TR-11-178.pdf)

# atacuri

## Refuz de servicii (*denial of service*)

exploatarea unor componente ale aplicației astfel încât funcționalitățile să nu poată fi oferite clienților reali

uzual, inițierea de procesări recursive

(eventual, via programe care se autoreproduc)

*fork bomb* – e.g., în Ruby: `loop { fork { __FILE__ } }`

*XML bomb*

*zip bomb* – [research.swtch.com/zip](http://research.swtch.com/zip)

## Exemplu real (*billions of lols*)

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
  <!ENTITY lol "lol">
  <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
  <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1; &lol1;&lol1;&lol1;&lol1;&lol1;">
  <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
  <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
  ...
  <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
  <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

B. Sullivan, *XML Denial of Service Attacks and Defenses* (2009)  
[msdn.microsoft.com/magazine/ee335713](http://msdn.microsoft.com/magazine/ee335713)

[www.owasp.org/index.php/XML\\_External\\_Entity\\_\(XXE\)\\_Processing](http://www.owasp.org/index.php/XML_External_Entity_(XXE)_Processing)



# atacuri

## *Ransomware*

încetarea unui tip de atac asupra unui sit Web – *e.g.*, DDoS  
sau criptarea conținutului – doar dacă proprietarul  
plătește o „taxă de protecție” (*i.e.* folosind Bitcoin)

exemplificări concrete:

[blog.sucuri.net/2015/12/ddos-extortions-campaigns.html](http://blog.sucuri.net/2015/12/ddos-extortions-campaigns.html)

[blog.sucuri.net/2016/01/ransomware-strikes-websites.html](http://blog.sucuri.net/2016/01/ransomware-strikes-websites.html)

# atacuri

## *Ransomware*

*The OWASP Anti-Ransomware Guide* (martie 2018):

[www.owasp.org/index.php/OWASP\\_Anti-Ransomware\\_Guide\\_Project](http://www.owasp.org/index.php/OWASP_Anti-Ransomware_Guide_Project)

detectie via capcane – *honeypots* (2016):

*Using honeypots to spot ransomware infections*

[www.owasp.org/images/0/03/OWASP\\_RansomwareHoneypots.pptx](http://www.owasp.org/images/0/03/OWASP_RansomwareHoneypots.pptx)

# atacuri

Tentative de acces la resurse presupus vulnerabile  
ori la secțiuni de administrare a unui sit Web

208.113.197.80 GET [/wp-admin/](#)  
5.196.16.176 GET [/~jromai/romaijournal//images/stories/post.gif](#)  
185.22.64.241 GET [/~busaco/docs/jdownloads/screenshots/has.php.j?rf](#)  
5.196.16.176 POST [/index.php?option=com\\_jce&task=plugin&file=imgmanager&method=form&cid=20&6bc427c8a7981f4fe1f5ac65c=cf6dd3cf1923c950586](#)  
38.87.45.121 GET [/~vcosmin/WikiLogica/index.php?title=BuckYoung847](#)  
74.220.207.111 GET [/wp-admin/admin-ajax.php?action=revslider\\_ajax\\_action](#)  
74.220.207.111 GET [/index.php?gf\\_page=upload](#)  
195.30.97.113 POST [//index.php?option=com\\_jdownloads&Itemid=0&view=upload](#)  
5.153.237.232 POST [/~flash/wiki/index.php?title=Special:Userlogin&action=submitlogin](#)  
46.102.103.137 POST [/~flash/wiki/index.php?title=Special:Userlogin&action=submitlogin](#)

# atacuri

Detectarea posibilelor vulnerabilități  
– datorate unor configurații incorecte/implicite  
ale serverelor și/sau aplicațiilor Web –  
se poate realiza apelând la un motor de căutare

proiectul *Google Hack HoneyPot* – [ghh.sourceforge.net](http://ghh.sourceforge.net)

alte resurse de interes la [www.honeynet.org](http://www.honeynet.org)

## Exemple de acțiuni:

detectia versiunilor de programe cu *bug*-uri cunoscute:

**"Apache/2.0.52 server at"**

accesul la fisiere *.bak*: **inurl:index.php.bak**

detectarea paginilor de administrare: **"admin login"**

instalări implicite: **intitle:"welcome to" intitle:internet IIS**

localizarea interfețelor spre sisteme de baze de date:

**inurl:main.php phpMyAdmin**

căutarea de aplicații ori a fișierelor de jurnalizare:

**inurl:error.log +filetype:log -cvs**

mesaje de eroare generate de aplicații ori servere de baze de date: **"ASP.NET\_SessionId" "data source="**

PHP	176,761
JavaScript	157,954
Python	14,922
HTML	13,865
C	12,343
VimL	2,514
HTML+ERB	1,934
Ruby	740
Text	683
JSON	415

alternativă: căutarea de programe potențial vulnerabile  
în depozite de cod-sursă disponibile public

cazul GitHub: detecția execuției de cod – *e.g.*, **exec(\$\_GET**

# prevenirea

## Studiu de caz: securizarea serverului Apache

eliminarea modulelor care nu sunt esențiale

`mod_autoindex`, `mod_dav`, `mod_info`, `mod_include`, `mod_status`,...

restrângerea permisiunilor implicite pentru  
directoarele `/`, `/var/www/html` (directorul *root* al sitului),  
directoarele `(public_)``html/` ale utilizatorilor

rularea serverului ca utilizator cu drepturi minime,  
cu limitarea accesului la resursele sistemului

# prevenirea

Studiu de caz: securizarea serverului Apache

„imunizarea” fișierelor de configurare importante

rularea Apache într-un *chroot jail*

a se vedea [github.com/ZenProjects/Apache-mod-chroot](https://github.com/ZenProjects/Apache-mod-chroot)

eliminarea creării „semnăturii” serverului

pentru paginile generate automat:

[ServerSignature Off](#) si [ServerTokens Prod](#)

recurgerea la [mod\\_ssl](#) pentru oferirea de conexiuni HTTPS



# prevenirea

Studiu de caz: securizarea serverului Apache

verificarea/ajustarea permisiunilor fișierelor publice

limitarea/inhibarea *upload*-urilor de fișiere

limitarea folosirii `.htaccess` de utilizatorii obișnuiți

interzicerea accesului la tabela `users` la MySQL

configurarea serverelor de aplicații să nu trimită  
*browser*-ului mesaje de eroare – la PHP: `display_errors off`

# prevenirea

Studiu de caz: securizarea serverului Apache

rularea *script*-urilor în mod „sigur”

Perl în *taint mode*, PHP: `safe_mode on, allow_url_fopen off`

semnarea codului ca fiind „sigur” – pentru Java/.NET

actualizarea sitului doar prin metode securizate:

`ssh, scp, sftp`

pentru reguli de bună practică, a se consulta

[httpd.apache.org/docs/2.4/misc/security\\_tips.html](http://httpd.apache.org/docs/2.4/misc/security_tips.html)

# prevenirea

La nivel de servere de aplicații/platforme Web

exemplificări diverse:

ASP.NET Core – [docs.microsoft.com/aspnet/core/security/](https://docs.microsoft.com/aspnet/core/security/)

Node.js – [hackerone.com/nodejs-ecosystem](https://hackerone.com/nodejs-ecosystem)

PHP – [phpsecurity.readthedocs.org](https://phpsecurity.readthedocs.org)

Python – [www.pythonsecurity.org](https://www.pythonsecurity.org)

altele la [github.com/OWASP/CheatSheetSeries](https://github.com/OWASP/CheatSheetSeries)

# prevenirea

Securitatea serviciilor (API-urilor) Web

folosirea obligatorie a HTTPS  
+  
certIFICATE digitale actualizate

# prevenirea

Securitatea serviciilor (API-urilor) Web

datele „sensibile” nu trebuie expuse în URL

corect:

[https://web.info/resourceCollection/\[ID\]/action](https://web.info/resourceCollection/[ID]/action)

incorect:

<http://web.info/controller/7/action?apiKey=s74b1901c07>

# prevenirea

## Securitatea serviciilor (API-urilor) Web

permiterea accesului la API doar prin cheie (*API key*)

+

verificarea acestei chei pentru fiecare cerere în parte

utilizatorii trebuie să recurgă la un mecanism solid de autentificare, precum cea multi-factor

# prevenirea

## Securitatea serviciilor (API-urilor) Web

privilegii reduse: restricționarea metodelor HTTP  
de pildă, doar GET

+

folosirea de liste de clienți/utilizatori agreați (*whitelist*)

# prevenirea

Securitatea serviciilor (API-urilor) Web

utilizarea OAuth pentru autorizare

a se considera și schimbul de date  
via JWT (*JSON Web Tokens*) pentru controlul accesului



# prevenirea

## Securitatea serviciilor (API-urilor) Web

validarea datelor de intrare  
(parametri, câmpuri de antet HTTP,...)

aspecte: lungime, tip de date, format, valori permise  
cereri având date neașteptate/dubioase trebuie rejectate

+

verificarea – și, eventual, validarea – tipului conținutului

# prevenirea

## Securitatea serviciilor (API-urilor) Web

tratarea erorilor:

recurgerea la mesaje generice și

evitarea transmiterii detaliilor tehnice clientului

+

jurnalizarea și analizarea cererilor (*audit logs*)

# prevenirea

Securitatea serviciilor (API-urilor) Web

folosirea adecvată a codurilor de stare HTTP

[www.restapitutorial.com/httpstatuscodes.html](http://www.restapitutorial.com/httpstatuscodes.html)

contra-exemplu:

orice succes via 200 *OK*

orice situație de eroare raportată cu 404 *Not Found*

# prevenirea

## Securitatea serviciilor (API-urilor) Web

recurgerea la anteturi HTTP  
vizând securizarea transferului datelor

[www.owasp.org/index.php/OWASP\\_Secure-Headers\\_Project](http://www.owasp.org/index.php/OWASP_Secure-Headers_Project)  
ghid util (2018): [www.keycdn.com/blog/http-security-headers](http://www.keycdn.com/blog/http-security-headers)

Modalități de supraviețuire în caz de atac?

# supraviețuirea

Sistemul trebuie să-și ducă până la capăt misiunea chiar dacă unele componente/părți din sistem sunt afectate ori scoase din uz

îndeplinirea funcționalităților vitale (*mission-critical*)

► identificarea serviciilor esențiale

exemplu:

oferirea unei copii *read-only* a conținutului

# supraviețuirea

Proprietăți importante ale sistemului:

rezistența la atacuri

recunoașterea atacurilor și efectelor lor

adaptarea la atacuri

# supraviețuirea

## Rezistența la atacuri

strategii de respingere a atacului:

validarea obligatorie a datelor

autentificarea utilizatorilor

acordarea privilegiilor minime

acces la servicii Web ori API-uri doar pe baza unei chei

...



# supraviețuirea

Recunoașterea atacurilor și efectelor lor

strategii pentru restaurarea datelor,  
limitarea efectelor, menținerea/restaurarea  
serviciilor compromise

ferme de servere Web (*Web farms*) – eventual, în *cloud*

RAID (*Redundant Array of Independent Disks*)

SAN (*Storage Area Network*)

copii de siguranță (*backup-uri*): complete/incrementale

# supraviețuirea

## Adaptarea la atacuri

strategii pentru îmbunătățirea nivelului (șansei)  
de supraviețuire

analiză (audit)

învățarea din greșeli

recurgerea la expertiza unor companii specializate

...

# răspunsul la incidente

**Răspunsurile agresive – *e.g., hack back* –  
sunt prohibite**

# răspunsul la incidente

**Răspunsurile agresive – *e.g., hack back* –  
sunt prohibite**

uzual, se recurge la metodologia SANS  
(*System Administration, Networking, and Security*)

etape:

pregătire ▶ identificare ▶ controlul efectelor (*containment*)  
▶ eradicare ▶ recuperare ▶ continuare (*follow-up*)

[www.sans.org/security-resources/](http://www.sans.org/security-resources/)

# răspunsul la incidente

## *Forensics*

proces de „prindere” a *cracker*-ilor

*investigation of digital evidence  
for use in criminal or civil courts of law*

[forensicswiki.org](http://forensicswiki.org)

# răspunsul la incidente

## *Forensics*

uzual, are loc după un incident de securitate

implică analizarea *hardware*-ului (discuri, RAM),  
„deșeurilor” (*information detritus*), *log*-urilor,  
fișierelor de configurare și altele

diverse instrumente software – bazate pe Linux:

[forensics.cert.org](http://forensics.cert.org)

[resources.infosecinstitute.com/computer-forensics-tools/](http://resources.infosecinstitute.com/computer-forensics-tools/)

# răspunsul la incidente

## *Forensics*

acțiunea de „ștergere” a urmelor = *anti-forensics*

o serie de detalii la  
[forensicswiki.org/wiki/Anti-forensic\\_techniques](https://forensicswiki.org/wiki/Anti-forensic_techniques)

# monitorizare & testare

Teste de verificare a...

capacității de deservire a clienților

robusteței

rulării în situații extreme



# monitorizare & testare

Se iau în considerație:

caracteristicile *browser*-ului Web (+setările implicite)

platforma/platforme: hardware, sistem de operare,...

interfața: rezoluția ecranului, adâncimea de culoare,...

politica de *caching* (+siguranța *proxy*-ului)

suportul pentru redarea unor tipuri de documente  
(securitatea folosirii *plugin*-urilor)

limbajul/limbajele de programare utilizate  
(inclusiv serverul/serverele de aplicații, bibliotecile etc.)

# monitorizare & testare

Teste specifice legate de programare:

depășiri de *buffer*-e (*buffer overflow*)

exemplu: lungimea URI-urilor trimise de client

caz real:

Apple iTunes for Windows (versiunea < 8.2) permitea execuția de cod arbitrar la utilizarea schemei URL itms:

[www.securitytracker.com/id/1022313](http://www.securitytracker.com/id/1022313)

# monitorizare & testare

Teste specifice legate de programare:

probleme de prelucrare (*parsing*)

procesarea URI-urilor, a datelor primite via formulare, *cookie*-uri, entităților (X)HTML, datelor XML, cererilor HTTP, XML-RPC și SOAP, interogărilor SQL, datelor JSON etc.

N. Seriot, *Parsing JSON is a Minefield* (2016)  
[seriot.ch/parsing\\_json.php](http://seriot.ch/parsing_json.php)

# monitorizare & testare

Teste specifice legate de programare:

probleme de conversie a datelor

de exemplu, ASCII  $\leftrightarrow$  Unicode

reguli de bună practică:

RFC 5137 – [tools.ietf.org/html/rfc5137](https://tools.ietf.org/html/rfc5137)

# monitorizare & testare

Teste specifice legate de programare:

probleme de redare a datelor

exemplificare:

afişarea perechii *nume prenume* atunci când

**nume="<script>document.location="**

**prenume="un\_uri"</script>"**

# monitorizare & testare

Teste specifice legate de programare:

probleme de *escaping*

exemplu:

*character escaping* pentru șirul `cs/b`

`cs%2Fb`

`cs%%252Fb`

`cs%25%32%46b`

# monitorizare & testare

Teste specifice legate de programare:

probleme de *escaping*

„injectare” directă a datelor via URI sau prin intermediul  
interfeței Web sau via un fișier (*upload* ilegal)  
ori folosind un program  
(*e.g.*, de administrare la distanță a aplicației),...

► verificarea *escaping*-ului via instrumente dedicate

# monitorizare & testare

Soluții și strategii:

programare defensivă  
(*defensive programming*)

adoptarea standardelor de redactare a codului  
(*enforcing coding standards*)

recurgerea la unități de testare  
(*unit testing*)



# monitorizare & testare

## Soluții și strategii:

includerea unui sistem de prevenire, detectare  
și raportare a erorilor survenite în cod  
+ un sistem de urmărire a *bug*-urilor (*bug tracking*)

folosirea unui sistem de control al versiunilor

a se revedea cursul  
privitor la inginerie Web

# monitorizare & testare

## Teste specifice legate de intimitate (*privacy*):

datele obținute de la utilizator trebuie tratate ca fiind sigure și confidențiale

Ce date vor fi disponibile în *cache*-ul clientului?

*Cookie*-urile/datele din LocalStorage pot conține date sensibile, posibil de exploatat de persoane rău-voitoare?

Cum se invalidează *cache*-ul?

# monitorizare & testare

Teste privitoare la integrarea componentelor:

**gradul de securitate al unei aplicații  
este dat de gradul de securitate  
al celei mai vulnerabile componente**

# monitorizare & testare

Teste privitoare la integrarea componentelor:

**gradul de securitate al unei aplicații  
este dat de gradul de securitate  
al celei mai vulnerabile componente**

neverificarea validității identicatorului de utilizator  
la nivel de server, pe baza faptului că această verificare  
s-a efectuat deja la nivelul *browser*-ului  
caz real: [www.ifc0nfig.com/dominos-pizza-and-payments/](http://www.ifc0nfig.com/dominos-pizza-and-payments/)

# monitorizare & testare

Teste privind opacizarea datelor (*obfuscation*):

datele nu trebuie stocate în locații predictibile

conținutul propriu-zis poate conduce  
la probleme de securitate – *information disclosure*

*e.g.*, acces la Webcam – context: IoT (*Internet Of Things*)  
[www.ifc0nfig.com/a-close-look-at-the-philips-in-sight-ip-camera-range/](http://www.ifc0nfig.com/a-close-look-at-the-philips-in-sight-ip-camera-range/)

# monitorizare & testare

Breşe referitoare la *information disclosure*:

accesarea câmpurilor ascunse ale formularelor Web  
şi/sau  
a comentariilor din codul-sursă HTML, CSS, JavaScript

# monitorizare & testare

Breșe referitoare la *information disclosure*:

consultarea fișierului **robots.txt**

- scanarea fișierelor de configurare sau a directoarelor temporare – *e.g.*, rapoarte ale traficului

User-agent: \*

Disallow: /plenum/data/5510903.doc

Disallow: organization/193959.pdf

Disallow: /en/community/thread/12819

...

detalii la [thiébaud.fr/robots.txt.html](http://thiébaud.fr/robots.txt.html)

# monitorizare & testare

Breșe referitoare la *information disclosure*:

mesaje de eroare emise de aplicații și/sau API-uri

fișiere având extensii incorecte

► acces la codul-sursă al *script*-urilor de pe server

vizualizarea conținutului directoarelor serverului

scanarea traficului de rețea

(URI-uri, date XML/JSON transmise asincron,...)



```
500 TypeError: /usr/local/sparqls/node/views/content/performance.jade:45 43|  
span(onmouseover='tooltip.show(\#{configPerformance["Cold-Warm"]})\')', onmouseout='tooltip.hide();')  
(Cold-Warm) 44| tbody > 45| - each ep, i in ptasks_agg 46| tr(class=(i % 2 == 0) ? 'odd' : 'even') 47|  
//-Display Endpoint Label 48| //-TODO: if more than one endpoint then display how many and their names  
Cannot read property 'length' of undefined
```

```
43| span(onmouseover='tooltip.show(\#{configPerformance["Cold-Warm"]})\')', onmouseout='tooltip.hide();') (Cold-Warm)  
44| tbody  
> 45| - each ep, i in ptasks_agg  
46| tr(class=(i % 2 == 0) ? 'odd' : 'even')  
47| //-Display Endpoint Label  
48| //-TODO: if more than one endpoint then display how many and their names
```

Cannot read property 'length' of undefined

```
at jade_debug.unshift.lineno (eval at (/usr/local/sparqls/node/node_modules/jade/lib/jade.js:179:8), :708:31)  
at eval (eval at (/usr/local/sparqls/node/node_modules/jade/lib/jade.js:179:8), :1061:4)  
at eval (eval at (/usr/local/sparqls/node/node_modules/jade/lib/jade.js:179:8), :1378:22)  
at res (/usr/local/sparqls/node/node_modules/jade/lib/jade.js:180:38)  
at Object.exports.render (/usr/local/sparqls/node/node_modules/jade/lib/jade.js:305:10)  
at Object.exports.renderFile (/usr/local/sparqls/node/node_modules/jade/lib/jade.js:341:18)  
at View.exports.renderFile [as engine] (/usr/local/sparqls/node/node_modules/jade/lib/jade.js:326:21)  
at View.render (/usr/local/sparqls/node/node_modules/express/lib/view.js:76:8)  
at Function.app.render (/usr/local/sparqls/node/node_modules/express/lib/application.js:505:10)  
at ServerResponse.res.render (/usr/local/sparqls/node/node_modules/express/lib/response.js:756:7)
```

acces nedorit la datele privind erorile survenite +  
codul-sursă al unei aplicații Web  
(aici, Node.js recurgând la *framework*-ul Express)

# monitorizare & testare

## Teste specifice legate de exploatare:

pregătirea judicioasă a exploatării în practică  
(*deployment*)

detectarea problemelor de flux

tratarea corespunzătoare a codurilor HTTP 4xx și 5xx,  
acces la resurse autentificate (*e.g.*, obținerea unor date  
fără autentificarea prealabilă a utilizatorului),  
execuția anormală a *script*-urilor etc.

# monitorizare & testare

## Teste specifice legate de exploatare:

testarea interacțiunii cu aplicația Web

► programe simulând vizitatori virtuali  
de experimentat **Selenium** – [www.seleniumhq.org](http://www.seleniumhq.org)

realizarea testelor de încărcare (*load testing*)

► scenarii și interpretarea rezultatelor

# monitorizare & testare

Instrumentele de stresare (*stressing tools*)  
pot oferi informații privitoare la...

performanță

*e.g.*, timp de răspuns, timp de generare a conținutului,...

detalii la „Dezvoltarea aplicațiilor Web cu JavaScript”  
[profs.info.uaic.ro/~busaco/teach/courses/staw/](http://profs.info.uaic.ro/~busaco/teach/courses/staw/)

# monitorizare & testare

Instrumentele de stresare (*stressing tools*)  
pot oferi informații privitoare la...

scalabilitate

memorie ocupată, utilizarea discului, numărul de  
conexiuni privind alte servicii, comportament etc.

# monitorizare & testare

Instrumentele de stresare (*stressing tools*)  
pot oferi informații privitoare la...

corectitudine

rapoarte privind funcționarea  
(eronată a) unor componente

*e.g.*, pe baza fișierelor de jurnalizare (*log-uri*)

# monitorizare & testare

Instrumentele de stresare (*stressing tools*)  
pot oferi informații privitoare la...

lacune de securitate

# instrumente (exemple)

**AppScan, skipfish, w3af, WebInspect**  
scanare de vulnerabilități

**Burp, Paros, WebScarab**  
suite de testare Web

instrumentele native pentru dezvoltatori  
oferite de navigatoarele Web + extensii specifice

a se consulta și [sectools.org/tag/web-scanners/](https://sectools.org/tag/web-scanners/)



# de reținut

Securitatea unei aplicații Web:  
trebuie să ia în considerație **arhitectura,**  
**funcționalitatea, codul-sursă**  
**și conținutul în ansamblu**

# de reținut

Securitatea unei aplicații Web:

nu vizează vulnerabilitățile sistemului de operare  
ori ale programelor auxiliare

# de reținut

Vulnerabilitățile unei aplicații Web  
nu sunt neapărat „celebre”  
și pot fi independente deseori de securitatea  
sistemului pe care este exploatat situl

liste ale vulnerabilităților Internet, inclusiv Web:

[cve.mitre.org/cve/](https://cve.mitre.org/cve/)


[www.exploit-db.com/webapps/](https://www.exploit-db.com/webapps/)

[www.hackerone.com/internet-bug-bounty](https://www.hackerone.com/internet-bug-bounty)

# OWASP Top 10 Most Critical Web Application Security Risks (2013 *versus* 2017)

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	⊗	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	⊗	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

[www.owasp.org/index.php/Category:OWASP\\_Top\\_Ten\\_Project](http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project)

RISK							Score
	Threat Agents	Exploitability	Prevalence	Detectability	Technical	Business	
A1:2017-Injection	App Specific	EASY: 3	COMMON: 2	EASY: 3	SEVERE: 3	App Specific	8.0
A2:2017-Authentication	App Specific	EASY: 3	COMMON: 2	AVERAGE: 2	SEVERE: 3	App Specific	7.0
A3:2017-Sens. Data Exposure	App Specific	AVERAGE: 2	WIDESPREAD: 3	AVERAGE: 2	SEVERE: 3	App Specific	7.0
A4:2017-XML External Entities (XXE)	App Specific	AVERAGE: 2	COMMON: 2	EASY: 3	SEVERE: 3	App Specific	7.0
A5:2017-Broken Access Control	App Specific	AVERAGE: 2	COMMON: 2	AVERAGE: 2	SEVERE: 3	App Specific	6.0
A6:2017-Security Misconfiguration	App Specific	EASY: 3	WIDESPREAD: 3	EASY: 3	MODERATE: 2	App Specific	6.0
A7:2017-Cross-Site Scripting (XSS)	App Specific	EASY: 3	WIDESPREAD: 3	EASY: 3	MODERATE: 2	App Specific	6.0
A8:2017-Insecure Deserialization	App Specific	DIFFICULT: 1	COMMON: 2	AVERAGE: 2	SEVERE: 3	App Specific	5.0
A9:2017-Vulnerable Components	App Specific	AVERAGE: 2	WIDESPREAD: 3	AVERAGE: 2	MODERATE: 2	App Specific	4.7
A10:2017-Insufficient Logging&Monitoring	App Specific	AVERAGE: 2	WIDESPREAD: 3	DIFFICULT: 1	MODERATE: 2	App Specific	4.0

factori de risc asociați celor mai importante vulnerabilități

# de reținut

## Principii de securitate a aplicațiilor Web

### separarea serviciilor

sisteme diferite pentru server Web,  
server de aplicații, server de stocare (baze de date) etc.

# de reținut

## Principii de securitate a aplicațiilor Web

### limitarea privilegiilor

la nivel de sistem de fișiere,  
pentru baze de date,  
acordarea de permisiuni utilizatorilor  
sub care rulează aplicațiile – *e.g.*, Apache, Tomcat,...

# de reținut

Principii de securitate a aplicațiilor Web

ascundere a secretelor – *e.g.*, parole, SID-uri,...

recurgere la biblioteci standard ...

actualizate!

menținere + studiere a fișierelor de jurnalizare (*log-uri*)

efectuare de teste și ajustări (*Web tuning*)



## Reguli/bune practici (Sverre Huseby):

*Do not underestimate the power of the dark side*

*Use POST/PUT requests when actions have side effects*

***In a server-side context,  
there is no such thing as client-side security***

*Always generate a new session ID once the user logs in*

*Never pass detailed error messages to the client*

*Identify every possible meta-character to a subsystem*

*When possible, pass data separate from control information*

## Reguli/bune practici (Sverre Huseby):

*Do not blindly trust the API documentation*

*Identify all sources of input to the application*

*When filtering data,  
use white-listing rather than black-listing*

*Create application-level logs*

***Never use client-side scripts for security***

*Pass as little internal state information  
as possible to the client*

## Reguli/bune practici (Sverre Huseby):

*Don't assume that requests will come in a certain order*

*Filter all data before including them in a Web page,  
no matter what the origin*

*Stick to existing cryptographic algorithms,  
do not create your own*

***Never store clear-text passwords***

*Assume that server-side code is available to attackers*

***Security is not a product – it is a process***

# de reținut

Riscurile de securitate nu vizează doar  
proprietarul sitului/aplicației Web,  
ci și utilizatorul final

## de reținut

Riscurile de securitate nu vizează doar proprietarul sitului/aplicației Web, ci și utilizatorul final

acțiuni tipice:

spionare a utilizatorului (*user tracking*)

incluere de mesaje promoționale (*ad injection malware*)

evenimente + resurse:

[www.ieee-security.org](http://www.ieee-security.org) • [www.w3.org/Security/  
security.googleblog.com](http://www.w3.org/Security/security.googleblog.com)

# de reținut

Disconforturi cauzate de un sit/aplicație nesigur(ă)

**financiare** – pierdere de bani/informații

**de performanță** – *e.g.*, blocarea/încetinirea acțiunilor

**psihologice** – insatisfacție ► influență asupra UX

**sociale** – *e.g.*, incapacitatea de muncă, lipsa comunicării,...

**de timp** – navigare greoaie, deturnare spre alt sit etc.

Finalmente, testul #3...

[A—L] Imaginați posibile soluții de contracarare a atacurilor (D)DoS asupra *mash-up*-urilor Web.

[M—Z] Specificați trăsături + comportamente malefice ale unui API REST de tip troian Web.



[A—K] Cum pot fi prevenite atacuri de tip *injection* asupra serviciilor Web?

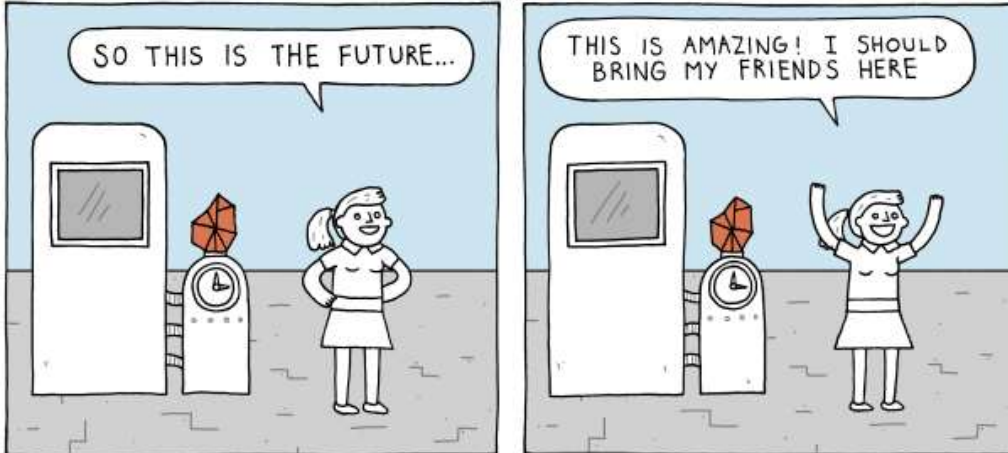
[L—Z] Discutați posibile atacuri Web vizând transmiterea asincronă a datelor via Ajax.

# rezumat

## securitatea aplicațiilor Web



context, tipuri de atacuri, vulnerabilități, prevenire,  
reguli de bună practică, studii de caz



# Mult succes!