



Security @ HyperScale

Claudio Criscione



Manager @ Google - Large scale security testing



Disclaimer: my own opinions, not my employer's.

Claudio Criscione - Security @ Hyperscale - HackInBo Spring 2018

Mi hanno detto
che c'è roba da
hackare...



HACK IN BO®
Spring 2018 Edition
10ª EDIZIONE



“Tutto il codice è in quel repo lì dietro. [...] Guarda non penso nemmeno si accorgano fino a che stai sotto le 1.000 CPU.”





Il mio primo progetto

PappaReale: test semantico automatico su larga scala per JSON APIs.

Deriva automaticamente la struttura delle APIs per fuzzing “semantico”: identifica candidati per analisi manuale con FP < 2%.

Obiettivo: trovare decine di bug su decine di APIs!





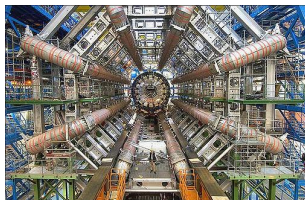
Aspetta, che è
successo?



La natura del problema



Linux Kernel 4.15.8
20.323.000 Righe di codice



Large Hadron Collider
50.000.000 Righe di codice



La natura del problema



Google codebase

2.000.000.000+ Righe
86 TB, 9.000.000 files
dati del 2015

La natura del problema

16.000 modifiche al giorno da devs,
24.000 automatizzate

25.000+ sviluppatori

7+ linguaggi di programmazione





PostMortem: che ho sbagliato?

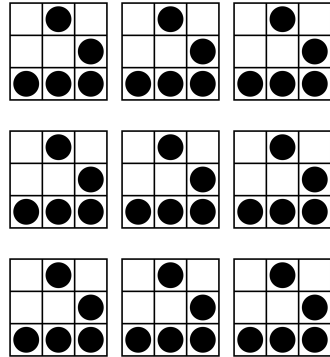
Release the dogs
of war!



HACK IN BO®
Spring 2018 Edition
10ª EDIZIONE

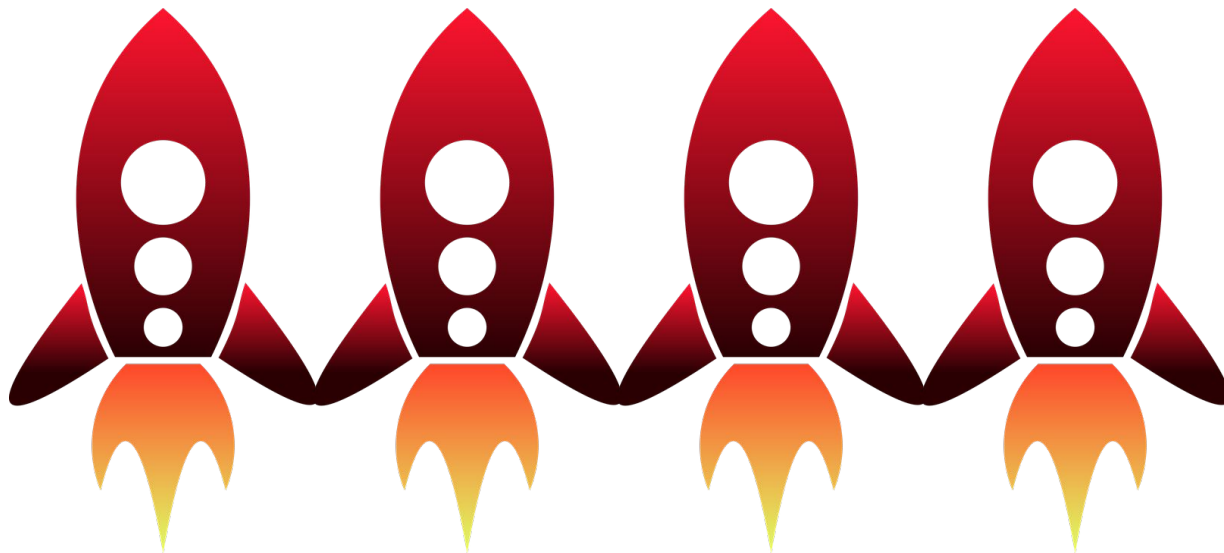
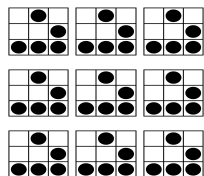


Abbiamo un sacco di gente, vero?



ISE (Information Security Engineering)

Ma abbiamo un sacco più progetti...



Una risorsa da difendere



Tooling e automazione:
questionari e segnali

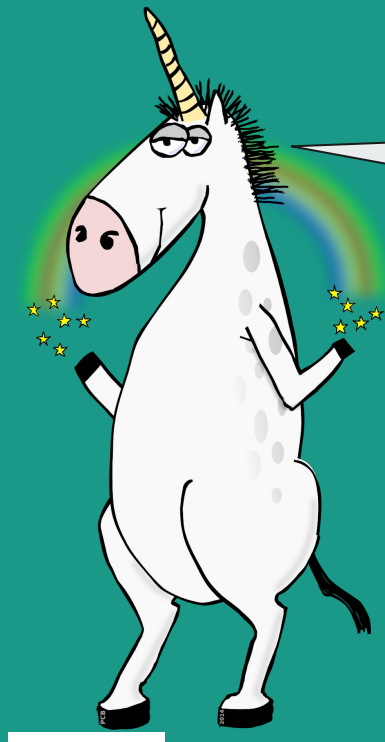
Triage e minimizzazione

Focus sugli errori di design.

#1

**Il personale di security è raro
e (si spera!) costoso, e non è
la chiave per scalare.**





Idea: abbiamo un
sacco di sviluppatori!

HACKINBO®
Spring 2018 Edition
10ª EDIZIONE



Il problema con l'*educascion*

Quanti in questa stanza
pensano di passare i
colloqui come software
engineer a Google?

Mettici un captcha

Non usare PHP
Non usare Electron

Implementa rate limiting

HACK IN BO®
Spring 2018 Edition
10ª EDIZIONE

Il problema con l'educascion

E allora perchè ci
aspettiamo che un
software engineer sia
anche security engineer?

Niente output senza escaping

Non fare input senza validazione

Unbounded Array Allocation
Bounded Memory

Attento agli XSS!!

Controlla il token XSRF

Guarda questi 7231 warnings

Ama il DOM tuo come te stesso

Un approccio diverso

Modificare framework, API, linguaggi e processi per rendere quasi impossibile introdurre vulnerabilità*.

*Terms and conditions apply

Claudio Criscione - Security @ Hyperscale - HackInBo Spring 2018



Sviluppare codice sicuro, anno 2018



ATTENZIONE: IMMERSIONE TECNICA


Hack in Bo®
Spring 2018 Edition
10ª EDIZIONE



Bug di design vs implementazione

Bug di Design

Diffusi nel sistema

Complessi e costosi

Subdoli ed infrequenti



Bug di Implementazione

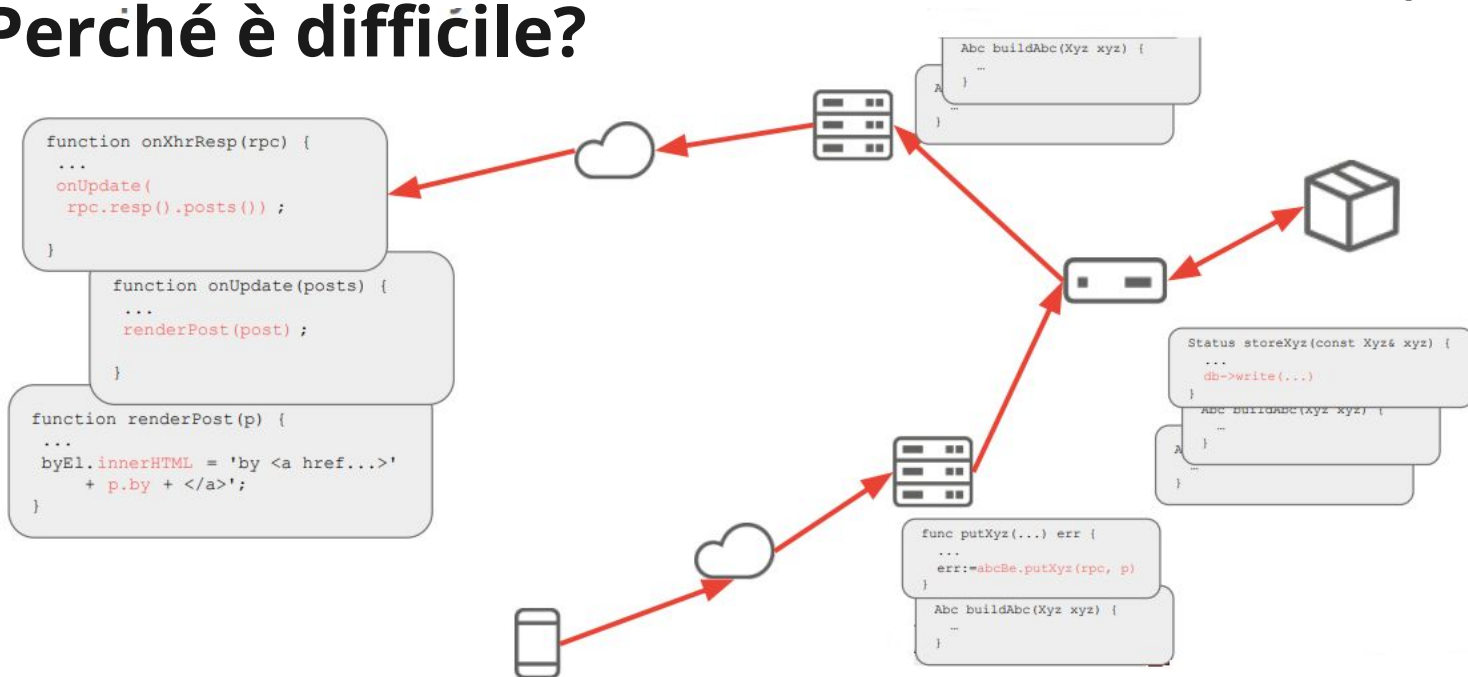
Locali e ***patchabili***

Semplici e ***testabili***

Ricorrenti ed ubiqui



Perché è difficile?



L'idea fondamentale

Trasferire la “sicurezza” di un dato (i.e. precondizioni) con il suo Tipo (SafeType)

Tutte le (nuove) API che lavorano sui SafeTypes mantengono le precondizioni.

API su tipi unsafe li controllano/traducono.

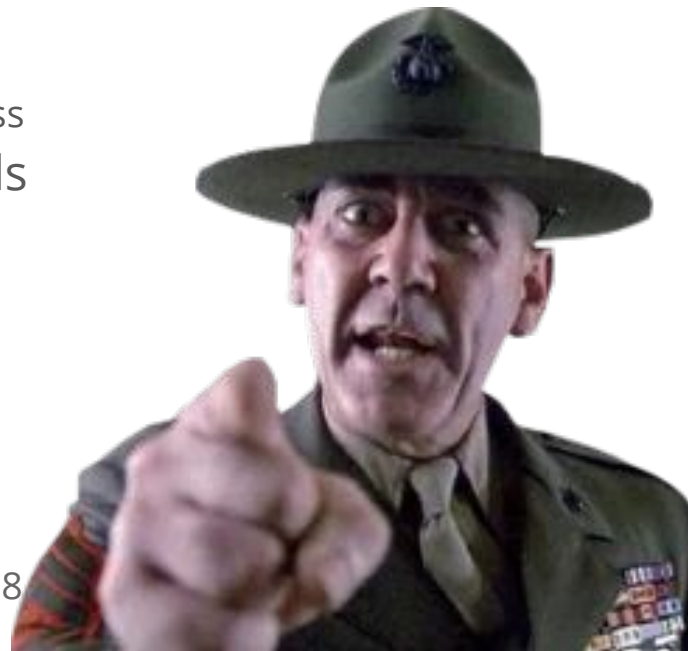



Per esempio

```
* @param {!HTMLAnchorElement} anchor The anchor element whose href property
*   is to be assigned to.
* @param {string|!goog.html.SafeUrl} url The URL to assign.
* @see goog.html.SafeUrl#sanitize
*/
goog.dom.safe.setAnchorHref = function(anchor, url) {
  goog.dom.asserts.assertIsHTMLAnchorElement(anchor);
  /** @type {!goog.html.SafeUrl} */ var safeUrl;
  if (url instanceof goog.html.SafeUrl) {   safeUrl = url; }
  else { safeUrl = goog.html.SafeUrl.sanitizeAssertUnchanged(url); }
  anchor.href = goog.html.SafeUrl.unwrap(safeUrl);
};
```

Implementazione

- Linter - Compile checks
 - Verificano che non siano utilizzabili API di bypass
- Verifica manuale delle (poche) unsafe calls
- Supporto nativo nei framework con equivalenza semantica

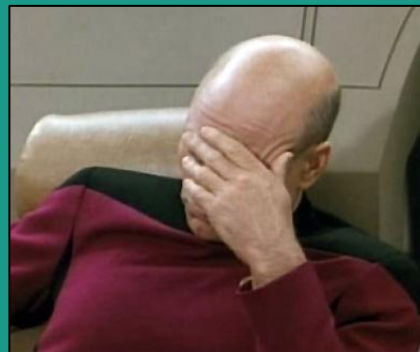


A dramatic scene of Roman soldiers in a narrow corridor, with a large black circle containing text overlaid. The soldiers are wearing helmets and carrying shields and spears, creating a sense of a tight, crowded space. The lighting is dramatic, with strong highlights and deep shadows.

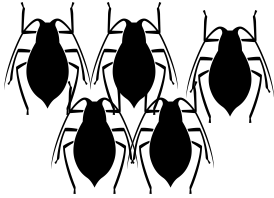
**Si, ma quante
risorse ci
vogliono?**

5 Ingegneri, in un
angusto corridoio

NonUsareOTiVeniamoACercare (..)



Funziona?



Per il codice nuovo, **nessun** costo aggiuntivo.

Prevenire è meglio che curare



Se avete capito perchè c'è una mela in questa slide, siete vecchi come me. Mi spiace (era l'88).

#2

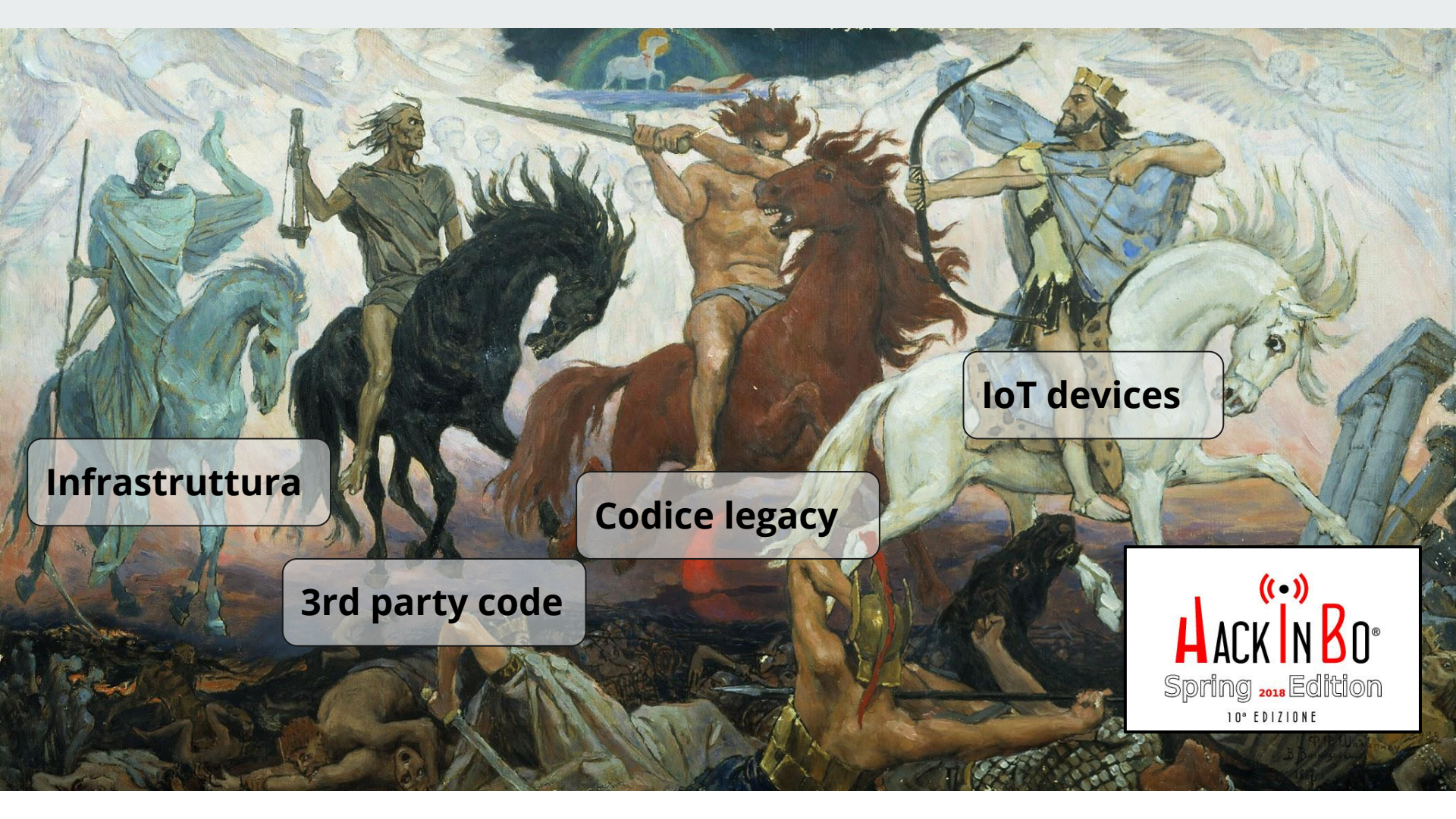
Training, testing, auditing
etc. sono fantastici, ma
eliminare i bug alla radice
scala (molto) meglio.
Poka-yoke!



Eh, ma su questa roba
non si può fare!



HACK IN BO®
Spring 2018 Edition
10ª EDIZIONE



Infrastruttura

IoT devices

Codice legacy

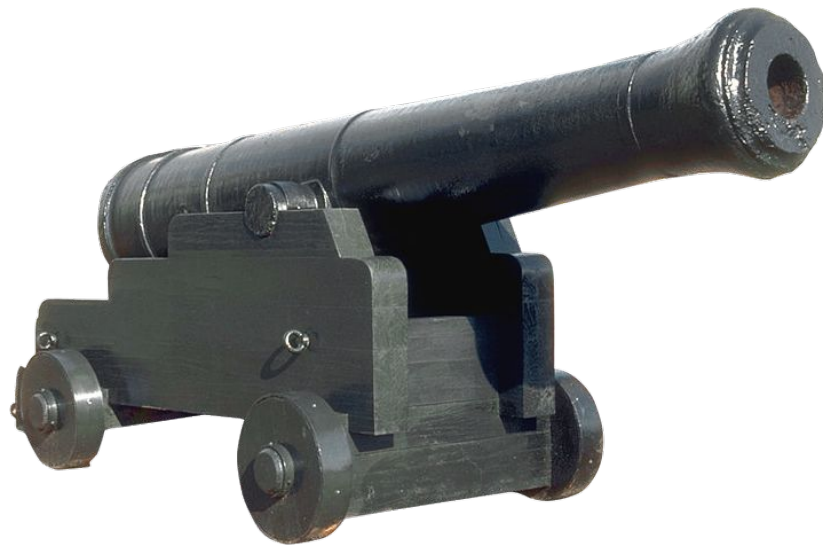
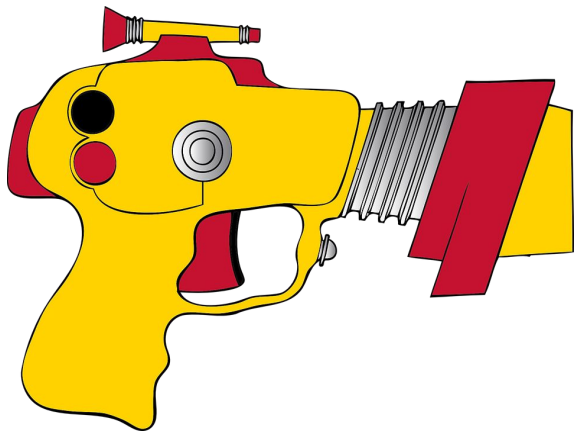
3rd party code

HACK IN BO®
Spring 2018 Edition
10ª EDIZIONE



**Se non puoi
eliminare il bug
almeno puoi
trovarlo**

AACK IN BO®
Spring 2018 Edition
10ª EDIZIONE





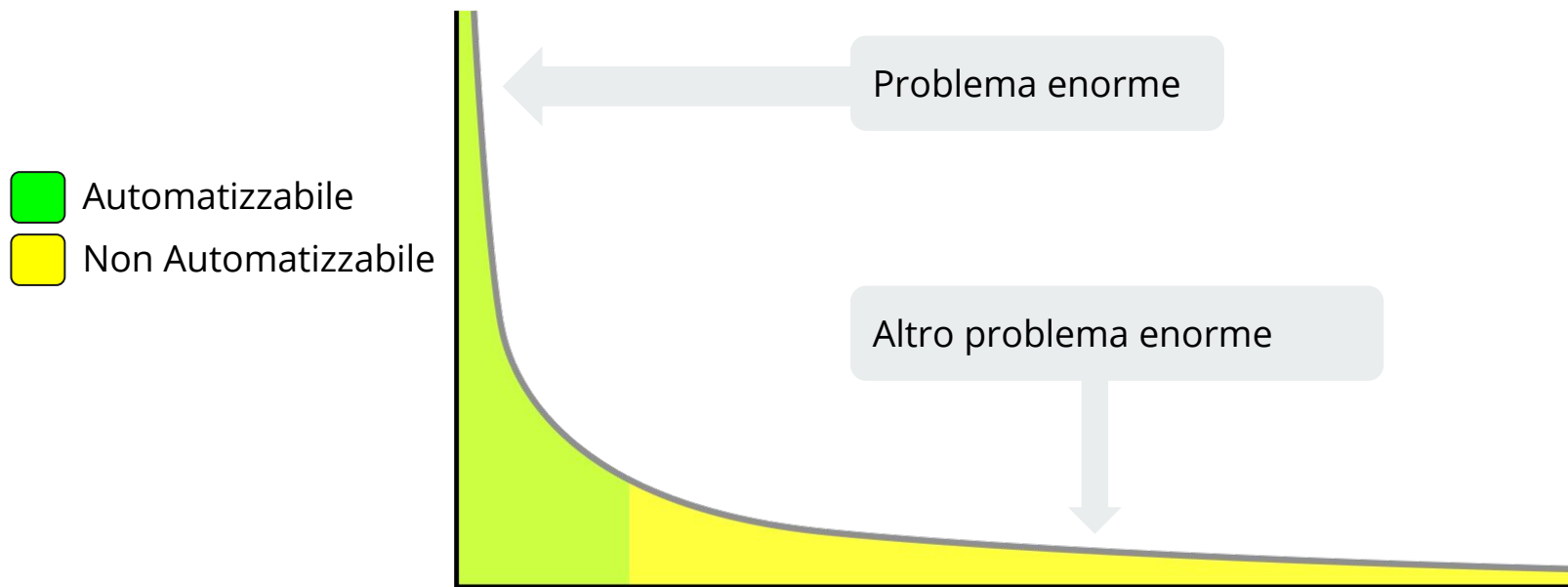
DNA Sequencer

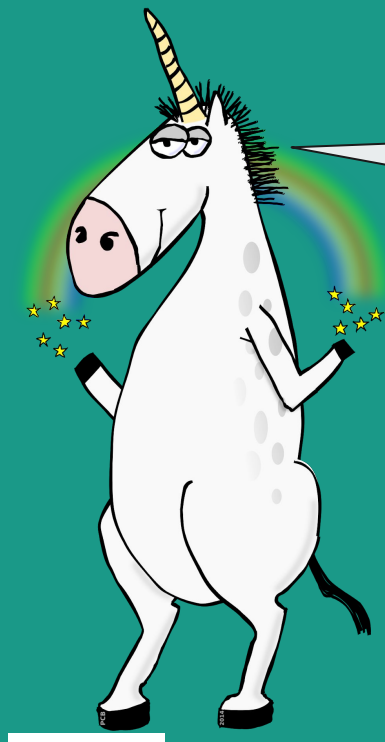
Dispositivi sospesi in alta quota

Rete corporate distribuita
su 50+ nazioni

Datacenters con "un
fracco" di server

Automatizzabile?





Idea: abbiamo un
sacco di ingegneri!

«•»
HACKINBO®
Spring 2018 Edition
10ª EDIZIONE



Aspetta un attimo!

#1 Mi hai appena detto di non usare
il team di sicurezza!

#2 Mi hai anche detto di non
aspettarmi che altri ingegneri
facciano lavoro di security!

The background is a grayscale image of a Super Mario Bros. game screen. At the top, there is a status bar with a health meter, a coin counter showing 'X0', two empty inventory slots, and three heart icons. Below this, a large crowd of small, pixelated human figures fills the scene. In the center, a signpost with a white sign is visible. The sign contains the text 'IT'S DANGEROUS TO GO ALONE! TAKE THIS.' in a pixelated font. The signpost is flanked by two small, pixelated flames.

IT'S DANGEROUS TO GO
ALONE! TAKE THIS.

**Fornire gli
strumenti giusti**

Target:

At least one of these options has to be provided to define the target(s)

-d DIRECT Connection string for direct database connection
-u URL, --url=URL Target URL (e.g. "http://www.site.com/vuln.php?id=1")
-l LOGFILE Parse target(s) from Burp or WebScarab proxy log file
-x SITEMAPURL Parse target(s) from remote sitemap.xml file
-m BULKFILE Scan multiple targets given in a textual file
-r REQUESTFILE Load HTTP request from a file
-g GOOGLEDORK Process Google dork results as target URLs
-c CONFIGFILE Load options from a configuration INI file

Request:

These options can be used to specify how to connect to the target URL

--method=METHOD Force usage of given HTTP method (e.g. PUT)
--data=DATA Data string to be sent through POST
--param-del=PARAM Character used for splitting parameter values
--cookie=COOKIE HTTP Cookie header value
--cookie-del=COO Character used for splitting cookie values
--load-cookies=L File containing cookies in Netscape/wget format
--drop-set-cookie Ignore Set-Cookie header from response
--user-agent=AGENT HTTP User-Agent header value
--random-agent Use randomly selected HTTP User-Agent header value
--host=HOST HTTP Host header value
--referrer=REFERER HTTP Referer header value
--headers=HEADERS Extra headers (e.g. "Accept-Language: fr\nETag: 123")
--auth-type=AUTH HTTP authentication type (Basic, Digest, NTLM or PKI)
--auth-cred=AUTH HTTP authentication credentials (name:password)
--auth-private=A HTTP authentication PEM private key file
--ignore-401 Ignore HTTP Error 401 (Unauthorized)
--proxy=PROXY Use a proxy to connect to the target URL
--proxy-cred=PRO Proxy authentication credentials (name:password)
--proxy-file=PRO Load proxy list from a file
--ignore-proxy Ignore system default proxy settings
--tor Use Tor anonymity network
--tor-port=TORPORT Set Tor proxy port other than default
--tor-type=ORTYPE Set Tor proxy type (HTTP (default), SOCKS4 or SOCKS5)
--check-tor Check to see if Tor is used properly
--delay=DELAY Delay in seconds between each HTTP request
--timeout=TIMEOUT Seconds to wait before timeout connection (default 30)
--retries=RETRIES Retries when the connection timeouts (default 3)
--randomize=RPARAM Randomly change value for given parameter(s)
--safe-url=SAFURL URL address to visit frequently during testing
--safe-freq=SAFREQ Test requests between two visits to a given safe URL

[...]

Optimization:

These options can be used to optimize the performance of sqlmap

-o Turn on all optimization switches
--predict-output Predict common queries output
--keep-alive Use persistent HTTP(s) connections
--null-connection Retrieve page length without actual HTTP response body
--threads=THREADS Max number of concurrent HTTP(s) requests (default 1)

Injection:

These options can be used to specify which parameters to test for, provide custom injection payloads and optional tampering scripts

-p TESTPARAMETER Testable parameter(s)
--skip=SKIP Skip testing for given parameter(s)
--dbms=DBMS Force back-end DBMS to this value
--dbms-cred=DBMS DBMS authentication credentials (user:password)
--os=OS Force back-end DBMS operating system to this value
--invalid-bignum Use big numbers for invalidating values
--invalid-logical Use logical operations for invalidating values
--invalid-string Use random strings for invalidating values
--no-cast Turn off payload casting mechanism
--no-escape Turn off string escaping mechanism
--prefix=PREFIX Injection payload prefix string
--suffix=SUFFIX Injection payload suffix string
--tamper=TAMPER Use given script(s) for tampering injection data

Detection:

These options can be used to customize the detection phase

--level=LEVEL Level of tests to perform (1-5, default 1)
--risk=RISK Risk of tests to perform (0-3, default 1)
--string=STRING String to match when query is evaluated to True
--not-string=NOT String to match when query is evaluated to False
--regex=REGEXP Regexp to match when query is evaluated to True
--code=CODE HTTP code to match when query is evaluated to True
--text-only Compare pages based only on the textual content
--titles Compare pages based only on their titles

Techniques:

These options can be used to tweak testing of specific SQL injection techniques

--technique=TECH SQL injection techniques to use (default "BEUSTQ")
--time-sec=TIMESEC Seconds to delay the DBMS response (default 5)

[...]

Enumeration:

These options can be used to enumerate the back-end database management system information, structure and data contained in the tables. Moreover you can run your own SQL statements

-a, --all Retrieve everything
-b, --banner Retrieve DBMS banner
--current-user Retrieve DBMS current user
--current-db Retrieve DBMS current database
--hostname Retrieve DBMS hostname
--is-dba Detect if the DBMS user is DBA
--users Enumerate DBMS users
--passwords Enumerate DBMS passwords
--privileges Enumerate DBMS privileges
--roles Enumerate DBMS roles
--dbs Enumerate DBMS databases
--tables Enumerate DBMS tables
--columns Enumerate DBMS columns
--schema Enumerate DBMS schema
--count Retrieve number of entries for table(s)
--dump Dump DBMS database table entries
--dump-all Dump all DBMS databases tables entries
--search Search column(s), table(s) and/or database name(s)
--comments Retrieve DBMS comments
-D DB DBMS database to enumerate
-T TBL DBMS database table(s) to enumerate
-C COL DBMS database table column(s) to enumerate
-X EXCLUDECOL DBMS database table column(s) to not enumerate
-U USER DBMS user to enumerate
--exclude-sysdbs Exclude DBMS system databases when enumerating tables
--where=DUMPWHERE Use WHERE condition while table dumping
--start=LIMITSTART First query output entry to retrieve
--stop=LIMITSTOP Last query output entry to retrieve
--first=FIRSTCHAR First query output word character to retrieve
--last=LASTCHAR Last query output word character to retrieve
--sql-query=QUERY SQL statement to be executed
--sql-shell Prompt for an interactive SQL shell
--sql-file=SQLFILE Execute SQL statements from given file(s)

Brute force:

These options can be used to run brute force checks

--common-tables Check existence of common tables
--common-columns Check existence of common columns

User-defined function injection:

These options can be used to create custom user-defined functions

[...]

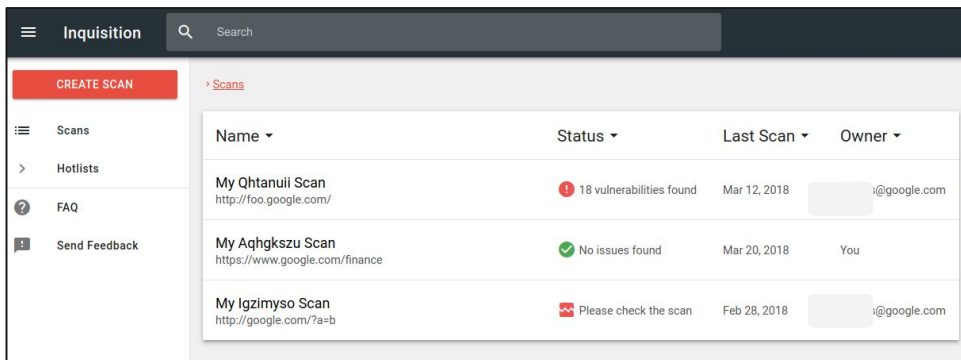


Cosa pensavo servisse

1. Trovare **tutti i tipi di bug** del reame, almeno in qualche istanza
2. Segnali di **bug potenziali**
3. **Configurabilità**



Cosa serve veramente



The screenshot shows the 'Inquisition' web application. On the left is a sidebar with a menu containing 'Scans', 'Hotlists', 'FAQ', and 'Send Feedback'. The main area displays a table of scans under the heading 'Scans'. The table has four columns: 'Name', 'Status', 'Last Scan', and 'Owner'. It lists three scans: 'My Qhtanuii Scan' (18 vulnerabilities found), 'My Aqhgszu Scan' (No issues found), and 'My Igzimyo Scan' (Please check the scan).

Name	Status	Last Scan	Owner
My Qhtanuii Scan http://foo.google.com/	18 vulnerabilities found	Mar 12, 2018	i@google.com
My Aqhgszu Scan https://www.google.com/finance	No issues found	Mar 20, 2018	You
My Igzimyo Scan http://google.com/?a=b	Please check the scan	Feb 28, 2018	i@google.com

1. Usabilità
2. Poche classi di bug, ma in tutte le istanze
3. Nessun falso positivo

Si ma, allora tu cosa fai?

Controllo!

Metriche “di sistema”: quanta copertura sul totale della galassia, tempo medio di risoluzione, etc.



#3

Se non potete prevenire i
bug, nè potete automatizzare
tutto, allora:
Delegare, ma verificare.





HACKINBO®
Spring 2018 Edition
10ª EDIZIONE

Claudio Criscione - Security @ Hyperscale - HackInBo Spring 2018



- #1 Don't detect, prevent
 - #2 Automate: Humans don't scale
 - #3 Delegate & Empower && verify
-

THANK YOU

GRACIAS
ARIGATO
SHUKURIA
GOZAIMASHITA
EFCHARISTO

DANKSCHEEN
JUSPAXAR
BAIKA
SPASSIBO
NUHUN
SNACHALHUYA
CHALTU
YAQHANYELAY
TASHAKKUR ATU
WABEEJA
MAITEKA
HUI
YUSPAGARATAM
SUKSAMA
EKHMET
SPASIBO
DENKAUJA
NEHAACHALHYA
UNALCHEESH
TINGKI
BIYAN
SHUKRIA
MAAKE
GRAZIE
MEHRBANI
PALDIES
BOLZİN
MERCI
MINMONCHAR
MAKETAI
SAIKO
MERASTAWHY
GAEJTHO
AGUYJE
FAKAAUE
KOMAPSUMNIDA
LAH
TAVTAPUCH
MEDAWIAGSE

GRAZIE!

Domande?

