

# Lezione 4

# Vulnerabilità software

Programmazione Sicura (6 CFU), LM Informatica, A. A. 2017/2018

Dipartimento di Scienze Fisiche, Informatiche e Matematiche

Università di Modena e Reggio Emilia

<http://weblab.ing.unimo.it/people/andreolini/didattica/programmazione-sicura>

# Quote of the day

(Meditate, gente, meditate...)

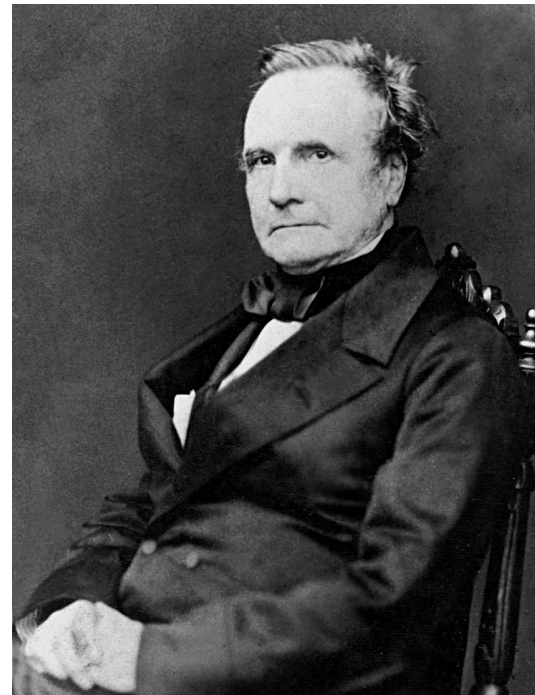
**"On two occasions I have been asked:  
'Pray, Mr. Babbage, if you put into the  
machine the wrong figures, will the right  
answers come out?'**

**I am not able rightly to apprehend the  
kind of confusion of ideas that could  
provoke such a question."**

*Charles Babbage (1791-1871)*

*Matematico, filosofo, inventore, ingegnere meccanico*

*Inventore del primo calcolatore meccanico*



# Memento

(Difetto, bug, debolezza, vulnerabilità, exploit)

**Difetto.** Una qualunque deviazione dalle specifiche.

**Bug.** Un errore di implementazione.

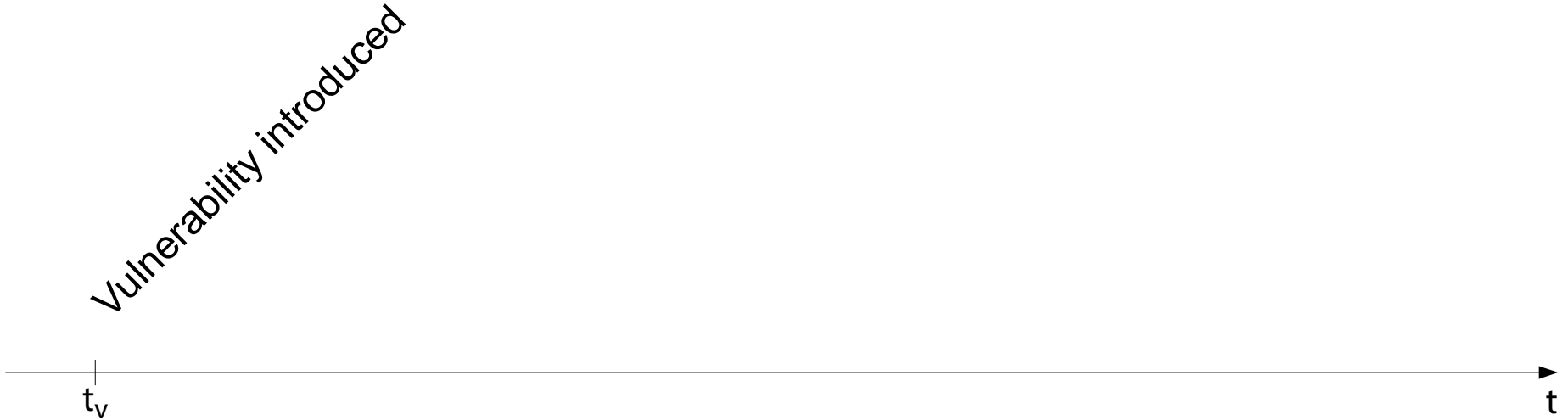
**Debolezza.** difetto o bug che potrebbe, sotto opportune ipotesi, rendere reale una minaccia di sicurezza.

**Vulnerabilità.** Una debolezza presente, comprensibile e sfruttabile da un attaccante.

**Exploit.** Una procedura con cui si evidenzia una vulnerabilità.

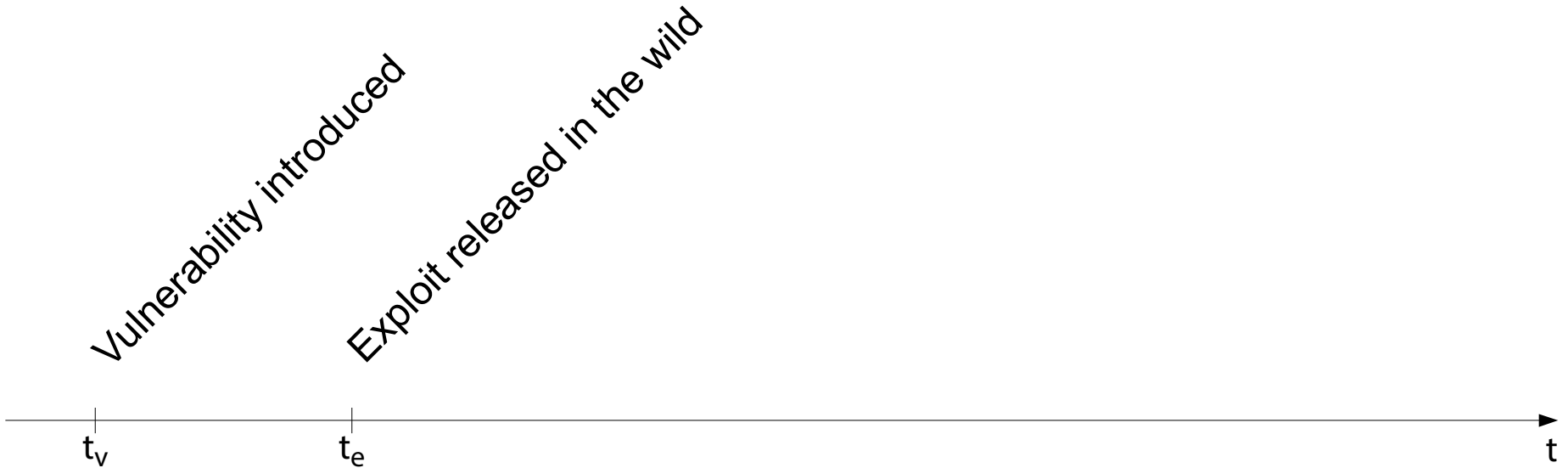
# Ciclo di vita di vulnerabilità software

(Un fornitore rilascia una nuova versione di un software con una vulnerabilità)



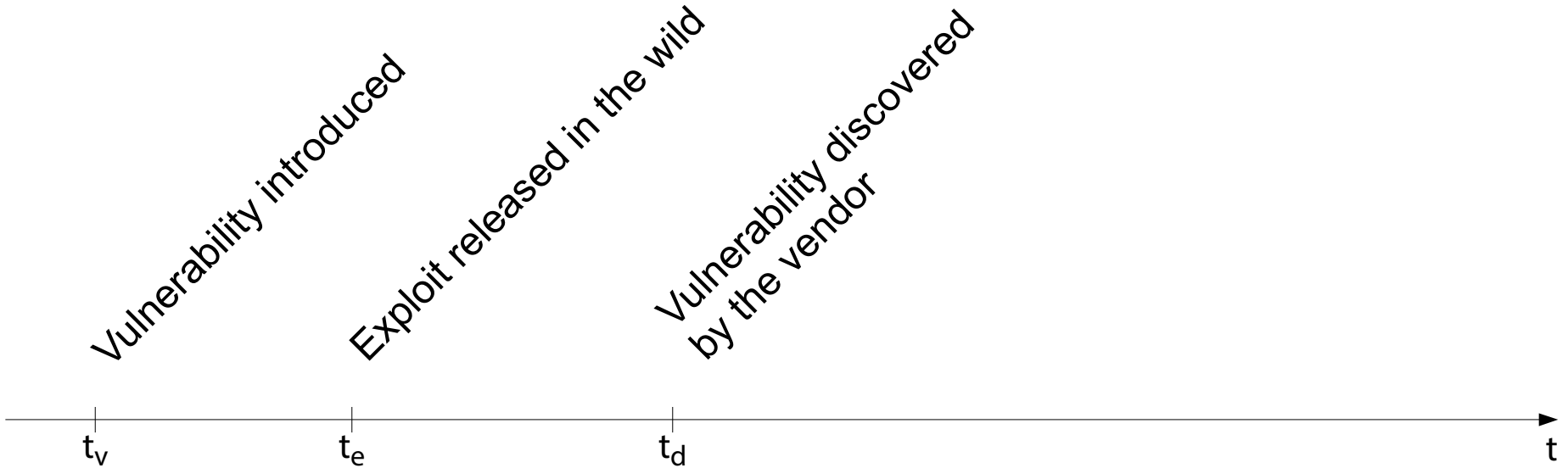
# Ciclo di vita di vulnerabilità software

(Un attaccante rilascia un exploit, non notificando di ciò il fornitore)



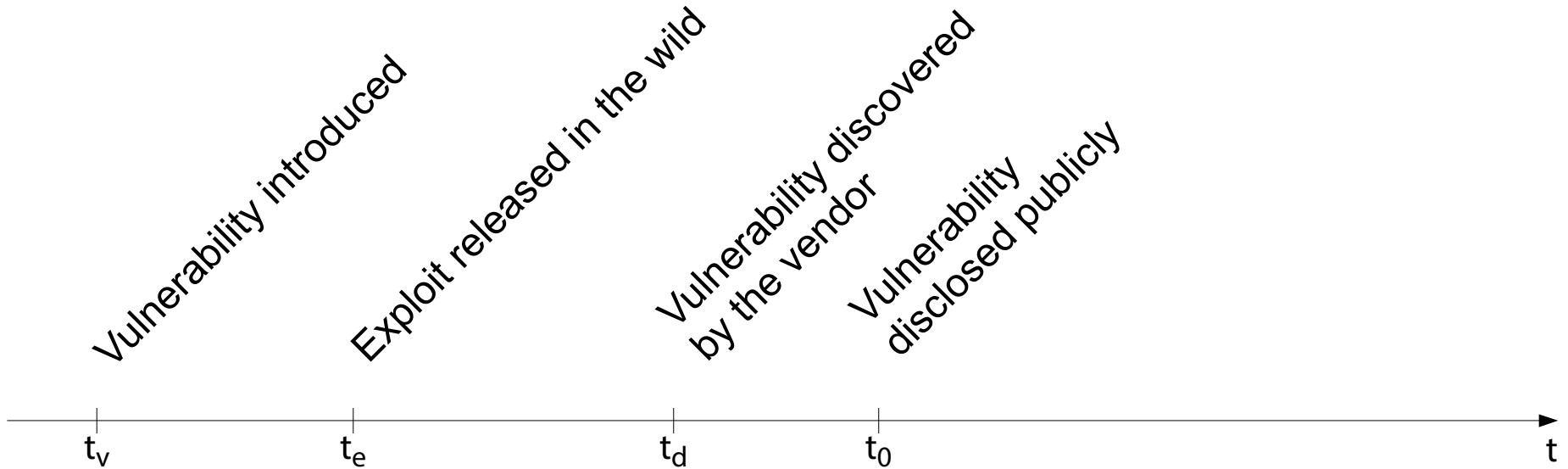
# Ciclo di vita di vulnerabilità software

(Il fornitore si accorge dell'exploit, in proprio o tramite segnalazioni esterne)



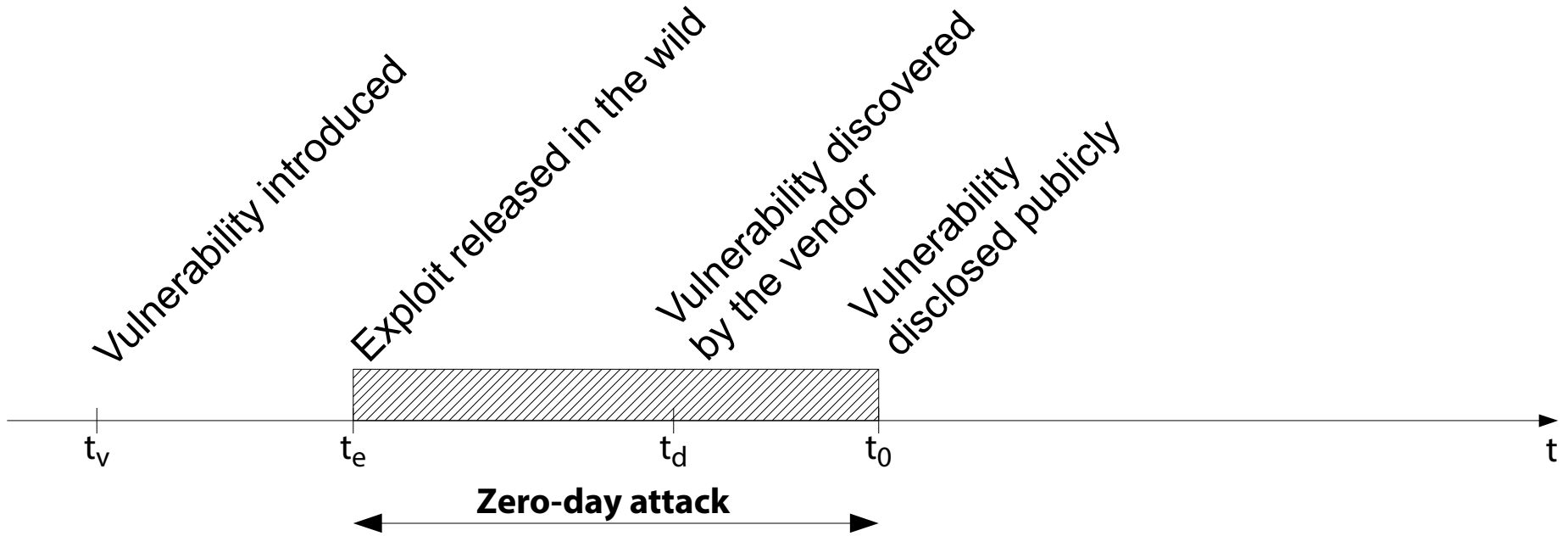
# Ciclo di vita di vulnerabilità software

(La vulnerabilità è divulgata pubblicamente)



# Ciclo di vita di vulnerabilità software

(Nell'intervallo  $[t_e, t_0]$  la vulnerabilità è stata sfruttata "nell'oscurità")



Nell'intero periodo  $[t_e, t_0]$  l'attacco avviene in assenza di una sua pubblica conoscenza ("giorno zero" di presa di coscienza della vulnerabilità). Si parla di **attacco zero-day (zero-day attack)**.



# Etimologia del termine zero-day

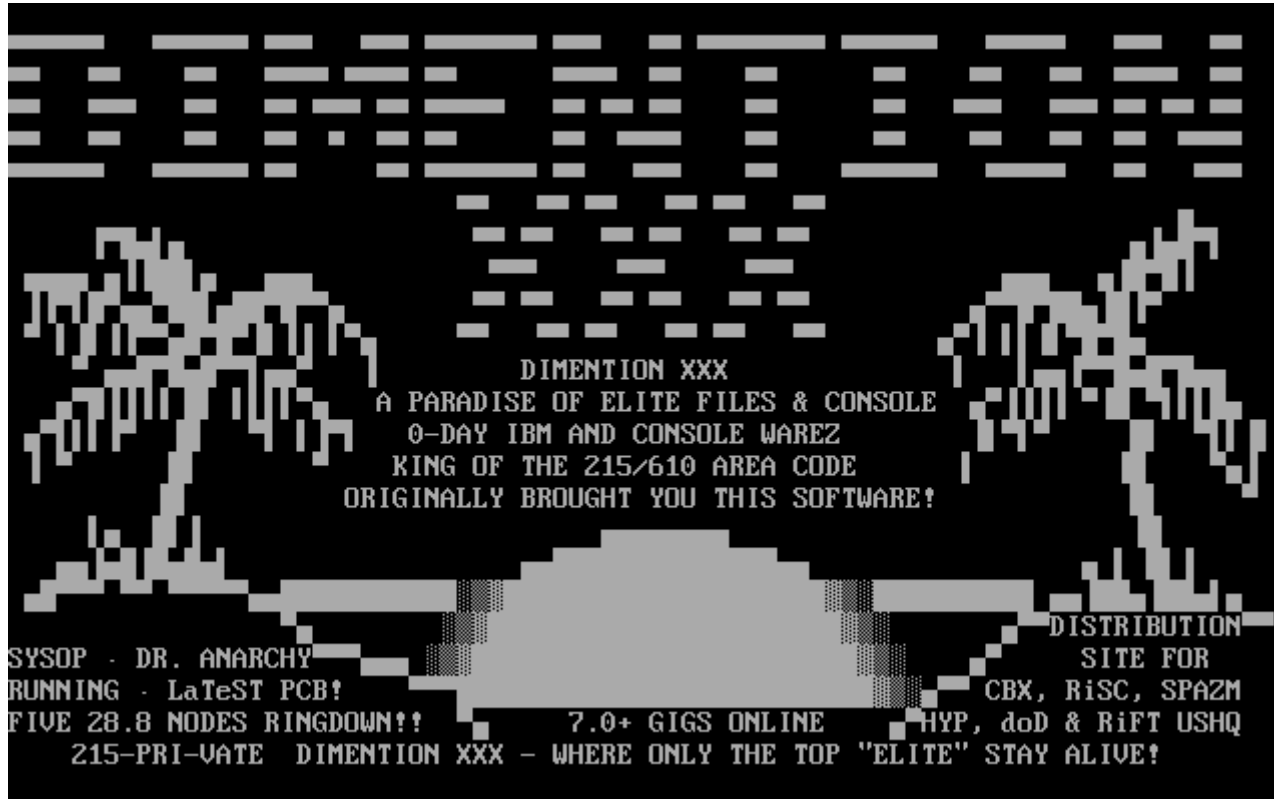
(Warez scene, late '90s)

Termine usato originalmente negli anni '90 dai cracker di software e relativi utenti (**warez scene**).

*"0-day (pronounced as zero day) – This refers to any copyrighted work that has been released the same day as the original product, or sometimes even before. It is considered a mark of skill among warez distro groups to crack and distribute a program on the same day of its commercial release."*

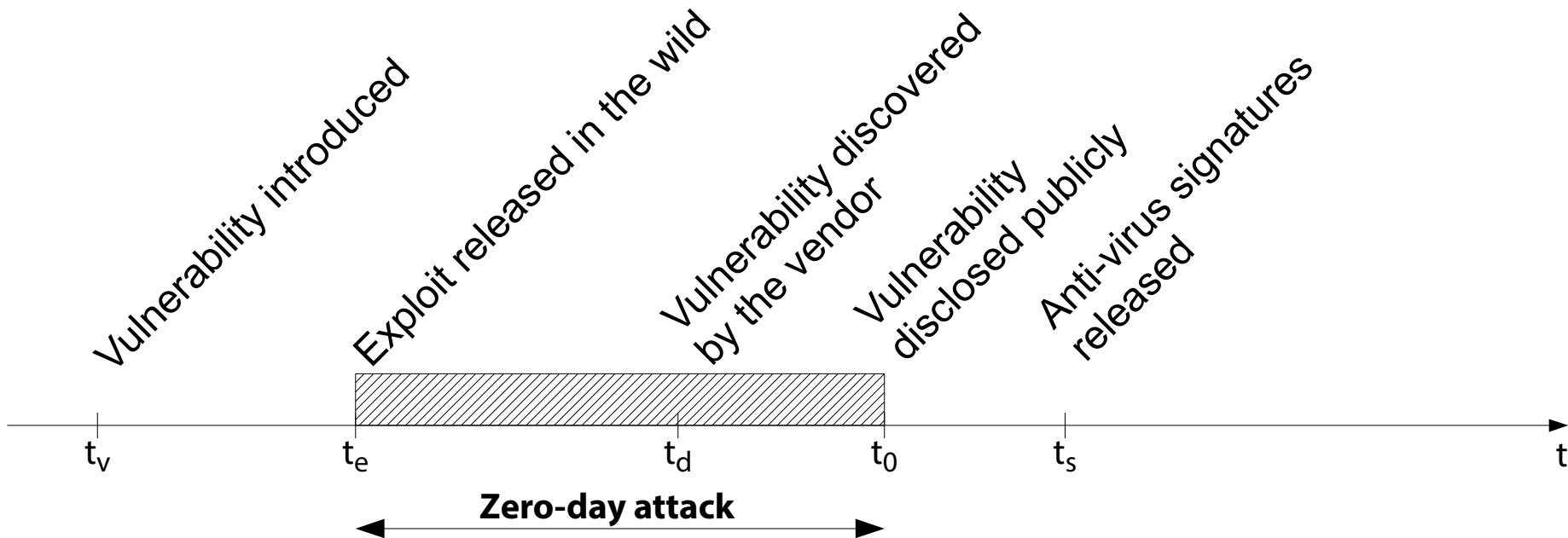
# Etimologia del termine zero-day

(Back in the good ol' days...)



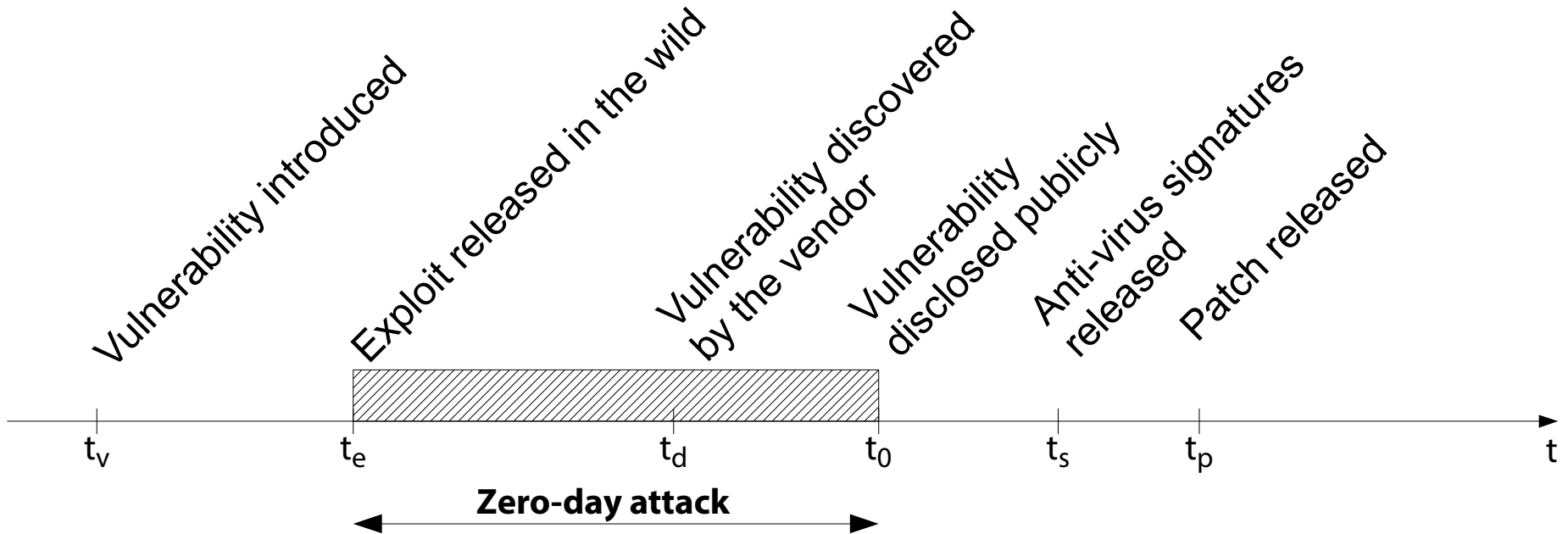
# Ciclo di vita di vulnerabilità software

(Gli anti-virus sono in grado di rilevare l'exploit)



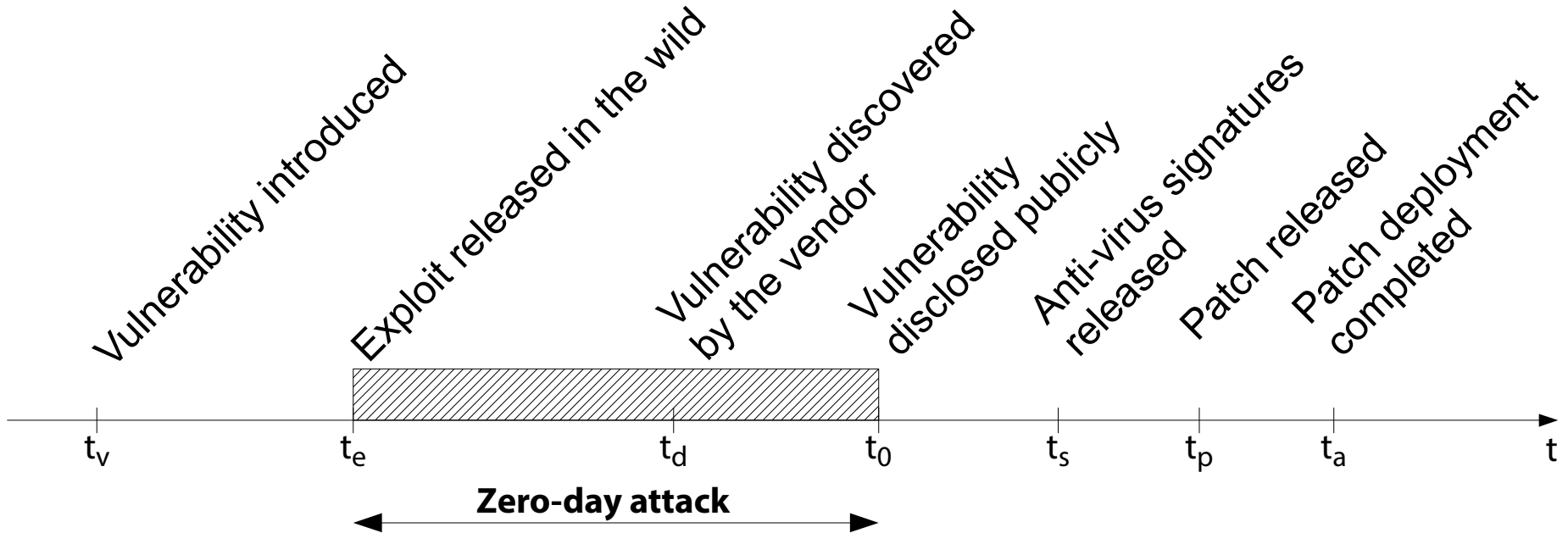
# Ciclo di vita di vulnerabilità software

(Viene rilasciata una correzione pubblica)



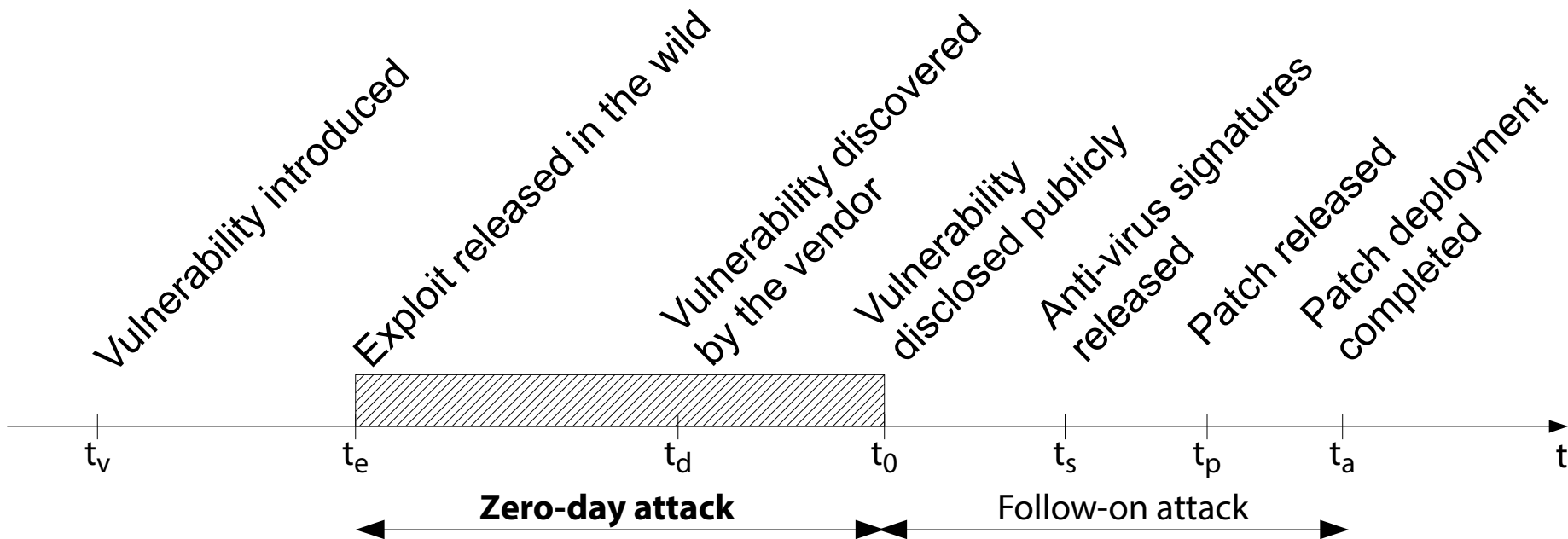
# Ciclo di vita di vulnerabilità software

(L'exploit è mitigato su tutti i sistemi pubblicamente in produzione)



# Ciclo di vita di vulnerabilità software

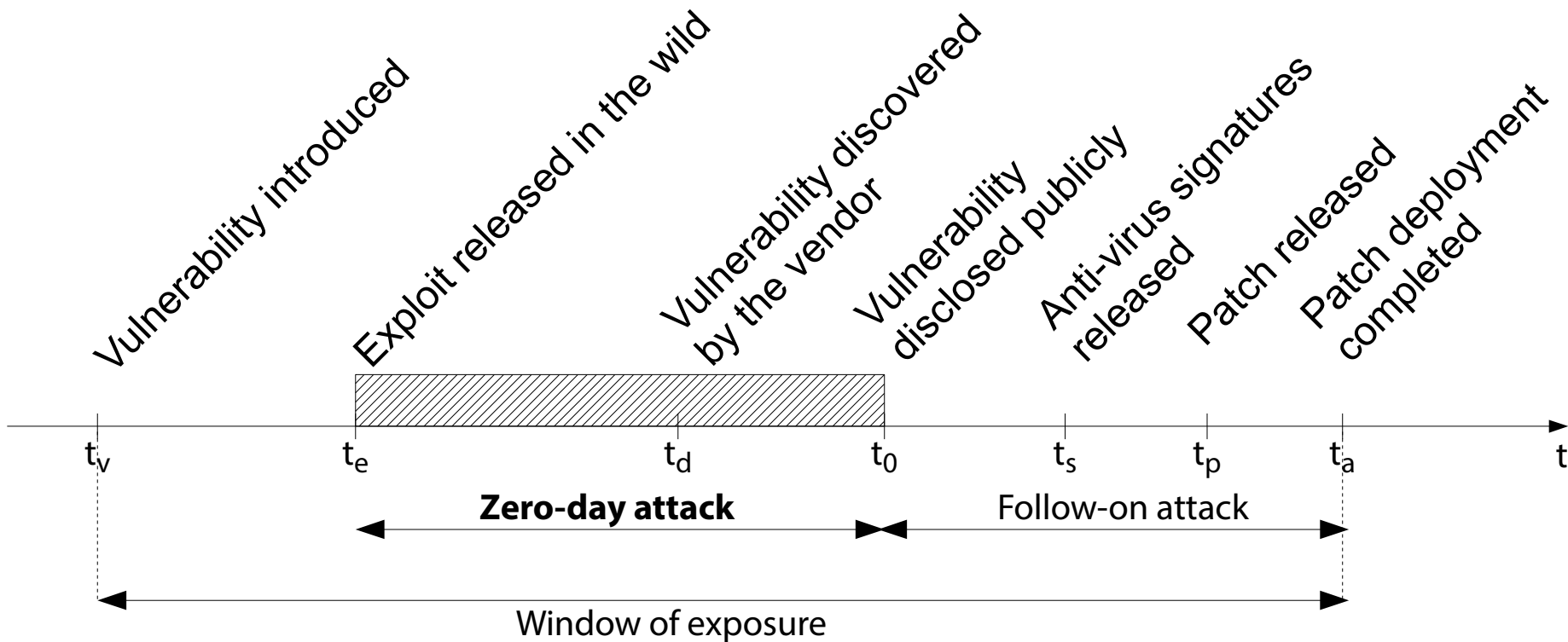
(Nell'intervallo  $[t_0, t_a]$  la vulnerabilità è sfruttata pubblicamente)



Nell'intero periodo  $[t_0, t_a]$  l'attacco avviene in presenza di una sua pubblica conoscenza. La sua forza è notevolmente ridotta rispetto al periodo  $[t_e, t_0]$ .

# Ciclo di vita di vulnerabilità software

(L'intervallo  $[t_v, t_a]$  costituisce la "finestra di esposizione" della vulnerabilità)



# Il costo di un exploit 0-day (Elevatissimo)

## ZERODIUM Payout Ranges \*

LPE: Local Privilege Escalation  
 MTB: Mitigation Bypass  
 RCE: Remote Code Execution  
 RJB: Remote Jailbreak  
 SBX: Sandbox Escape  
 VME: Virtual Machine Escape



\* All payout amounts are chosen at the discretion of ZERODIUM and are subject to change or cancellation without notice.

2015/11 © zerodium.com



# Welcome to the jungle

(It gets worse here everyday)



# Catalogazione delle vulnerabilità

(Un'attività necessaria)

Svariati team di sicurezza scoprono e divulgano le vulnerabilità in modo indipendente.

Ciascun team di sicurezza costruire un proprio archivio storico degli attacchi passati.

**Enumerazione:** costruzione di una tupla univoca a partire da una vulnerabilità. Ad esempio:

(id, tipo vuln., vettore attacco, minaccia, exploit).

**Catalogazione:** inserimento della tupla in un apposito database.

# Problemi di un catalogo non uniforme

(Duplicazione degli sforzi, mancanza di interoperabilità)

In passato, i diversi team hanno costruito altrettanti archivi storici.

Problemi della catalogazione indipendente:

duplicazione dello sforzo (una vulnerabilità è scoperta da più team);

eterogeneità del catalogo (formati diversi, spesso non interoperabili).

# Un esempio: PHF phonebook CGI

(Vulnerability Towel of Babel, 1998)

Nel 1998, la medesima vulnerabilità (PHF phonebook CGI vulnerability) è stata catalogata da diversi team di sicurezza nei dodici modi seguenti.

Table 1 - Vulnerability Tower of Babel, 1998

Organization	Name referring to vulnerability
AXENT (now Symantec)	phf CGI allows remote command execution
BindView	#107-cgi-phf
Bugtraq	PHF Attacks-fun and games for the whole family
CERIAS	http_escshellcmd
CERT	CA-96.06.cgi_example_code
Cisco Systems	HTTP-cgi-phf
CyberSafe	Network: HTTP 'phf' attack
DARPA	0x00000025 = HTTP PHF attack
IBM ERS	ERS-SVA-E01-1996:002.1
ISS	http-cgi-phf
Symantec	#180 HTTP server CGI example code compromises http server
SecurityFocus	#629-phf Remote Command Execution Vulnerability

# The Tower of Babel

(The inevitable consequence)



# La creazione di un catalogo uniforme

(Operata dal MITRE nel 1999)

Per risolvere questo problema, nel 1999 il MITRE ha introdotto un catalogo uniforme di vulnerabilità.

**MITRE:** ente no-profit finanziatore della ricerca a livello governativo.

The MITRE logo is displayed in a bold, blue, sans-serif font. The letters are thick and closely spaced, with a clean, modern appearance.

# Common Vulnerabilities and Exposures

(Un catalogo uniforme di vulnerabilità ed esposizioni)

Il sistema delle **Common Vulnerabilities and Exposures (CVE)** cataloga in modo uniforme vulnerabilità ed esposizioni.

**Vulnerabilità:** una debolezza nel software e/o nel firmware che, se sfruttata, viola almeno una tra confidenzialità, integrità, disponibilità.

**Esposizione:** un errore nel software/nella sua configurazione che permette l'accesso a funzioni ed informazioni.

# Identificatore CVE

(La singola vulnerabilità, opportunamente catalogata)

Il sistema CVE implementa un database di singole vulnerabilità. Ogni vulnerabilità è rappresentata da un **identificatore CVE (CVE id)**.

Formato identificatore: CVE-ANNO-NUMERO.

ANNO: l'anno in cui è stata scoperta la vulnerabilità (4 digit).

NUMERO: un numero intero progressivo.

Un esempio: CVE-2014-6271.



# Scheda CVE

(Le informazioni associate alla vulnerabilità)

Ad ogni CVE id corrisponde una scheda descrittiva contenente informazioni specifiche sulla vulnerabilità.

**Description.** Una descrizione testuale del problema insito nella vulnerabilità.

**References.** Un URL descrivente in maggiore dettaglio il problema (ve ne può essere più di uno).

**Date Entry Created.** La data di inserimento del CVE nel database.

# Scheda CVE

(Campi obsoleti nella scheda)

Alcuni campi della scheda sono obsoleti e non più aggiornati.

Phase.

Votes.

Comments.

Proposed.

# L'archivio CVE

(Interrogabile dal sito <https://cve.mitre.org>)

Il sito <https://cve.mitre.org> ospita l'archivio CVE, informazioni sul progetto ed un form di ricerca.

Cliccando sul link "Search CVE List" si accede alle funzioni di ricerca.

È possibile cercare uno specifico CVE (form "By CVE Identifier") o per parola chiave (form "By Keyword(s)").

Ad esempio, si ricerchi la scheda della vulnerabilità CVE-2014-6271.

# La scheda di CVE-2014-6271

(CVE id, Description)

Il CVE id

CVE-ID	
<b>CVE-2014-6271</b>	<a href="#">Learn more at National Vulnerability Database (NVD)</a> • Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings
Description	
GNU Bash through 4.3 processes trailing strings after function definitions in the values of environment variables, which allows remote attackers to execute arbitrary code via a crafted environment, as demonstrated by vectors involving the ForceCommand feature in OpenSSH sshd, the mod_cgi and mod_cgid modules in the Apache HTTP Server, scripts executed by unspecified DHCP clients, and other situations in which setting the environment occurs across a privilege boundary from Bash execution, aka "ShellShock." NOTE: the original fix for this issue was incorrect; CVE-2014-7169 has been assigned to cover the vulnerability that is still present after the incorrect fix.	

Una descrizione dettagliata della vulnerabilità.

# La scheda di CVE-2014-6271

## (References)

Un elenco di URL descrivente la vulnerabilità in maggiore dettaglio.  
URL diversi sono redatti da team di sicurezza diversi:  
associati al software;  
associati alla distribuzione;  
Indipendenti.

### References

**Note:** [References](#) are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- BUGTRAQ:20141001 NEW VMSA-2014-0010 - VMware product updates address critical Bash security vulnerabilities
- [URL:http://www.securityfocus.com/archive/1/archive/1/533593/100/0/threaded](http://www.securityfocus.com/archive/1/archive/1/533593/100/0/threaded)
- EXPLOIT-DB:39918
- [URL:https://www.exploit-db.com/exploits/39918/](https://www.exploit-db.com/exploits/39918/)
- FULLDISC:20141001 FW: NEW VMSA-2014-0010 - VMware product updates address critical Bash security vulnerabilities
- [URL:http://seclists.org/fulldisclosure/2014/Oct/0](http://seclists.org/fulldisclosure/2014/Oct/0)
- [MISC:http://lcamtuf.blogspot.com/2014/09/quick-notes-about-bash-bug-its-impact.html](http://lcamtuf.blogspot.com/2014/09/quick-notes-about-bash-bug-its-impact.html)
- [MISC:http://packetstormsecurity.com/files/128517/VMware-Security-Advisory-2014-0010.html](http://packetstormsecurity.com/files/128517/VMware-Security-Advisory-2014-0010.html)
- [MISC:http://packetstormsecurity.com/files/128567/CA-Technologies-GNU-Bash-Shellshock.html](http://packetstormsecurity.com/files/128567/CA-Technologies-GNU-Bash-Shellshock.html)
- [MISC:http://packetstormsecurity.com/files/128573/Apache-mod\\_cgi-Remote-Command-Execution.html](http://packetstormsecurity.com/files/128573/Apache-mod_cgi-Remote-Command-Execution.html)
- [MISC:http://packetstormsecurity.com/files/137376/IPFire-Bash-Environment-Variable-Injection-Shellshock.html](http://packetstormsecurity.com/files/137376/IPFire-Bash-Environment-Variable-Injection-Shellshock.html)

...

# La scheda di CVE-2014-6271

## (References)

La data di creazione del CVE id,  
in formato YYYYMMDD.



Date Entry Created	
<b>20140909</b>	Disclaimer: The entry creation date may reflect when the CVE-ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

# La scheda di CVE-2014-6271

(I campi obsoleti Phase, Votes, Comments, Proposed)

Campi obsoleti (legacy),  
non più aggiornati.



<b>Phase (Legacy)</b>
Assigned (20140909)
<b>Votes (Legacy)</b>
<b>Comments (Legacy)</b>
<b>Proposed (Legacy)</b>
N/A

# La funzionalità di BASH in questione

(Funzioni di ambiente, analoghe alle variabili)

BASH permette di esportare variabili e funzioni di ambiente, rendendole disponibili alle shell figlie.

```
$ export foo=' () { echo "In foo"; }'
```

```
$ bash -c 'foo'
```

```
In foo
```

Esportazione di  
una funzione

Invocazione di  
una shell figlia

Output  
della funzione



# Il difetto

(Se `foo` è invocata, viene valutata ed eseguita l'intera riga)

Sfortunatamente, se richiesto, BASH valuta non solo `foo`, bensì l'intera linea.

```
$ export foo='() { echo "In foo"; };  
echo vulnerable  
$ bash -c `foo`  
vulnerable  
In foo
```

# L'attivazione del difetto

(Avviene invocando `foo`)

Sfortunatamente, se richiesto, BASH valuta non solo `foo`, bensì l'intera linea.

```
$ export foo='() { echo "In foo"; };  
echo vulnerable'
```

```
$ bash -c `foo`
```

```
vulnerable
```

```
In foo
```

Viene invocata la funzione `foo`.

# L'effetto collaterale del difetto

(Esecuzione arbitraria di codice)

Sfortunatamente, se richiesto, BASH valuta non solo `foo`, bensì l'intera linea.

```
$ export foo='() { echo "In foo"; };
```

```
echo vulnerable'②
```

```
$ bash -c `foo`
```

```
vulnerable
```

```
In foo
```

Viene eseguito lo statement  
`echo vulnerable`.

# L'operazione normale

(L'esecuzione dello statement dentro `foo`)

Sfortunatamente, se richiesto, BASH valuta non solo `foo`, bensì l'intera linea.

```
$ export foo='() { echo "In foo"; };  
echo vulnerable'  
$ bash -c `foo`  
vulnerable  
In foo
```

③ Viene eseguito lo statement `echo "In foo"`.

# Well, it's just a damn parser bug, right?

(What could possibly go wrong? Dude, calm down!)



# Sfruttamento tramite vettore locale

(Si può forzare l'esecuzione di un comando ad ogni partenza di BASH)

Un attaccante con accesso ad un terminale può iniettare nel file di inizializzazione `.bashrc` la seguente definizione di funzione di ambiente:

```
export foo=' () { echo "In foo"; }; evil_command'
```

Ogni volta che l'utente vittima apre un terminale, il comando `evil_command` viene eseguito.

# Debolezza o vulnerabilità?

(Quale delle due?)

Tramite una semplice procedura, un attaccante può eseguire comandi che, in condizioni normali, non ha il permesso di eseguire.

L'attaccante non ha sempre la chance di sedersi di fronte al terminale della vittima.

Che cosa è questa, se non una privilege escalation (la **E** di STRIDE)?

# Vulnerabilità

(Of course!)

Debolezza accessibile

+ procedura di sfruttamento (exploit)

=

**VULNERABILITÀ!**



# The bug is locally exploitable

(You dirty little tw\*\*!)



# Well, but it's not remotely, right?

(Who cares, as long as no one gets access to a terminal? Calm down, dude!)



# Sfruttamento tramite vettore remoto

(Si può forzare l'esecuzione di un comando interagendo con un Web server)

Ogni server remoto che accetta in ingresso codice di BASH e lo valuta senza controllarlo è potenzialmente sfruttabile.

Ad esempio il Web server Apache, quando esegue uno script CGI scritto in BASH, salva gli header della richiesta in apposite variabili di ambiente e le valuta.

È sufficiente costruire una linea di BASH maligna e passarla come header di una richiesta per sfruttare la vulnerabilità (anche da remoto!).

# Un esempio indicativo

(Va adattato ad un caso reale)

Si può usare il client HTTP `curl` per “iniettare” un header a caso, malformato in modo tale da provocare l’esecuzione di `evil_command`.

```
$ curl -v http://server/cgi-bin/bashcgi -H  
"custom:() { ;; } ; evil_command"
```

# Un esempio indicativo

(Esecuzione di uno script CGI in BASH via `curl`)

Si può usare il client HTTP `curl` per “iniettare” un header a caso, malformato in modo tale da provocare l’esecuzione di `evil_command`.

```
$ curl -v http://server/cgi-bin/bashcgi -H  
"custom: () { ;; } ; evil_command"
```

① Viene invocato il CGI scritto in BASH.

# Un esempio indicativo

(Aggiunta di un header malizioso)

Si può usare il client HTTP `curl` per “iniettare” un header a caso, malformato in modo tale da provocare l’esecuzione di `evil_command`.

```
$ curl -v http://server/cgi-bin/bashcgi -H  
"custom:() { ;; } ; evil_command"
```

- ② Viene passato un header HTTP di nome “custom” e con un valore tale da provocare la vulnerabilità (se valutato in ambito BASH).

# Un esempio indicativo

(L'header è trasformato in una variabile di ambiente e passato allo script)

Si può usare il client HTTP `curl` per “iniettare” un header a caso, malformato in modo tale da provocare l'esecuzione di `evil_command`.

```
$ curl -v http://server/cgi-bin/bashcgi -H  
"custom:() { ;; } ; evil_command"
```

- ③ Come da specifica CGI (RFC3875), Apache crea una variabile di ambiente `HTTP_CUSTOM` e la aggiunge all'ambiente dello script `bashcgi`.

# Un esempio indicativo

(Quando lo script esegue, valuta la variabile di ambiente)

Si può usare il client HTTP `curl` per “iniettare” un header a caso, malformato in modo tale da provocare l’esecuzione di `evil_command`.

```
$ curl -v http://server/cgi-bin/bashcgi -H  
"custom:() { ;; } ; evil_command"
```

---

- ④ Non appena lo script `bashcgi` esegue, la variabile di ambiente `HTTP_CUSTOM` è valutata ed il comando `evil_command` è eseguito sul server con i diritti dell’utente con cui esegue Apache.



# The bug is remotely exploitable

(You f\*\*\*in' m\*\*\*n!)



Holy s\*\*t!  
(Hope it's not too late!)



# SHELLSHOCK

(The more popular name of CVE-2014-6271)



# SHELLSHOCK

(It made the news in 2014)

BBC ID Menu Search

## NEWS

Sections

### Shellshock: 'Deadly serious' new vulnerability found

By Dave Lee  
Technology reporter, BBC News

© 25 September 2014 | Technology

## Schneier on Security

Blog Newsletter Books Essays News Talks

[Blog >](#)

### Nasty Vulnerability found in Bash

[It's a big and nasty one.](#)

CNN tech business culture gadgets future startups search

## 'Bash' bug could let hackers attack through a light bulb

by Jose Pagliery @Jose\_Pagliery

September 25, 2014: 12:54 PM ET

Recommend 2.4K f t in

### Bash gets shellshocked

By Jake Edge  
October 1, 2014

It's been a crazy week for the [Bash shell](#), its maintainer, and many Linux distributions that use the shell. A remote code-execution vulnerability that was [reported](#) on September 24 has now morphed into multiple related vulnerabilities, which have now mostly been fixed and updates released by distributions. The vulnerabilities have been dubbed "Shellshock" and the technical (and mainstream) press has had a field day reporting on the incident. It all revolves around a somewhat dubious Bash feature, but the widespread use of Bash in places where it may not really make sense contributed to the severity of the bug.

# SHELLSHOCK

(Even in Italy!)



Tecnologia

Home	News	Speciali	Mobile	Social Network	Sicurezza	Prodotti	Interattivi	Video
------	------	----------	--------	----------------	-----------	----------	-------------	-------

[Consiglia](#) [Condividi](#) 371 [Tweet](#) [G+](#) 18 [LinkedIn](#) 40

"Shellshock", il **virus** che minaccia i sistemi Linux ed Apple

```
example]# ./example.sh -a
>>> [DEBUG] (example.lib.sh:49), __init(): command line argument: -a
>>> [DEBUG] (bashinator.lib.0.sh:151), __prepare(): successfully created temporary script subcommand logfl
>>> [DEBUG] (bashinator.lib.0.sh:159), __prepare(): script subcommand logfile: '/tmp/example.log.6DUe0Y'
>>> [DEBUG] (example.lib.sh:76), __main(): this is a debug test
>>> [INFO] (example.lib.sh:76), __main(): this is a info test
>>> [NOTICE] (example.lib.sh:76), __main(): this is a notice test
!!! [WARNING] (example.lib.sh:76), __main(): this is a warning test
!!! [ERROR] (example.lib.sh:76), __main(): this is a err test
!!! [CRITICAL] (example.lib.sh:76), __main(): this is a crit test
!!! [ALERT] (example.lib.sh:76), __main(): this is a alert test
!!! [EMERGENCY] (example.lib.sh:76), __main(): this is a emerg test
>>> [DEBUG] (example.lib.sh:117), exampleFunction(): fooArgument: foo
>>> [DEBUG] (example.lib.sh:129), exampleFunction(): barArgument: bar
>>> [INFO] (example.lib.sh:124), exampleFunction(): this is an example function
!!! [ALERT] (example.lib.sh:138), bazFunction(): FATAL: dying for test purposes
!!! [ALERT] (example.lib.sh:138), bazFunction(): function call stack (most recent last):
!!! [ALERT] (example.lib.sh:138), bazFunction(): -> __dispatch('-a') called in 'example.sh' on line 71
!!! [ALERT] (example.lib.sh:138), bazFunction(): -> __main() called in 'bashinator.lib.0.sh' on line 115
!!! [ALERT] (example.lib.sh:138), bazFunction(): -> FooFunction('fooArgs') called in 'example.lib.sh' on
!!! [ALERT] (example.lib.sh:138), bazFunction(): -> barFunction('barArgs') called in 'example.lib.sh' on
```

# Stephane Chazelas (1975-)

(The hacker who discovered SHELLSHOCK)

Ingegnere informatico.

Esperto di UNIX/Linux.

Esperto di telecomunicazioni.



# Domanda

(Spontanea, se avete seguito fino a questo punto)

Quale di questi due vettori di attacco preoccupa di più per semplicità d'uso?

**Vettore di attacco locale.** L'attaccante deve avere accesso ad un terminale aperto e non custodito per il tempo necessario a modificare il file `.bashrc`.

**Vettore di attacco remoto.** L'attaccante deve inviare una richiesta HTTP maliziosa al server.

# Risposta

(Shame on you if you answered 'local'!)

Quale di questi due vettori di attacco preoccupa di più per semplicità d'uso?

**Vettore di attacco locale.** L'attaccante deve avere accesso ad un terminale aperto e non custodito per il tempo necessario a modificare il file `.bashrc`.

**Vettore di attacco remoto.** L'attaccante deve inviare una richiesta HTTP maliziosa al server.



# Il limite del sistema CVE

(Abbiatene pietà; CVE non era nato per tutto questo)

Il sistema CVE enumera le vulnerabilità.

Il sistema CVE NON misura l'impatto di una vulnerabilità (non è nato per questo scopo).

Alcune (tristi) conseguenze:

dati due CVE, non si è in grado di dire quale dei due sia più urgente da gestire;

lo stesso CVE può avere un impatto diverso nel tempo;

lo stesso CVE può avere impatti diversi in sistemi diversi.

# Common Vulnerability Scoring System

(CVSS; misura la gravità di una vulnerabilità)

Il **Common Vulnerability Scoring System (CVSS)** è un sistema di stima della gravità di una vulnerabilità.

Ad ogni CVE id è assegnato un **punteggio (score)** da 0 a 10.

0: impatto nullo.

(0, 4): impatto basso.

[4, 7): impatto medio.

[7, 9): impatto elevato.

[9, 10]: impatto critico.

0-1
1-2
2-3
3-4
4-5
5-6
6-7
7-8
8-9
9-10

# Versioni del CVSS

(CVSS v2, CVSS v3)

Le versioni del sistema CVSS correntemente in uso sono due.

**Versione 2 (v2):** introdotta nel 2005, pubblicata nel 2007.

<https://www.first.org/cvss/v2/guide>

**Versione 3 (v3):** introdotta nel 2012, pubblicata nel 2015.

<https://www.first.org/cvss/specification-document>

Affronta i limiti di CVSS v2.

Non è ancora diffusa su larga scala.

# Una avvertenza

(CVSS v2  $\neq$  CVSS v3)

Nel seguito si fornisce una sintetica descrizione del CVSS v2.

**NOTA BENE:** i punteggi CVSS v2 e CVSS v3 variano leggermente. Ci si informi bene sulla natura del punteggio offerto per l'analisi!

# Le metriche CVSS

(Metrica: grandezza da misurare tramite una procedura)

Il punteggio CVSS è composto di tre gruppi di metriche:  
base (**Base**);  
temporali (**Temporal**);  
ambientali (**Environmental**).

## Base Metric Group

Access Vector

Confidentiality  
Impact

Access Complexity

Integrity  
Impact

Authentication

Availability  
Impact

## Temporal Metric Group

Exploitability

Remediation Level

Report  
Confidence

## Environmental Metric Group

Collateral Damage  
Potential

Confidentiality  
Requirement

Target  
Distribution

Integrity  
Requirement

Availability  
Requirement

# Metriche "Base"

(Stimano la gravità della vulnerabilità in sé)

**Metriche Base:** stimano la gravità della vulnerabilità in sé, a prescindere da fattori temporali ed ambientali.

Da dove si riesce a sfruttare? Quanto è semplice metterla in atto? Che cosa permette di ottenere?

## Base Metric Group

Access Vector

Confidentiality  
Impact

Access Complexity

Integrity  
Impact

Authentication

Availability  
Impact

## Temporal Metric Group

Exploitability

Remediation Level

Report  
Confidence

## Environmental Metric Group

Collateral Damage  
Potential

Confidentiality  
Requirement

Target  
Distribution

Integrity  
Requirement

Availability  
Requirement

# Metriche “Temporal”

(Stimano la gravità della vulnerabilità dal punto di vista temporale)

**Metriche Temporal:** stimano la gravità della vulnerabilità dal punto di vista temporale.

È disponibile un exploit? Sono disponibili patch/fix?

## Base Metric Group

Access Vector

Confidentiality Impact

Access Complexity

Integrity Impact

Authentication

Availability Impact

## Temporal Metric Group

Exploitability

Remediation Level

Report Confidence

## Environmental Metric Group

Collateral Damage Potential

Confidentiality Requirement

Target Distribution

Integrity Requirement

Availability Requirement

# Metriche “Environmental”

(Stimano la gravità della vulnerabilità dal punto di vista environmental)

**Metriche Environmental:** stimano la gravità della vulnerabilità dal punto di vista ambientale.

Qual è la conseguenza di un exploit su persone e cose?  
Quanti sistemi dell’infrastruttura sono vulnerabili?

## Base Metric Group

Access Vector

Confidentiality  
Impact

Access Complexity

Integrity  
Impact

Authentication

Availability  
Impact

## Temporal Metric Group

Exploitability

Remediation Level

Report  
Confidence

## Environmental Metric Group

Collateral Damage  
Potential

Confidentiality  
Requirement

Target  
Distribution

Integrity  
Requirement

Availability  
Requirement



# Calcolo del punteggio

(Concettualmente semplice)

Ad ogni metrica è associata una domanda con una risposta chiusa.

Si risponde alla domanda scegliendo la risposta più consona nel modo più oggettivo possibile.

Le risposte alle domande dei questionari forniscono dei pesi numerici.

I pesi numerici sono usati per calcolare un punteggio finale tramite una serie di formule.

→ Approccio ingegneristico (non matematico). 65

# Base: Access Vector (AV)

(Local, adjacent network, network)

Tramite quale vettore di accesso può essere sfruttata una vulnerabilità?

## Base Metric Group

Access Vector

Confidentiality  
Impact

Access Complexity

Integrity  
Impact

Authentication

Availability  
Impact

Valore	Descrizione	Punt.
Local (L)	L'attaccante deve avere accesso fisico/un account sul sistema.	0.395
Adjacent Network (A)	L'attaccante deve avere accesso al dominio di broadcast o di collisione del sistema.	0.646
Network (N)	L'interfaccia vulnerabile è al livello 3 o superiore della pila ISO/OSI.	1.0

# Base: Access Vector (AV)

**OSS.:** la metrica e ciascuna risposta hanno una abbreviazione.

Access Vector → AV.

Local → L.

Adjacent Network → A.

Network → N.


Tenete a mente questo dettaglio.

Local (L)		
Adjacent Network (A)		
Network (N)		

# Base: Access Vector (AV)

**OSS.:** i punteggi parziali nell'ultima colonna sono usati per calcolare il punteggio finale. Più alto è il punteggio parziale, più è grave la vulnerabilità dal punto di vista considerato.

Local (L)		0.395
Adjacent Network (A)		0.646
Network (N)		1.0



**NON IMPARATE  
IL QUESTIONARIO  
A MEMORIA!**

# Base: Access Complexity (AC)

(High, medium, low)

Qual è la difficoltà di sfruttamento della vulnerabilità?

## Base Metric Group

Access Vector

Confidentiality  
Impact

Access Complexity

Integrity  
Impact

Authentication

Availability  
Impact

Valore	Descrizione	Punt.
High (H)	Lo sfruttamento richiede condizioni particolari (corsa critica, tecniche di social engineering).	0.35
Medium (M)	Lo sfruttamento richiede alcune condizioni (ad es., configurazione non di default).	0.646
Low (L)	Lo sfruttamento non richiede nulla di particolare (funziona su sistemi standard).	1.0

# Base: Authentication (Au)

(Multiple, single, none)

Quante volte si deve autenticare un attaccante per sfruttare la vulnerabilità?

## Base Metric Group

Access Vector

Confidentiality  
Impact

Access Complexity

Integrity  
Impact

Authentication

Availability  
Impact

Valore	Descrizione	Punt.
Multiple (M)	Lo sfruttamento richiede due o più autenticazioni (anche con le stesse credenziali).	0.45
Single (S)	Lo sfruttamento richiede una sola autenticazione.	0.56
None (N)	Lo sfruttamento non richiede alcuna forma di autenticazione.	0.704

# Base: Confidentiality Impact (C)

(None, partial, complete)

Qual è l'impatto della vulnerabilità sull'attributo di confidenzialità del sistema?

## Base Metric Group

Access Vector

Confidentiality  
Impact

Access Complexity

Integrity  
Impact

Authentication

Availability  
Impact

Valore	Descrizione	Punt.
None (N)	Non vi è impatto alcuno.	0.0
Partial (P)	É possibile divulgare solo un sotto-insieme dei dati offerti dal sistema.	0.275
Complete (C)	É possibile divulgare l'intero insieme dei dati offerti dal sistema.	0.660



# Base: Integrity Impact (I)

(None, partial, complete)

Qual è l'impatto della vulnerabilità sull'attributo di integrità del sistema?

## Base Metric Group

Access Vector

Confidentiality  
Impact

Access Complexity

Integrity  
Impact

Authentication

Availability  
Impact

Valore	Descrizione	Punt.
None (N)	Non vi è impatto alcuno.	0.0
Partial (P)	É possibile modificare solo un sotto-insieme dei dati offerti dal sistema.	0.275
Complete (C)	É possibile modificare l'intero insieme dei dati offerti dal sistema.	0.660

# Base: Availability Impact (A)

(None, partial, complete)

Qual è l'impatto della vulnerabilità sull'attributo di disponibilità del sistema?

## Base Metric Group

Access Vector

Confidentiality  
Impact

Access Complexity

Integrity  
Impact

Authentication

Availability  
Impact

Valore	Descrizione	Punt.
None (N)	Non vi è impatto alcuno.	0.0
Partial (P)	É possibile ridurre parzialmente le prestazioni e/o le funzioni offerte dal sistema.	0.275
Complete (C)	É possibile ridurre completamente le prestazioni e/o le funzioni offerte dal sistema.	0.660

# Calcolo del punteggio "Base"

(Tramite un insieme perverso di formule)

Volendo, è già possibile calcolare un **Punteggio Base (Base Score)** che stima la gravità di una vulnerabilità, trascurando i fattori tempo ed ambiente.

$$Exploitability = 20 * AccessVector * AccessComplexity * Authentication$$

$$Impact = 10.41 * (1 - (1 - ConfImpact) * (1 - IntegImpact) * (1 - AvailImpact))$$

$$f(Impact) = \begin{cases} 0 & \text{if } Impact = 0 \\ 1.176 & \text{otherwise} \end{cases}$$

$$BaseScore = roundTo1\ Decimal\ (((0.6 * Impact) + (0.4 * Exploitability) - 1.5) * f(Impact))$$

# Rappresentazione sintetica risposte

(Ecco a che cosa servono le abbreviazioni|)

Le risposte alle domande sono presentate in modo sintetico e non ambiguo tramite una stringa di testo detta **vector string**.

Formato generale: coppie di abbreviazioni *metrica:risposta* separate da un carattere speciale (/).

MetrAbbr1:RispAbbr1/MetrAbbr2:RispAbbr2/...

Esempio:

AV:N/AC:L/Au:N/C:P/I:P/A:C

# Se l'impressione che avete è questa...

(...avete pienamente ragione!)



# E gli altri punteggi?

(“Temporal” e “Environmental”?)

Lo standard CVSS non richiede il calcolo degli altri due punteggi (Temporal, Environmental).

In generale:

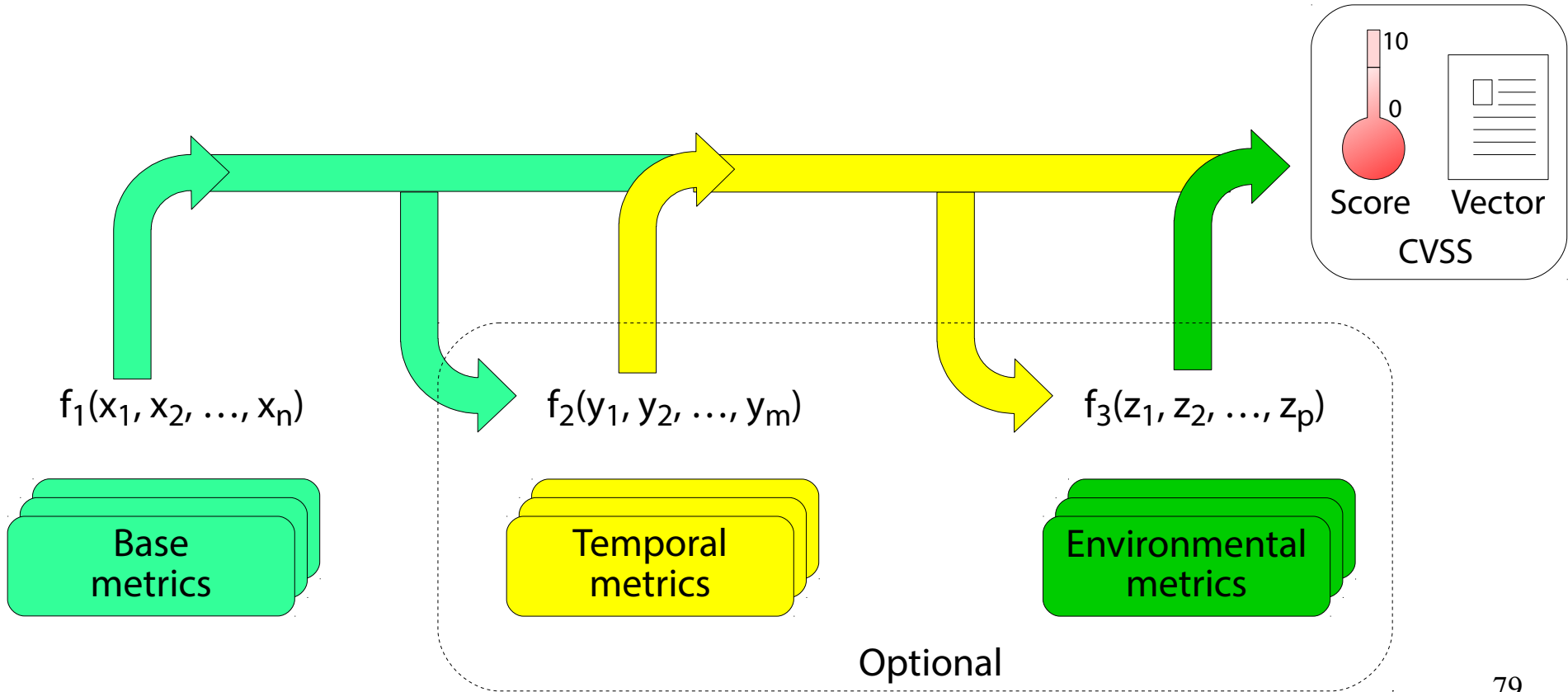
i punteggi “Temporal” ed “Environmental” si calcolano nello stesso modo del punteggio “Base” (cambiano i questionari);

il punteggio “Temporal” “ingloba” il punteggio Base;

Il punteggio “Environmental” “ingloba” il punteggio Temporal.

# Relazioni tra i diversi punteggi

(Base → Temporal → Environmental)



# Temporal: Exploitability (E)

(Unproven, proof of concept, functional, high, not defined)

Qual è lo stato attuale delle tecniche di sfruttamento della vulnerabilità?

Temporal  
Metric Group

Exploitability

Remediation Level

Report  
Confidence

Valore	Descrizione	Punt.
Unproven (U)	L'exploit non è pubblico, oppure esiste in linea solo teorica.	0.85
Proof of Concept (P)	È disponibile una bozza dimostrativa ( <b>Proof of Concept, PoC</b> ). Richiede adattamenti non banali per funzionare.	0.9
Functional (F)	È disponibile un exploit funzionante nella maggioranza dei casi in cui la vulnerabilità è presente.	0.95
High (H)	La vulnerabilità può essere sfruttata in modo automatico (anche da worm e virus).	1.0
Not Defined (ND)	Si ignori tale punteggio.	1.0



# Temporal: Remediation Level (RL)

(Official fix, temporary fix, workaround, unavailable, not defined)

È presente un rimedio per mitigare la vulnerabilità?

## Temporal Metric Group

Exploitability

Remediation Level

Report  
Confidence

Valore	Descrizione	Punt.
Official fix (O)	Il vendor mette a disposizione un rimedio ufficiale (patch, aggiornamento software).	0.87
Temporary fix (T)	Il vendor mette a disposizione un rimedio ufficiale, ma temporaneo.	0.90
Workaround (W)	Una terza parte (NON il vendor) mette a disposizione un rimedio non ufficiale.	0.95
Unavailable (U)	Non è disponibile un rimedio, o è impossibile applicare una soluzione suggerita.	1.0
Not Defined (ND)	Si ignori tale punteggio.	1.0

# Temporal: Report Confidence (RC)

(Unconfirmed, uncorroborated, confirmed, not defined)

La vulnerabilità esiste veramente? È descritta in maniera credibile?

Temporal  
Metric Group

Exploitability

Remediation Level

Report  
Confidence

Valore	Descrizione	Punt.
Unconfirmed (UC)	La vulnerabilità è divulgata da una singola fonte non confermata, o da più fonti in mutuo conflitto.	0.9
Uncorroborated (UR)	La vulnerabilità è divulgata da più fonti concordi. Può esistere un livello residuo di incertezza.	0.95
Confirmed (C)	La vulnerabilità è confermata dal vendor.	1.0
Not Defined (ND)	Si ignori tale punteggio.	1.0

# Calcolo del punteggio "Temporal"

(Tramite un insieme perverso di formule)

Il **Punteggio Temporal (Temporal Score)** stima la gravità di una vulnerabilità, includendo il fattore tempo.

*TemporalScore = roundTo1 Decimal ( BaseScore \* Exploitab \* RemedLvl \* ReportConf )*

# Se l'impressione che avete è questa...

(...avete pienamente ragione!)



# Environmental: Collateral Damage Potential (CDP)

(None, low, low-medium, medium-high, high, not defined)

Qual è l'impatto potenziale della vulnerabilità sui sistemi fisici, sulle persone e sulle risorse finanziarie?

Environmental  
Metric Group

Collateral Damage  
Potential

Target  
Distribution

Valore	Descrizione	Punt.
None (N)	Nessun impatto.	0
Low (L)	Danno fisico basso, perdita marginale di guadagno.	0.1
Low-Medium (LM)	Danno fisico ed economico moderato.	0.3
Medium-High (MH)	Danno fisico ed economico significativo.	0.4
High (H)	Danno fisico ed economico catastrofico.	0.5
Not Defined (ND)	Si ignori tale punteggio.	1.0

# Environmental: Target Distribution (TD)

(None, low, low-medium, medium-high, high, not defined)

Quale percentuale di asset nell'infrastruttura è soggetta alla vulnerabilità?

Environmental  
Metric Group

Collateral Damage  
Potential

Target  
Distribution

Valore	Descrizione	Punt.
None (N)	Percentuale nulla.	0
Low (L)	1%-25% degli asset.	0.25
Medium (M)	26%-75% degli asset.	0.75
High (H)	76%-100% degli asset.	1.0
Not Defined (ND)	Si ignori tale punteggio.	1.0

# Environmental: Confidentiality Req. (CR)

(Low, medium, high, not defined)

Qual è l'impatto di una perdita di confidenzialità?

## Environmental Metric Group

Confidentiality  
Requirement

Integrity  
Requirement

Availability  
Requirement

Valore	Descrizione	Punt.
Low (L)	L'impatto è lieve.	0.5
Medium (M)	L'impatto è serio.	1.0
High (H)	L'impatto è catastrofico.	1.51
Not Defined (ND)	Si ignori tale punteggio.	1.0

# Environmental: Integrity Req. (IR)

(Low, medium, high, not defined)

Qual è l'impatto di una perdita di integrità?

## Environmental Metric Group

Confidentiality  
Requirement

Integrity  
Requirement

Availability  
Requirement

Valore	Descrizione	Punt.
Low (L)	L'impatto è lieve.	0.5
Medium (M)	L'impatto è serio.	1.0
High (H)	L'impatto è catastrofico.	1.51
Not Defined (ND)	Si ignori tale punteggio.	1.0



# Environmental: Availability Req. (AR)

(Low, medium, high, not defined)

Qual è l'impatto di una perdita di disponibilità?

## Environmental Metric Group

Confidentiality  
Requirement

Integrity  
Requirement

Availability  
Requirement

Valore	Descrizione	Punt.
Low (L)	L'impatto è lieve.	0.5
Medium (M)	L'impatto è serio.	1.0
High (H)	L'impatto è catastrofico.	1.51
Not Defined (ND)	Si ignori tale punteggio.	1.0

# Calcolo del punteggio “Environmental”

(Tramite un insieme perverso di formule)

**Il Punteggio Environmental (Environmental Score)** stima la gravità di una vulnerabilità, includendo il fattore tempo.

$$AdjImp = \min(10, 10.41 * (1 - (1 - ConfImp * ConfReq) * (1 - IntImp * IntReq) * (1 - AvImp * AvReq)))$$

*AdjTemp = punteggio Temporal ricalcolato con AdjImp al posto di Impact*

$$EnvironmentalScore = \text{roundTo 1 Decimal}((AdjTemp + (10 - AdjTemp) * CollatDamPot) * TargetDist)$$

# Se l'impressione che avete è questa...

(...avete pienamente ragione!)



# Il vettore CVSS

(Rappresenta sinteticamente le risposte al questionario)

Il **vettore CVSS (CVSS vector)** è una stringa che riassume sinteticamente le risposte alle domande del questionario.

Formato del vettore CVSS: coppie *abbr\_domanda:abbr\_risposta*, separate dal carattere /.

Ad es. (Base): AV:L/AC:H/Au:N/C:N/I:P/A:C

# Chi calcola il punteggio CVSS?

(Dipende dalla tipologia di punteggio)

**Base, Temporal Score:** vendor hardware e software (conoscono molto bene i dettagli di funzionamento dei loro prodotti e le dinamiche della vulnerabilità).

**Environmental Score:** amministratori ed utenti di infrastrutture informatiche (conoscono molto bene l'ambiente in cui è installato il prodotto vulnerabile).

# Chi usa il punteggio CVSS?

(Chiunque abbia a che fare con il processo di sicurezza)

Il punteggio CVSS è usato da chiunque abbia a che fare con il processo di gestione della sicurezza.

Enti di sicurezza (pubblicazione di bollettini).

Vendor hardware e software.

Ricercatori (accademici, liberi professionisti).

Responsibili di sicurezza in azienda.

Pubblica Amministrazione.

Sviluppatori di software.

# Il foglio di calcolo CVSS v2

(Permette di calcolare il punteggio CVSS v2 con pochi click)

All'URL seguente:

<https://nvd.nist.gov/cvss.cfm?calculator&adv&version=2>

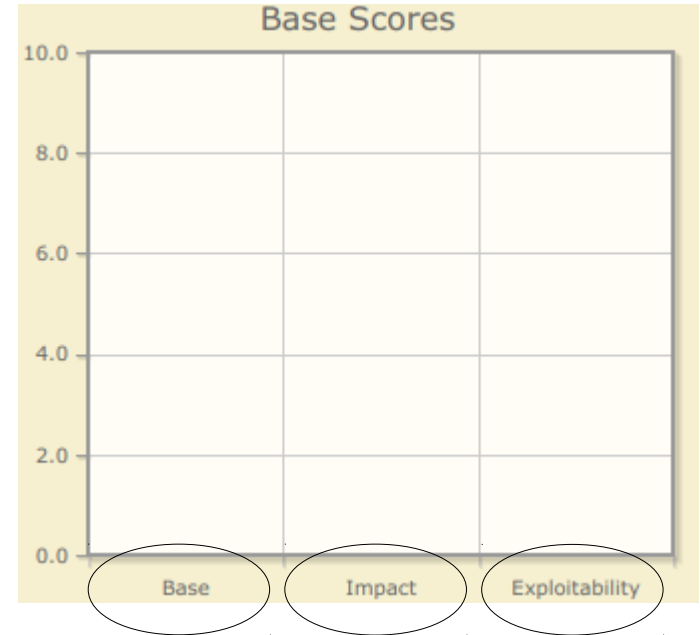
è presente un foglio di calcolo Web per la stima dei punteggi CVSS v2 (Base, Temporal ed Environmental).

# Visualizzazione punteggi

(Punteggio Base)

La parte iniziale della pagina Web mostra i diagrammi a barre dei punteggi (e relative metriche).

Il primo diagramma è relativo al punteggio Base.



Punteggio Base

Metriche usate per calcolare Base

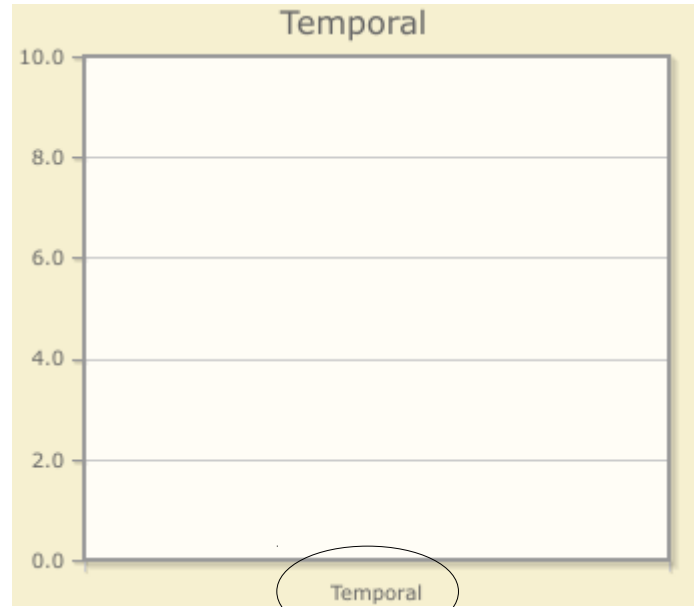


# Visualizzazione punteggi

(Punteggio Temporal)

La parte iniziale della pagina Web mostra i diagrammi a barre dei punteggi (e relative metriche).

Il secondo diagramma è relativo al punteggio Temporal.



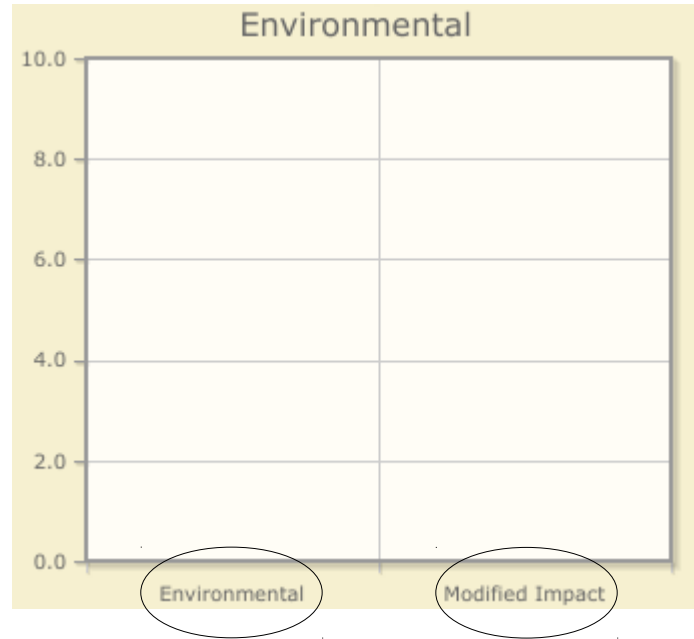
Punteggio  
Temporal

# Visualizzazione punteggi

(Punteggio Environmental)

La parte iniziale della pagina Web mostra i diagrammi a barre dei punteggi (e relative metriche).

Il terzo diagramma è relativo al punteggio Environmental.



Punteggio  
Environmental

Metrica usata per  
calcolare  
Environmental 98

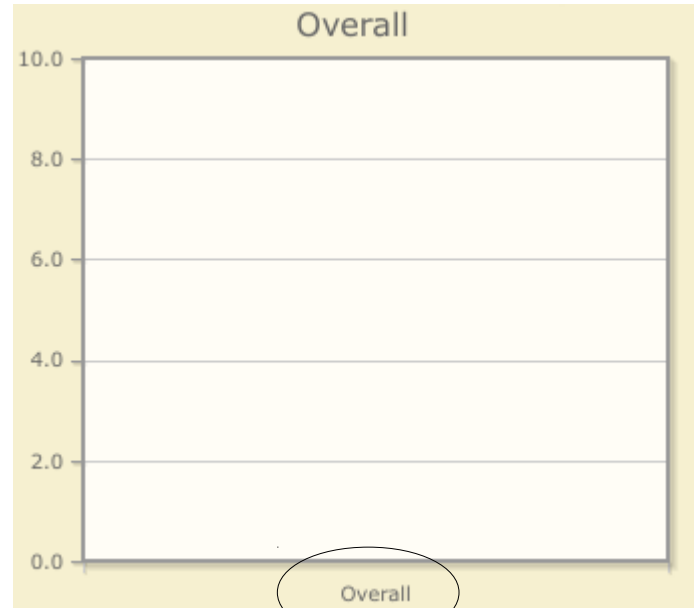
# Visualizzazione punteggi

(Punteggio Overall)

La parte iniziale della pagina Web mostra i diagrammi a barre dei punteggi (e relative metriche).

L'ultimo diagramma riporta il punteggio finale (Overall).

**Punteggio Overall:** coincide con quel punteggio che si è scelto di calcolare.



Base, Temporal  
o Environmental

# Visualizzazione punteggi

(Scheda riassuntiva)

Sotto il grafico Overall è presente una scheda riassuntiva contenente il tipo ed il valore del punteggio calcolato.

Il link Show Equations mostra le equazioni usate nel calcolo.

<b>CVSS Base Score</b>	<b>0</b>
Impact Subscore	0
Exploitability Subscore	0
<b>CVSS Temporal Score</b>	<b>0</b>
<b>CVSS Environmental Score</b>	<b>0</b>
Modified Impact Subscore	0
<b>Overall CVSS Score</b>	<b>0</b>
<a href="#"><u>Show Equations</u></a>	

# Immissione metriche

(Punteggio Base)

Sotto la scheda riassuntiva sono presenti tre tab (uno per tipologia di punteggio).

Il primo tab illustra le scelte possibili per le metriche del punteggio Base.

▼ Base Score Metrics

<b>Exploitability Metrics</b>	<b>Impact Metrics</b>
Access Vector (AV)*	Confidentiality Impact (C)*
<input type="button" value="Local (AV:L)"/> <input type="button" value="Adjacent Network (AV:A)"/> <input type="button" value="Network (AV:N)"/>	<input type="button" value="None (C:N)"/> <input type="button" value="Partial (C:P)"/> <input type="button" value="Complete (C:C)"/>
Access Complexity (AC)*	Integrity Impact (I)*
<input type="button" value="High (AC:H)"/> <input type="button" value="Medium (AC:M)"/> <input type="button" value="Low (AC:L)"/>	<input type="button" value="None (I:N)"/> <input type="button" value="Partial (I:P)"/> <input type="button" value="Complete (I:C)"/>
Authentication (Au)*	Availability Impact (A)*
<input type="button" value="Multiple (Au:M)"/> <input type="button" value="Single (Au:S)"/> <input type="button" value="None (Au:N)"/>	<input type="button" value="None (A:N)"/> <input type="button" value="Partial (A:P)"/> <input type="button" value="Complete (A:C)"/>

\* - All base metrics are required to generate a base score.

# Immissione metriche

(Punteggio Temporal)

Sotto la scheda riassuntiva sono presenti tre tab (uno per tipologia di punteggio).

Il secondo tab illustra le scelte possibili per le metriche del punteggio Temporal.

▼ Temporal Score Metrics

Exploitability (E)

Not Defined (E:ND)	Unproven that exploit exists (E:U)	Proof of concept code (E:POC)	Functional exploit exists (E:F)	High (E:H)
--------------------	------------------------------------	-------------------------------	---------------------------------	------------

Remediation Level (RL)

Not Defined (RL:ND)	Official fix (RL:OF)	Temporary fix (RL:TF)	Workaround (RL:W)	Unavailable (RL:U)
---------------------	----------------------	-----------------------	-------------------	--------------------

Report Confidence (RC)

Not Defined (RC:ND)	Unconfirmed (RC:UC)	Uncorroborated (RC:UR)	Confirmed (RC:C)
---------------------	---------------------	------------------------	------------------

# Immissione metriche

(Punteggio Environmental)

Sotto la scheda riassuntiva sono presenti tre tab (uno per tipologia di punteggio).

Il terzo tab illustra le scelte possibili per le metriche del punteggio Environmental.

▼ Environmental Score Metrics

<b>General Modifiers</b>	<b>Impact Subscore Modifiers</b>
Collateral Damage Potential (CDP)	Confidentiality Requirement (CR)
<input type="button" value="Not Defined (CDP:ND)"/> <input type="button" value="None (CDP:N)"/> <input type="button" value="Low (light loss) (CDP:L)"/> <input type="button" value="Low-Medium (CDP:LM)"/> <input type="button" value="Medium-High (CDP:MH)"/> <input type="button" value="High (catastrophic loss) (CDP:H)"/>	<input type="button" value="Not Defined (CR:ND)"/> <input type="button" value="Low (CR:L)"/> <input type="button" value="Medium (CR:M)"/> <input type="button" value="High (CR:H)"/>
Target Distribution (TD)	Integrity Requirement (IR)
<input type="button" value="Not Defined (TD:ND)"/> <input type="button" value="None [0%] (TD:N)"/> <input type="button" value="Low [0-25%] (TD:L)"/> <input type="button" value="Medium [26-75%] (TD:M)"/> <input type="button" value="High [76-100%] (TD:H)"/>	<input type="button" value="Not Defined (IR:ND)"/> <input type="button" value="Low (IR:L)"/> <input type="button" value="Medium (IR:M)"/> <input type="button" value="High (IR:H)"/>
	Availability Requirement (AR)
	<input type="button" value="Not Defined (AR:ND)"/> <input type="button" value="Low (AR:L)"/> <input type="button" value="Medium (AR:M)"/> <input type="button" value="High (AR:H)"/>

# Gestione foglio di calcolo

(Update Scores; Clear Form)

Infine, gli ultimi due bottoni in fondo alla pagina permettono di:

aggiornare i grafici e la scheda di punteggio (**Update Scores**);

reimpostare i grafici e la scheda di punteggio (**Clear Form**).





# Un esempio concreto

(Vale spesso più di 0x3e8 parole)

La A.C.M.E. Srl\* è una azienda che produce beni e servizi di ogni tipo per il pubblico.

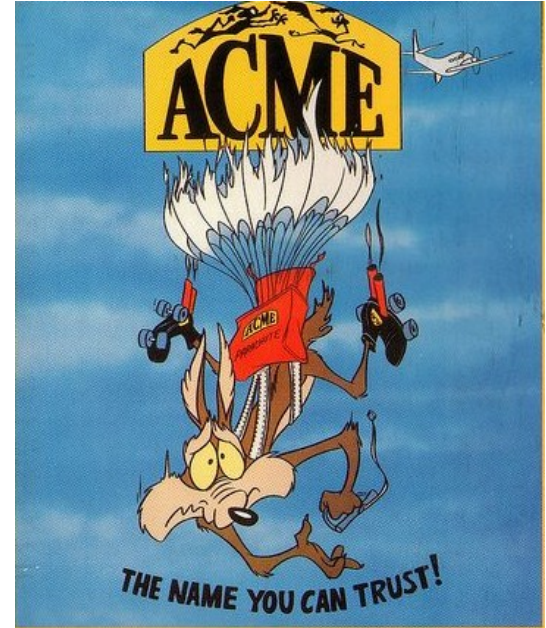
A.C.M.E. espone al pubblico un server Web contenente:

- un catalogo dei prodotti;

- un negozio elettronico.

Il Web server è vulnerabile a CVE-2014-6271.

\* A.C.M.E. Srl è un nome di fantasia e non ha alcuna attinenza con fatti e/o personaggi reali.



# Calcolo del punteggio CVSS

(Environmental, CVSS v2, CVE-2014-6271)

Si vuole calcolare il punteggio CVSS con riferimento a tale vulnerabilità.

Tipologia di punteggio: Environmental.

Versione CVSS: v2.

Vulnerabilità: CVE-2014-6271.

# Punteggio Base: Exploitability Metrics

(Access Vector, Access Complexity, Authentication)

Il Web server è accessibile pubblicamente tramite Internet (TCP/IP/v4).

→ AV:N.

Il Web server è vulnerabile nella sua configurazione di default.

→ AC:L.

Lo sfruttamento non richiede autenticazione.

→ Au:N.

Exploitability Metrics		
Access Vector (AV)*		
Local (AV:L)	Adjacent Network (AV:A)	Network (AV:N)
Access Complexity (AC)*		
High (AC:H)	Medium (AC:M)	Low (AC:L)
Authentication (Au)*		
Multiple (Au:M)	Single (Au:S)	None (Au:N)

# Punteggio Base: Impact Metrics

(Confidentiality Impact, Integrity Impact, Availability Impact)

Il Web server esegue con un utente (**www-data**) avente privilegi ridotti.

→ Non tutti i file del Web server sono accessibili (solo quelli accessibili a **www-data**).

Il database back-end associato al Web server non memorizza tutte le informazioni aziendali.

→ Violando il Web server, non si riesce a raggiungerle.

Impact Metrics		
Confidentiality Impact (C)*		
None (C:N)	Partial (C:P)	Complete (C:C)
Integrity Impact (I)*		
None (I:N)	Partial (I:P)	Complete (I:C)
Availability Impact (A)*		
None (A:N)	Partial (A:P)	Complete (A:C)

# Punteggio Base: Impact Metrics

(Confidentiality Impact, Integrity Impact, Availability Impact)

Per tali motivi, l'impatto sulle proprietà di confidenzialità, integrità e disponibilità è parziale.

→ C:P.

→ I:P.

→ A:P.

## Impact Metrics

Confidentiality Impact (C)\*

None (C:N)

Partial (C:P)

Complete (C:C)

Integrity Impact (I)\*

None (I:N)

Partial (I:P)

Complete (I:C)

Availability Impact (A)\*

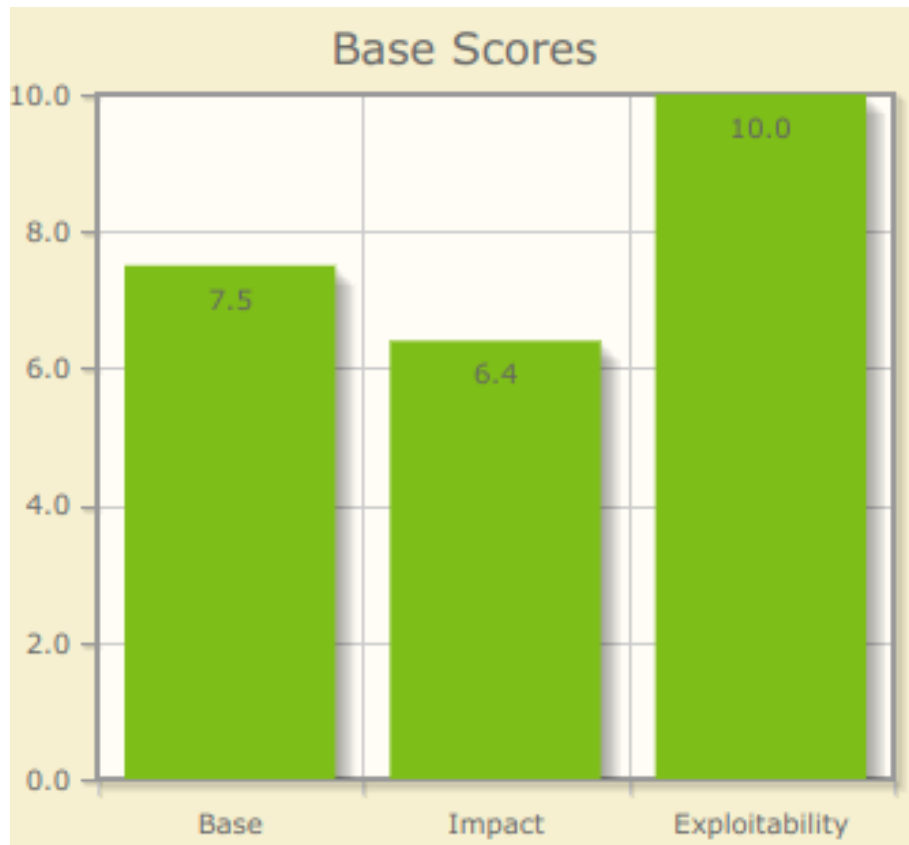
None (A:N)

Partial (A:P)

Complete (A:C)

# Il punteggio Base

(In numeri e diagrammi)



<b>CVSS Base Score</b>	<b>7.5</b>
Impact Subscore	6.4
Exploitability Subscore	10
<b>CVSS Temporal Score</b>	<b>NaN</b>
<b>CVSS Environmental Score</b>	<b>NaN</b>
Modified Impact Subscore	NaN
<b>Overall CVSS Score</b>	<b>NaN</b>

Show Equations

# Punteggio Temporal: Temporal Score Metrics

(Exploitability, Remediation level, Report Confidence)

Esistono script pronti (a prova di utonto) per l'esecuzione di tale exploit.

→ E:H.

La GNU Free Software Foundation ha rilasciato una nuova versione della shell BASH che elimina il difetto.

→ RL:OF.

La vulnerabilità è vera.

→ RC:C.

## Exploitability (E)

Not Defined (E:ND)	Unproven that exploit exists (E:U)	Proof of concept code (E:POC)	Functional exploit exists (E:F)	High (E:H)
--------------------	------------------------------------	-------------------------------	---------------------------------	------------

## Remediation Level (RL)

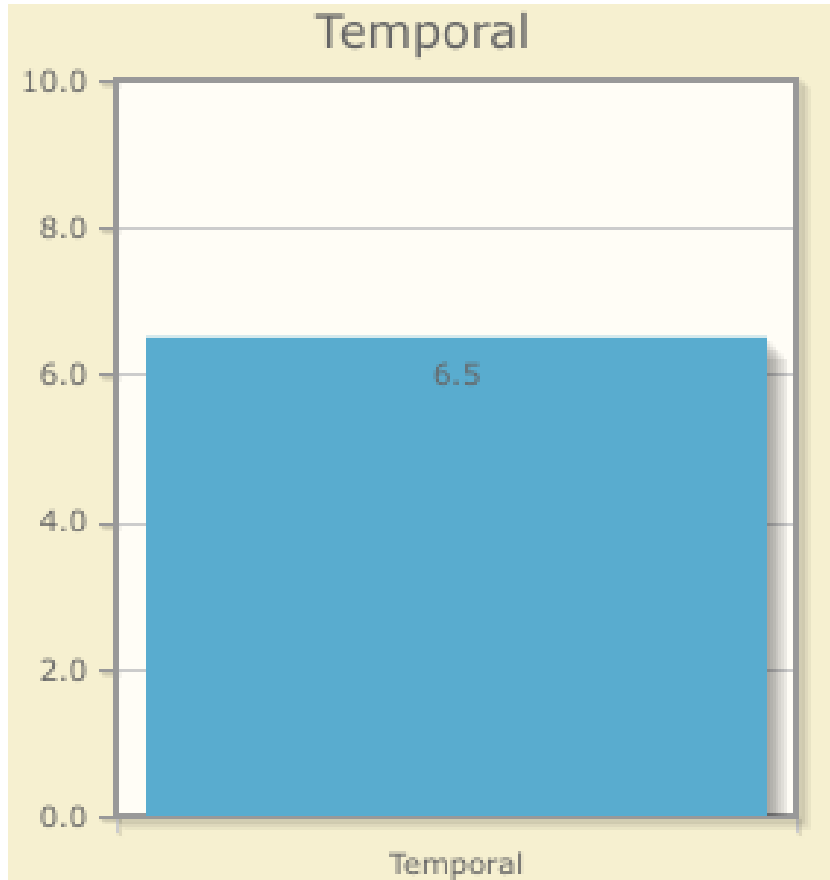
Not Defined (RL:ND)	Official fix (RL:OF)	Temporary fix (RL:TF)	Workaround (RL:W)	Unavailable (RL:U)
---------------------	----------------------	-----------------------	-------------------	--------------------

## Report Confidence (RC)

Not Defined (RC:ND)	Unconfirmed (RC:UC)	Uncorroborated (RC:UR)	Confirmed (RC:C)
---------------------	---------------------	------------------------	------------------

# Il punteggio Temporal

(In numeri e diagrammi)



<b>CVSS Base Score</b>	<b>7.5</b>
Impact Subscore	6.4
Exploitability Subscore	10
<b>CVSS Temporal Score</b>	<b>6.5</b>
<b>CVSS Environmental Score</b>	<b>NaN</b>
Modified Impact Subscore	NaN
<b>Overall CVSS Score</b>	<b>NaN</b>

[Show Equations](#)



# Punteggio Environmental: General Modifiers

(Collateral Damage Potential, Target Distribution)

Il danno massimo stimato sul Web server è un Denial of Service dovuto all'esecuzione di una fork bomb.

Il negozio elettronico opera molto più lentamente.

→ CDP:MH.

Il Web server è l'unico asset soggetto a CVE-2014-6271.

→ TD:L.

## General Modifiers

### Collateral Damage Potential (CDP)

Not Defined (CDP:ND)

None (CDP:N)

Low (light loss) (CDP:L)

Low-Medium (CDP:LM)

**Medium-High (CDP:MH)**

High (catastrophic loss) (CDP:H)

### Target Distribution (TD)

Not Defined (TD:ND)

None [0%] (TD:N)

**Low [0-25%] (TD:L)**

Medium [26-75%] (TD:M)

High [76-100%] (TD:H)

# Punteggio Environmental: Impact Subscore Modifiers

(Confidentiality Requirement, Integrity Requirement, Availability Requirement)

Il Web server esegue con un utente non privilegiato.

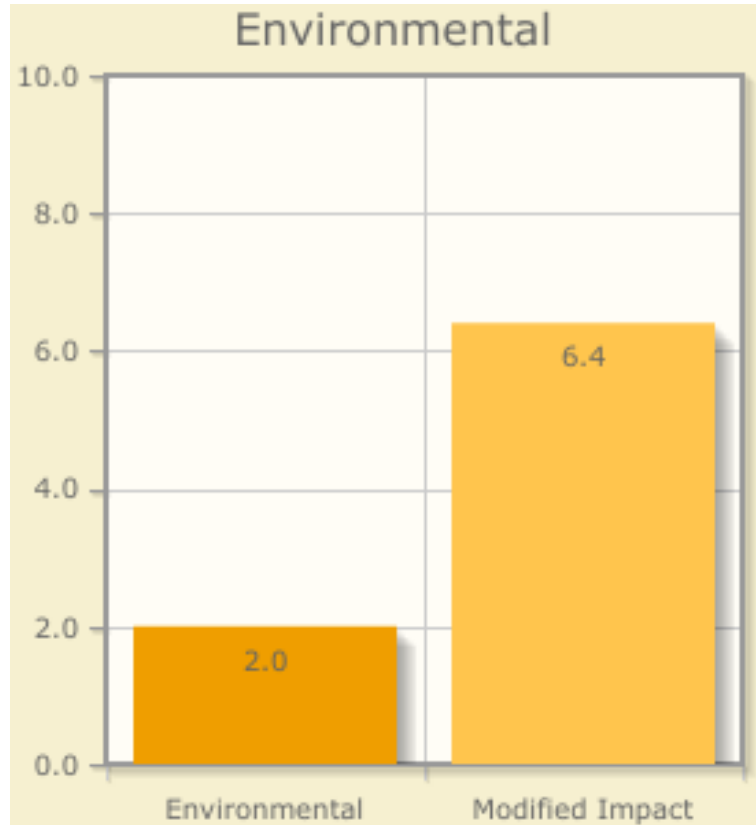
Il database back-end non ospita tutte le informazioni aziendali.

- CR:M.
- IR:M.
- AR:M.

Impact Subscore Modifiers			
Confidentiality Requirement (CR)			
Not Defined (CR:ND)	Low (CR:L)	<b>Medium (CR:M)</b>	High (CR:H)
Integrity Requirement (IR)			
Not Defined (IR:ND)	Low (IR:L)	<b>Medium (IR:M)</b>	High (IR:H)
Availability Requirement (AR)			
Not Defined (AR:ND)	Low (AR:L)	<b>Medium (AR:M)</b>	High (AR:H)

# Il punteggio Environmental

(In numeri e diagrammi)



<b>CVSS Base Score</b>	<b>7.5</b>
Impact Subscore	6.4
Exploitability Subscore	10
<b>CVSS Temporal Score</b>	<b>6.5</b>
<b>CVSS Environmental Score</b>	<b>2</b>
Modified Impact Subscore	6.4
<b>Overall CVSS Score</b>	<b>2</b>

[Show Equations](#)

# Qualche osservazione

(Doverosa)

Le risposte ai questionari sono soggettive.

Tecnici con diverse sensibilità hanno una propensione diversa alle risposte.

In questo caso specifico, la componente temporale fa abbassare il punteggio di base.

Esiste un fix ufficiale, distribuito dal vendor.

Inoltre, la componente ambientale fa abbassare il punteggio ambientale (e di parecchio pure).

La vulnerabilità è applicabile a meno dell'1% dei sistemi.

# Altri approcci di catalogazione

(Molto importanti, in quanto adottati ovunque)

I sistemi CVE e CVSS, pur con tutti i loro limiti, rappresentano un buon passo verso la creazione di un catalogo uniforme di vulnerabilità.

Un approccio simile è stato portato avanti anche nei seguenti ambiti correlati:

- enumerazione degli asset hardware/software;

- procedure di verifica delle vulnerabilità;

- enumerazione e valutazione delle debolezze software.

# Common Platform Enumeration

(Un catalogo uniforme di piattaforme hardware/software)

Il **Common Enumeration (CPE)** è uno schema di denominazione di asset hardware e software.

Tutte le piattaforme informatiche:

- riferite nei bollettini di sicurezza;

- riferite nell'output di strumenti di scansione e di monitoraggio;

sono descritte in tale formato.

Home page:

<https://nvd.nist.gov/cpe.cfm>

# Perché serve?

(Otherwise it's Tower of Babel again and again and again...)

Team diversi possono nominare lo stesso asset in modi radicalmente diversi, ma equivalenti. Ad esempio:

Windows NT 5.0.2195

Windows 2000

sono lo stesso prodotto!

→ Con tanti asset e nomi equivalenti si rischia l'effetto "Torre di Babele" visto in precedenza.

# Componenti del sistema CPE

(Naming, matching, dictionary, attributed language)

Il CPE consta di quattro componenti chiave.

**Naming:** un meccanismo di assegnazione dei nomi agli asset hardware/software.

**Matching:** un meccanismo per il confronto di un CPE sorgente con un altro CPE destinazione.

**Dictionary:** un dizionario di schede relative ad asset hw/sw.

**Language:** un linguaggio per la descrizione di infrastrutture complesse.



# Well Formed CPE Name

(Naming)

Il **Well Formed CPE Name (WFN)** è un insieme non ordinato di coppie (attributo, valore).

$wfn: [ a_1=v_1, a_2=v_2, \dots, a_n=v_n ]$

$a_j$ : una stringa identificante un attributo. Gli attributi sono determinati nel documento di specifica di CPE (non si possono impostare a caso).

$v_j$ : una stringa alfanumerica. La comunità decide il nome di un asset una volta per tutte.

# Attributi

(Di base ed aggiuntivi)

- part
- vendor
- product
- version
- update
- edition
- language

Attributi di base  
(sempre presenti)

- sw\_edition
- target\_sw
- target\_hw
- other

Attributi aggiuntivi  
(non sempre presenti)

# Un esempio di WFN

(Tanto per chiarire le idee)

```
wfn:[part="a",vendor="microsoft",product="internet_explorer",  
version="8\0.6001",update="beta"]
```

## **NOTA BENE:**

part="a" → applicazione

part="o" → sistema operativo

part="h" → apparato hardware

# Rappresentazioni tramite stringa

(Stampabili)

Il WFN è una rappresentazione logica. Non è pensato per essere stampato nell'output dei programmi.

A tal scopo, lo standard CPE prevede due rappresentazioni concrete (basate su stringhe).

**URI:** valori degli attributi di base concatenati tramite :, attributo "part" preposto dal carattere /.

**FS:** valori degli attributi di base ed aggiuntivi, sempre concatenati tramite :.

# Alcuni esempi di binding URI/FS di WFN

(Sempre per chiarire le idee)

(Applicazione) Microsoft Office 2007 Professional Service Pack 2.

URI: cpe:/a:microsoft:office:2007:sp2:professional

FS: cpe:2.3:a:microsoft:office:2007:sp2:-:\*:professional:\*:\*.\*

(Operating System) Microsoft Windows 7 64-bit Service Pack 1

URI: cpe:/o:microsoft:windows\_7:-:sp1:x64

FS: cpe:2.3:o:microsoft:windows\_7:-:sp1:-:\*:\*:\*:x64:\*

(Hardware) 3Com Router 3012

URI: cpe:/h:3com:3c13612

FS: cpe:2.3:h:3com:3c13612:-:\*:\*:\*:\*:\*.\*

# Binding ed unbinding del WFN

(Rappresentazione logica ↔ Rappresentazione concreta, machine-readable)

**Binding:** operazione di trasformazione di un WFN in una rappresentazione basata su stringa (URI, FS).

**Unbinding:** operazione di trasformazione di una rappresentazione basata su stringa (URI, FS) in un WFN.

# Confronto tra CPE

(Matching)

Il confronto tra due CPE viene effettuato a partire dai WFN (matching “agnostico” rispetto al binding).

Due funzioni previste dallo standard.

**Compare\_WFNs (source, target)**

Confronta uno per uno valori sorgenti e destinazione.  
Ritorna una matrice di risultati.

**CPE\_x (source, target)**

**x** è **EQUAL**, **DISJOINT**, **SUBSET**, **SUPERSET**.

Confronta due WPN e ritorna **TRUE** se vale **x**.

# Dizionario dei CPE

(Dictionary)

Lo standard prevede la gestione di un dizionario di tutti i CPE noti.

Attualmente il dizionario è in formato XML.

```
<cpe-item name="cpe:/o:canonical:ubuntu_linux:16.10">
  <title xml:lang="en-US">Canonical Ubuntu Linux 16.10</title>
  <references>
    <reference href="http://people.canonical.com/~ubuntu-
security/cve/2016/CVE-2016-1576.html">Advisory</reference>
    <reference href="http://www.ubuntu.com/">Vendor</reference>
  </references>
</cpe-item>
```



# Applicability Language

(Linguaggio)

L'**applicability language** è un linguaggio per la rappresentazione di un sistema complesso a partire da più CPE.

**Applicability statement:** documento XML contenente la rappresentazione.

# Un esempio di applicability statement

(Microsoft Windows XP con Internet Explorer 7.x o 8.x)

```
<cpe:platform id="789">
  <cpe:title>
    Microsoft Windows XP with Internet Explorer 7.x or 8
  </cpe:title>
  <cpe:logical-test operator="AND" negate="FALSE">
    <cpe:fact-ref
      name="cpe:2.3:o:microsoft:windows_xp:*:*:*:*:*:*:*" />
    <cpe:logical-test operator="OR" negate="FALSE">
      name="cpe:2.3:a:microsoft:internet_explorer:7.*:*:*:*:*:*:*" />
    <cpe:fact-ref
      name="cpe:2.3:a:microsoft:internet_explorer:8.*:*:*:*:*:*:*" />
    </cpe:logical-test>
  </cpe:logical-test>
</cpe:platform>
```

# Open Vulnerability Assessment Language

(Standardizza la procedura di stima delle vulnerabilità di un asset)

## L'Open Vulnerability Assessment Language

(**OVAL**) è uno standard per:

promuovere la diffusione pubblica ed aperta di informazioni collegate al processo di sicurezza.

uniformare il trasferimento di tali informazioni tra le diverse applicazioni.

<http://www.itsecdb.com/oval/>

A cosa serve?

Ad automatizzare la gestione di configurazioni, patch, controllo delle vulnerabilità.

# Componenti di OVAL

(Un linguaggio basato su XML, un archivio di contenuti, alcune linee guida)

OVAL consta di tre componenti distinti.

Un linguaggio basato su XML per l'espressione di uno "stato interno" di un asset.

Un archivio di contenuti (in linguaggio OVAL).

Un insieme di linee guida la corretta implementazione dello standard.

[http://oval.mitre.org/about/images/how\\_oval\\_works.pdf](http://oval.mitre.org/about/images/how_oval_works.pdf)

# Il linguaggio OVAL

(Standardizza la procedura di stima delle vulnerabilità di un asset)

Il linguaggio XML di OVAL permette di descrivere in maniera uniforme le attività di stima delle vulnerabilità di un asset.

Rappresentazione della configurazione.

Analisi del sistema e verifica della presenza di uno specifico "stato interno" (vulnerable, patched, ...).

Presentazione dei risultati della verifica.

# Definizione OVAL

(La specifica descrizione XML di una attività di gestione della sicurezza)

La **definizione OVAL** è la singola descrizione XML di una specifica attività di gestione della sicurezza. Ogni definizione ha un proprio identificatore OVAL nel formato seguente:

***oval : oval\_source : def : id***

**oval\_source:** una sorgente di bollettini di sicurezza.

**id:** un numero intero crescente.

# L'archivio OVAL

(Memorizza i possibili aspetti del processo di gestione della sicurezza)

L'archivio OVAL è un elenco di oggetti descritti in XML e rappresentanti i diversi aspetti del processo di gestione della sicurezza.

**Compliance:** descrive una buona prassi di sicurezza.

**Inventory:** descrive l'installazione di un asset software.

**Patch:** descrive le condizioni per l'installazione di una patch.

**Vulnerability:** descrive una vulnerabilità.

# Chi usa OVAL?

(Un po' tutti, più o meno inconsciamente)

OVAL è usato:

da strumenti di monitoraggio avanzato della sicurezza di una infrastruttura (per l'individuazione di asset non in linea con la politica di sicurezza aziendale);

da diversi motori di ricerca di CVE (per una spiegazione

non ambigua di come si verifica la presenza di una vulnerabilità).



# OVAL in azione

(Una definizione per verificare l'installabilità di una patch per CVE-2014-6271)

## The patch should be installed

- IF : All of the following are true

### Prerequisites (Extended Definitions)

Debian 7 is installed [oval:org.mitre.oval:def:19338](#)

- IF : bash DPKG is earlier than 0:4.2+dfsg-0.1+deb7u1

- [Linux : Debian DPKG Test](#) : bash DPKG is earlier than 0:4.2+dfsg-0.1+deb7u1

At least one of the objects listed below must exist on the system (Existence check)

[Linux : Debian DPKG Package](#) bash package information

**bash**

**RPM Version** less than **0:4.2+dfsg-0.1+deb7u1** (datatype=evr\_string)

version is earlier than 0:4.2+dfsg-0.1+deb7u1 [linux : dpkginfo\\_state](#)

- IF : Any one of the following are true GNU/Linux or GNU/kFreeBSD kernel

### Prerequisites (Extended Definitions)

Debian GNU/kFreeBSD is installed [oval:org.mitre.oval:def:24698](#)

Debian GNU/Linux is installed [oval:org.mitre.oval:def:24894](#)

# Una osservazione

(Doverosa)

Tutti gli strumenti visti finora mirano al catalogo ed alla gestione automatica e non ambigua di vulnerabilità software.

Che siano nel codice sorgente, nella configurazione, ...

**Ottica:** catalogare gli errori dopo averli fatti.

# Domanda

(Spontanea)

Esiste un analogo catalogo specifico di debolezze software?

A tutti i livelli: progetto, implementazione, configurazione.

**Ottica:** catalogare gli errori fattibili.

# Common Weaknesses and Exposures

(Un catalogo uniforme di debolezze software)

Il sistema delle **Common Weaknesses and Exposures (CWE)** cataloga in modo uniforme le debolezze software (indipendentemente dalla loro sfruttabilità).

Home page: <https://cwe.mitre.org>

# Obiettivi

(Nobili)

Definire un linguaggio comune per la descrizione di debolezze nell'architettura, nel progetto, nel codice.

Definire un metro standard di misura per gli strumenti di verifica automatica del software.

Aiutare nella prevenzione delle debolezze.

# Organizzazione del catalogo

(Gerarchica e relazionale)

Il catalogo è un insieme di oggetti. Ogni oggetto ha un suo identificatore ed attributi.

Identificatore: un numero intero crescente.

Formato attributi: largamente condiviso tra oggetti.

Gli oggetti possono essere:

la descrizione di una singola debolezza.

un elenco di identificatori a singole debolezze (organizzato secondo uno specifico criterio).

in relazione tra loro.

# Oggetto CWE: attributi 1/4

(Dettaglia una specifica debolezza)

**Abstraction:** specifica il tipo di debolezza.

Non  
programmatori

**Class:** una debolezza descritta in termini del tutto generali, senza alcun riferimento a linguaggi oppure tecnologie.

**Base:** una debolezza descritta in termini generali, ma con un dettaglio sufficiente da poter intuire le tecniche di rilevazione e prevenzione.

Programmatori  
esperti

**Variant:** una debolezza descritta nei minimi dettagli, tipicamente nell'ambito di uno specifico linguaggio o tecnologia.

# Oggetto CWE: attributi 2/4

(Dettaglia una specifica debolezza)

**Description.** Una descrizione della debolezza.

**Applicable Platforms.** Un elenco degli ambienti affetti dalla debolezza (linguaggi, server, ...).

**Common consequences.** Un elenco di effetti indesiderati (e relative conseguenze in termini di CIA).

**Likelihood of exploit.** Da “molto elevata” a “molto bassa”.



# Oggetto CWE: attributi 3/4

(Dettaglia una specifica debolezza)

**Demonstrative examples.** Frammenti di codice contenenti la debolezza, commentati in modo tale da evidenziare i problemi.

**Potential mitigations.** Un elenco di consigli per la mitigazione della debolezza, organizzato per fase di sviluppo (progetto, implementazione, operazione, ...).




# Oggetto CWE: attributi 4/4

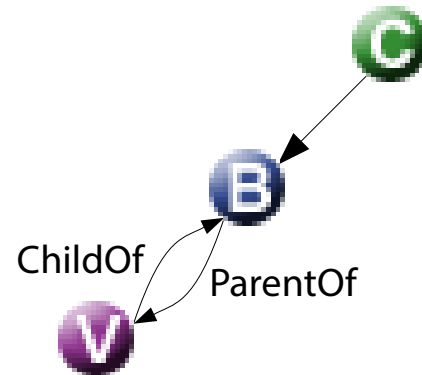
(Dettaglia una specifica debolezza)

**Relationships.** Un elenco di relazioni con altri oggetti del catalogo. Per un oggetto CWE esistono sempre le seguenti relazioni.

**ChildOf:** l'oggetto è nodo figlio di un altro elemento.

**ParentOf:** l'oggetto è nodo padre di un altro elemento.

-  Oggetto CWE Class
-  Oggetto CWE Base
-  Oggetto CWE Variant



# Un esempio di oggetto CWE

(CWE-121: Stack-based buffer overflow)

All'URL seguente:

<https://cwe.mitre.org/data/definitions/121.html>

è illustrato un tipico oggetto CWE.

Tale oggetto descrive una delle debolezze più famose: lo stack-based buffer overflow.

Il menu a tendina "Presentation Filter" seleziona più o meno attributi a seconda dell'interesse e/o livello di competenza del lettore.




# Oggetto Category

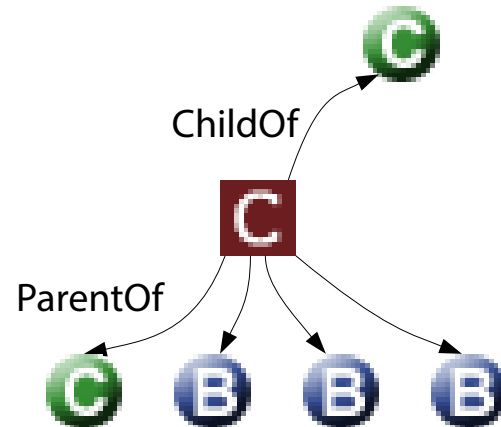
(Raccoglie diversi oggetti condividenti il valore di un attributo)

L'**oggetto Category (Category Object)** punta ad un insieme di oggetti che condividono uno specifico attributo.

Ad es., la piattaforma (J2EE, .NET).

Gli attributi sono identici a quelli di un oggetto CWE. Essendo un elemento raggruppatore, di solito ha più relazioni ParentOf che ChildOf.

-  Oggetto Category
-  Oggetto CWE Class
-  Oggetto CWE Base



# Un esempio di oggetto Category

(CWE-21: Pathname Traversal and Equivalence Errors)

All'URL seguente:

<https://cwe.mitre.org/data/definitions/21.html>

è illustrato un tipico oggetto Category.

Tale oggetto descrive una categoria di debolezze diffusa: il path traversal.

# Oggetto composto

(Mette insieme debolezze coinvolte nella stessa vulnerabilità)

Un **oggetto composto** (**Compound Object**) mette in relazione tra loro diverse debolezze implicate in una vulnerabilità.

Nella versione attuale dello standard CWE, si hanno due tipologie di oggetto composto (definite nell'attributo **Structure**).

Composite.

Chain.

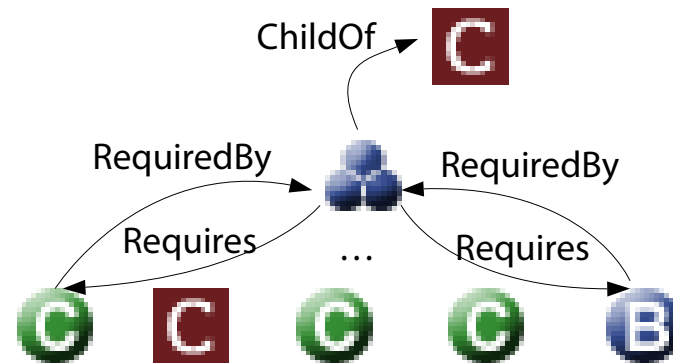
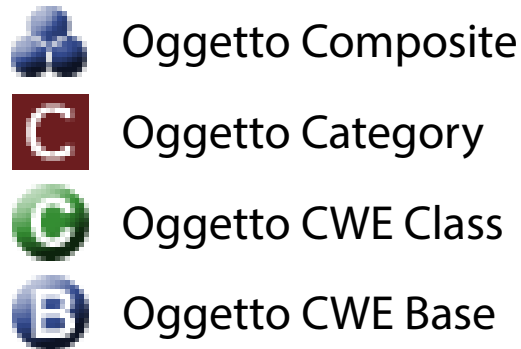
# Oggetto Composite

(Mette insieme debolezze cnella stessa vulnerabilità)

Un **oggetto Composite (Composite Object)** aggrega tutte le debolezze che, sfruttate simultaneamente, provocano una vulnerabilità.

**Requires:** relazione che lega l'oggetto Composite con un oggetto CWE).

**RequiredBy:** relazione che lega l'oggetto CWE con l'oggetto Composite.



# Un esempio di oggetto Composite

(CWE-61: UNIX Symbolic Link (Symlink) Following)

All'URL seguente:

<https://cwe.mitre.org/data/definitions/61.html>

è illustrato un tipico oggetto Composite.

Tale oggetto descrive una debolezza complessa: symbolic-link following.



# Oggetto Chain

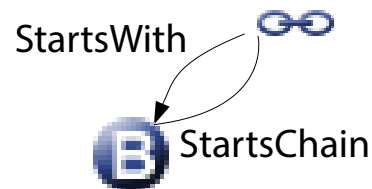
(Mette insieme debolezze coinvolte nella stessa vulnerabilità)

Un **oggetto Chain (Chain Object)** aggrega tutte le debolezze che, sfruttate in cascata, provocano una vulnerabilità. Chain è il primo anello della catena.

**StartsWith:** relazione che lega l'oggetto Chain con il successivo.

**StartsChain:** relazione inversa di StartsWith.

-  Oggetto Chain
-  Oggetto CWE Base



# Oggetto Chain

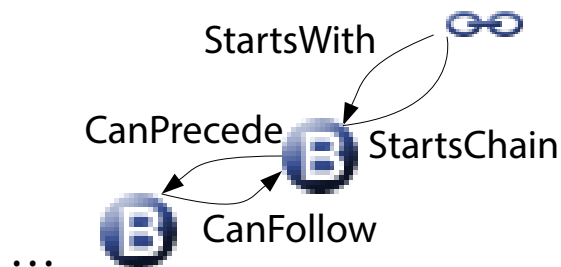
(Mette insieme debolezze coinvolte nella stessa vulnerabilità)

Gli anelli successivi della catena sono costruiti tramite relazioni.

**CanPrecede:** mette in relazione un oggetto con quello successivo nella catena.

**CanFollow:** mette in relazione un oggetto con quello precedente nella catena.

-  Oggetto Chain
-  Oggetto CWE Base



# Un esempio di oggetto Chain

(CWE-692: Incomplete Blacklist to Cross-Site-Scripting)

All'URL seguente:

<https://cwe.mitre.org/data/definitions/692.html>

è illustrato un tipico oggetto Chain.

Tale oggetto descrive una catena di debolezze tipica: da una blacklist incompleta fino ad un Cross Site Scripting.

# Oggetto View

(Raccoglie diversi oggetti condividenti il valore di un attributo)

Un **oggetto View (View Object)** punta ad un insieme di oggetti, raccolti secondo un criterio di utilità. In altre parole, è una **vista** sul catalogo.

**HasMember:** mette in relazione l'oggetto "vista" con un oggetto membro corrispondente.

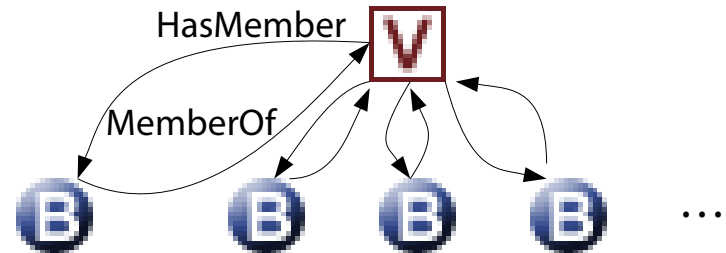
**MemberOf:** mette in relazione un oggetto membro con l'oggetto vista corrispondente.



Oggetto View



Oggetto CWE Base



# Rappresentazioni di viste

(Graph, slice)

Nella versione attuale dello standard CWE, si hanno due tipologie di vista (definite nell'attributo **Structure**).

Graph.

Slice.

# Rappresentazione graph

(Navigazione interattiva gerarchica)

**Graph.** È possibile navigare interattivamente la gerarchia degli oggetti.

<https://cwe.mitre.org/data/definitions/1000.html>

▼ Relationships

Expand All | Collapse All

**1000 - Research Concepts**

- [-] Coding Standards Violation - (710)
  - [+] Hidden Functionality - (912)
  - [+] Improper Fulfillment of API Contract ('API Abuse') - (227)
  - [-] Indicator of Poor Code Quality - (398)
    - Assignment to Variable without Use ('Unused Variable') - (563)
    - [+] Dead Code - (561)
    - Empty Synchronized Block - (585)
    - NULL Pointer Dereference - (476)

# Rappresentazione slice

(Un elenco di oggetti non navigabile gerarchicamente)

**Slice.** La gerarchia degli oggetti è presentata tramite un elenco piatto (**flat**). Non è possibile navigare gerarchicamente la gerarchia.

<https://cwe.mitre.org/data/slices/884.html>

▼ Relationships				
Nature	Type	ID	Name	V
HasMember	B	14	<a href="#">Compiler Removal of Code to Clear Buffers</a>	884
HasMember	C	22	<a href="#">Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')</a>	884
HasMember	B	23	<a href="#">Relative Path Traversal</a>	884
HasMember	B	36	<a href="#">Absolute Path Traversal</a>	884
HasMember	B	41	<a href="#">Improper Resolution of Path Equivalence</a>	884
HasMember	B	59	<a href="#">Improper Link Resolution Before File Access ('Link Following')</a>	884
HasMember	B	78	<a href="#">Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')</a>	884

# Alcune viste utili

(Per concetto, per errore di implementazione, varie top ten, per piattaforma)

All'URL <https://cwe.mitre.org/data/index.html> si può accedere ad un elenco di viste utili.

**Research/Development.** Classificazioni ottimali per i ricercatori o per gli sviluppatori.

Ricercatori → enfasi sul comportamento

Sviluppatori → enfasi sulla rilevazione

**External mappings.** Classificazioni di tipo "Top-N" fornite da organizzazioni famose.

**Helpful views.** Classificazioni comuni di debolezze.



# Domanda

(Spontanea)

Se CVE ha un sistema di punteggio (CVSS), esiste un analogo punteggio per CWE?

# Common Weakness Scoring System

(CWSS; misura la gravità di una debolezza)

Il **Common Weakness Scoring System (CWSS)** è un sistema di stima della gravità di una debolezza.

Sistema molto simile a CVSS:

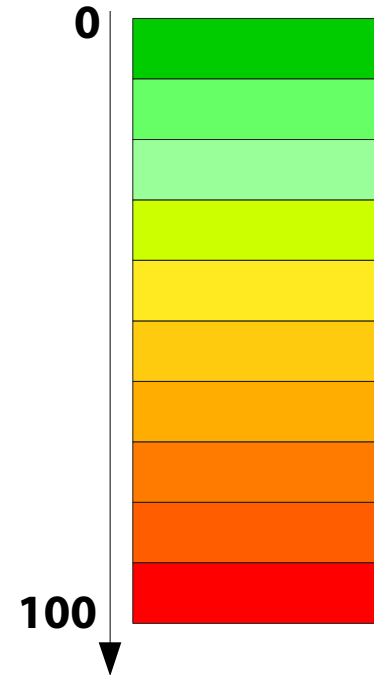
nella forma;

negli obiettivi.

Ad ogni CWE id è assegnato un **punteggio (score)** da 0 a 100.

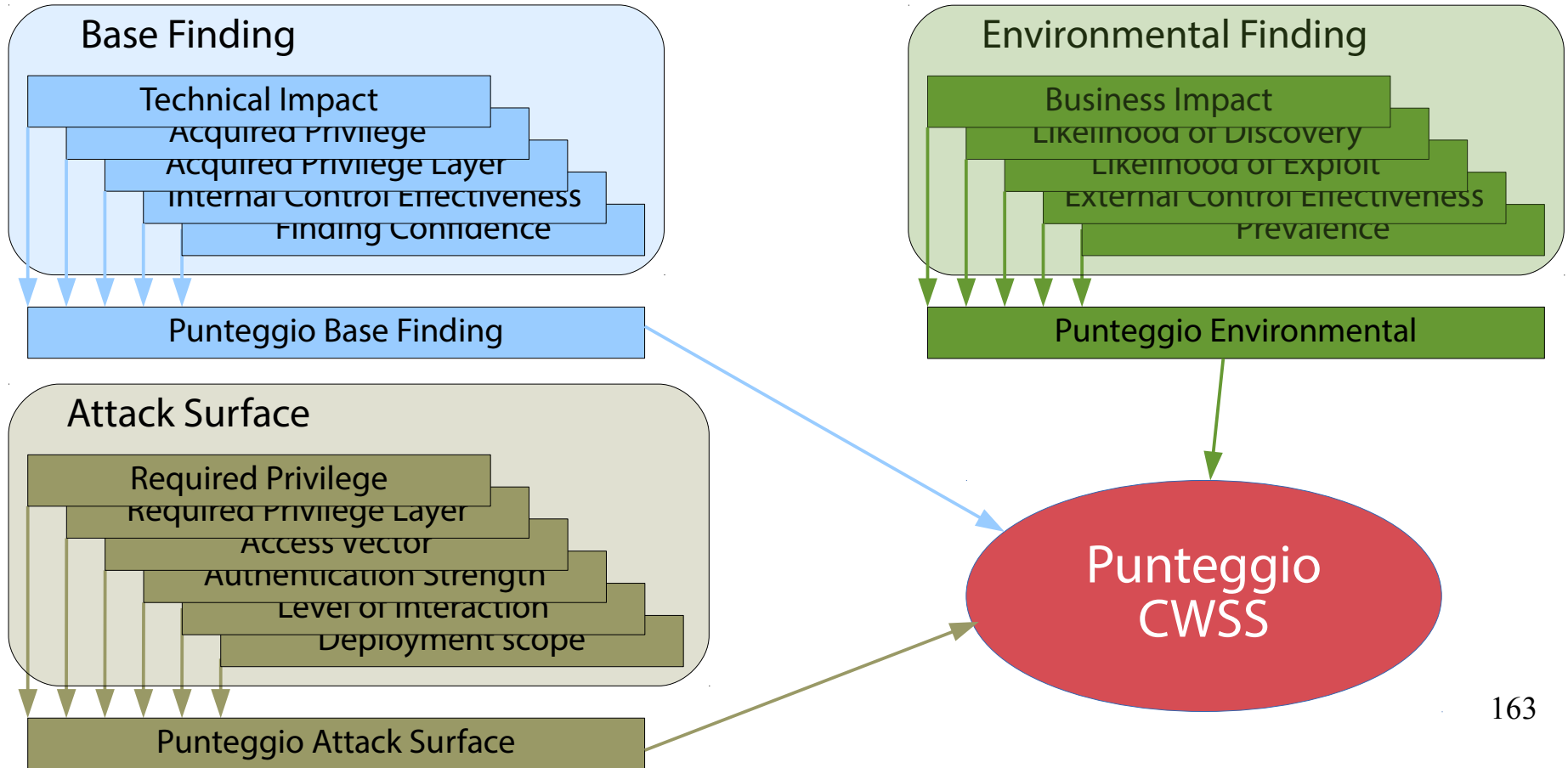
0: impatto nullo.

100: conseguenze catastrofiche.



# Il punteggio CWSS

(Funzione di tre sotto-punteggi Base, Attack Surface, Environmental)



# Le metriche CWSS

(Metrica: grandezza da misurare tramite una procedura)

Il punteggio CWSS è calcolato da tre gruppi di metriche:  
risultati di base (**Base Finding**);  
superficie di attacco (**Attack Surface**);  
ambientale (**Environmental**).

## Base Finding

Technical Impact

Acquired Privilege

Acquired Privilege Layer

Internal Control Effectiveness

Finding Confidence

## Attack Surface

Required Privilege

Required Privilege Layer

Access Vector

Authentication Strength

Level of Interaction

Deployment Scope

## Environmental

Business Impact

Likelihood of Discovery

Likelihood of Exploit

External Control Effectiveness

Prevalence

# Metriche “Base Finding”

(Stimano la gravità della debolezza in sé)

**Metriche Base Finding:** stimano il rischio della debolezza in sé, l’accuratezza della scoperta, la robustezza dei meccanismi di protezioni.

È veramente presente? È grave? È protetta bene?

## Base Finding

Technical Impact

Acquired Privilege

Acquired Privilege Layer

Internal Control Effectiveness

Finding Confidence

## Attack Surface

Required Privilege

Required Privilege Layer

Access Vector

Authentication Strength

Level of Interaction

Deployment Scope

## Environmental

Business Impact

Likelihood of Discovery

Likelihood of Exploit

External Control Effectiveness

Prevalence

# Metriche “Attack Surface”

(Stimano la difficoltà degli ostacoli da superare)

**Metriche Attack Surface:** stimano le barriere che un attaccante deve superare per sfruttare la debolezza.

Bisogna autenticarsi? Servono privilegi particolari?

## Base Finding

Technical Impact

Acquired Privilege

Acquired Privilege Layer

Internal Control Effectiveness

Finding Confidence

## Attack Surface

Required Privilege

Required Privilege Layer

Access Vector

Authentication Strength

Level of Interaction

Deployment Scope

## Environmental

Business Impact

Likelihood of Discovery

Likelihood of Exploit

External Control Effectiveness

Prevalence

# Metriche “Environmental”

(Stimano gli effetti dell’ambiente circostante)

**Metriche Environmental:** stimano le specificità legate ad uno specifico contesto operativo.

## Base Finding

Technical Impact

Acquired Privilege

Acquired Privilege Layer

Internal Control Effectiveness

Finding Confidence

## Attack Surface

Required Privilege

Required Privilege Layer

Access Vector

Authentication Strength

Level of Interaction

Deployment Scope

## Environmental

Business Impact

Likelihood of Discovery

Likelihood of Exploit

External Control Effectiveness

Prevalence

# Usi del CWSS 1/2

(I più comuni: Targeted e Generalized)

**Uso Targeted:** misurare una debolezza specifica nella fase di progetto o di implementazione di un software.

**Esempio:** stimare il rischio legato alla presenza di una debolezza “buffer overflow” nel campo “username” della funzione di autenticazione di un server FTP (linea 1234 del file `server.c`).



# Usi del CWSS 2/2

(I più comuni: Targeted e Generalized)

**Uso Generalized:** misurare una classe intera di debolezze, a prescindere dai prodotti software in cui può venirsi a trovare.

**Esempio:** stimare la pericolosità della debolezza “buffer overflow” in generale (e confrontarla con altre classi di debolezze).

# Chi usa il CWSS? 1/4

(Programmatori, capi sviluppo, acquirenti di software, responsabili sicurezza)

**Programmatori.** Sono spesso costretti a lavorare in un arco temporale limitato:

pressati da una scadenza (rilascio software imminente).

operati di incarichi (organico sottodimensionato).

→ Non sono in grado di analizzare tutte le debolezze.

→ Devono concentrarsi (a fondo) sulle debolezze più gravi (quelle con il punteggio più elevato).

# Chi usa il CWSS? 2/4

(Programmatori, capi sviluppo, acquirenti di software, responsabili sicurezza)

**Capi sviluppo.** Creano classifiche “Top-N” delle debolezze più gravi e mirano a rimuoverle dai software sotto la loro gestione.

- Devono capire le implicazioni di tali debolezze.
- Devono ridefinire le priorità in progetti diversi.
- Non devono necessariamente arrivare al dettaglio dei programmatori.

# Chi usa il CWSS? 3/4

(Programmatori, capi sviluppo, acquirenti di software, responsabili sicurezza)

**Acquirenti di software.** All'atto dell'acquisto di un software di terza parte, un cliente vuole avere la ragionevole certezza che il produttore abbia rimosso il maggior numero di debolezze possibili.

→ Vogliono capire più a fondo le debolezze del prodotto (amministratori di rete).

→ Vogliono avere una vaga idea dei rischi legati all'uso del prodotto (utenti finali).

# Chi usa il CWSS? 4/4

(Programmatori, capi sviluppo, acquirenti di software, responsabili sicurezza)

**Responsabili della sicurezza.** Hanno come obiettivo primario la minimizzazione del rischio di sicurezza della rispettiva infrastruttura.

→ Vogliono comprendere a fondo le debolezze nei prodotti software (fatti in casa o comprati).

→ Vogliono integrare CWSS nel proprio processo di sicurezza.

# Un problema

(Le metriche di CWSS non riescono sempre a catturare tutti gli interessi)

Gli utenti di CWSS sono eterogenei per competenze, interessi e finalità. Non tutte le metriche riescono a catturare l'interesse di tutti.

Ad un programmatore interessano più le metriche Base Findings rispetto alle Environmental.

Ad un manager interessano di più le metriche di tipo Environmental rispetto alle Base Findings.

→ È possibile che un utente con uno specifico insieme di competenze non sappia rispondere ad alcune domande del questionario.

# La soluzione

(Risposte “standard”, non definitive)

Per far fronte a tale problema, CWSS fornisce la possibilità di inserire alcune risposte “standard”.

Se non si sa rispondere con precisione, una di queste risposte dovrebbe essere giusta.

In seguito, con una maggiore conoscenza la domanda può essere nuovamente risposta in modo più preciso.

# Risposte standard

(Unknown, Not Applicable, Quantified, Default)

Valore	Descrizione	Punt.
Unknown (UK)	L'utente non ha informazioni sufficienti per poter rispondere alla domanda. Sono necessarie ulteriori analisi.	0.5
Not applicable (NA)	La domanda è ignorata ai fini del calcolo del punteggio. Ad esempio, un acquirente può voler ignorare tutte le domande che hanno a che fare con i rimedi.	1.0
Quantified (Q)	L'utente moltiplica il punteggio per un fattore in $[0, 1]$ . Viene data la possibilità di "smorzare" un punteggio.	
Default (D)	Media aritmetica dei punteggi sulle risposte non standard. È una alternativa ad Unknown.	



# Base Finding: Technical Impact (TI)

(Critical, High, Medium, Low, None)

Nell'ipotesi che la debolezza possa essere sfruttata con successo, qual è la principale conseguenza tecnica?

## Base Finding

Technical Impact

Acquired Privilege

Acquired Privilege Layer

Internal Control Effectiveness

Finding Confidence

Valore	Descrizione	Punt.
Critical (C)	Controllo completo; interruzione delle operazioni.	1.0
High (H)	Controllo di molte operazioni; accesso ad informazioni critiche.	0.9
Medium (M)	Controllo di alcune operazioni; accesso ad informazioni importanti.	0.6
Low (L)	Controllo minimo; accesso ad informazioni irrilevanti.	0.3
None (N)	La debolezza non porta ad una vulnerabilità.	0.0

# Base Finding: Acquired Privilege (AP)

(Administrator, Partially-privileged User, Regular User, Limited Guest, None)

Nell'ipotesi che la debolezza possa essere sfruttata con successo, che tipo di privilegi si riesce ad ottenere?

## Base Finding

Technical Impact

Acquired Privilege

Acquired Privilege Layer

Internal Control Effectiveness

Finding Confidence

Valore	Descrizione	Punt.
Administrator (A)	L'attaccante diventa amministratore ( <code>root</code> in UNIX, <code>SYSTEM</code> in Windows, <code>admin</code> su un router).	1.0
Partially Privileged User (P)	L'attaccante diventa un utente con alcuni privilegi, ma non tutti quelli di un amministratore.	0.9
Regular User (RU)	L'attaccante diventa un utente normale, senza privilegi particolari.	0.7
Limited or Guest (L)	L'attaccante diventa un utente con privilegi ristretti (ad esempio, <code>nobody</code> su UNIX).	0.6
None (N)	L'attaccante non riesce a diventare un utente.	0.1

# Base Finding: Acquired Privilege Layer (AL)

(Application, System, Network, Enterprise Infrastructure)

Nell'ipotesi che la debolezza possa essere sfruttata con successo, a che livello operativo si ottengono i privilegi?

## Base Finding

Technical Impact

Acquired Privilege

Acquired Privilege Layer

Internal Control Effectiveness

Finding Confidence

Valore	Descrizione	Punt.
Application (A)	L'attaccante acquisisce privilegi a livello di utente di una applicazione software.	1.0
System (S)	L'attaccante acquisisce privilegi a livello di utente di un sistema operativo.	0.9
Network (N)	L'attaccante acquisisce il privilegio di accesso alla rete.	0.7
Enterprise Infrastructure (E)	L'attaccante acquisisce l'accesso ad una porzione dell'infrastruttura (router, switch, DNS, controller di dominio, firewall, ... ).	1.0

# Base Finding: Internal Control Effectiveness (IC)

(None, Limited, Moderate, Indirect, Best-Available, Complete)

Qual è l'efficacia delle contromisure interne (a livello di codice)?

## Base Finding

Technical Impact

Acquired Privilege

Acquired Privilege Layer

Internal Control Effectiveness

Finding Confidence

Valore	Descrizione	Punt.
None (N)	Non esistono contromisure.	1.0
Limited (L)	Esiste un meccanismo semplice o fortuito, in grado di rintuzzare un attaccante occasionale.	0.9
Moderate (M)	Esiste un meccanismo standard con dei limiti, aggirabile con un po' di impegno da un esperto.	0.7
Indirect (I)	Un meccanismo non specifico per la debolezza ne riduce l'impatto in maniera indiretta.	0.5
Best-Available (B)	È implementato il meccanismo migliore noto. Un attaccante esperto e determinato potrebbe aggirarlo con l'aiuto di altre debolezze.	0.3
Complete (C)	Il meccanismo impedisce lo sfruttamento.	0.0

# Una assunzione azzardata

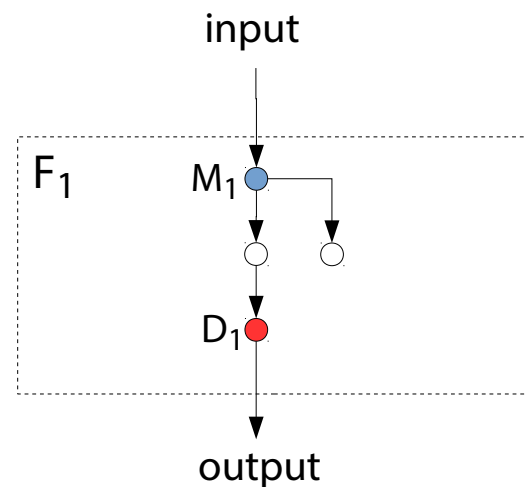
(Una debolezza → un meccanismo di protezione)

La domanda precedente fa una assunzione piuttosto importante.

Una debolezza è protetta da un solo meccanismo.

Ciò non è vero in generale.

- M: meccanismo
- F: funzione
- D: debolezza
- P: Punteggio risposta

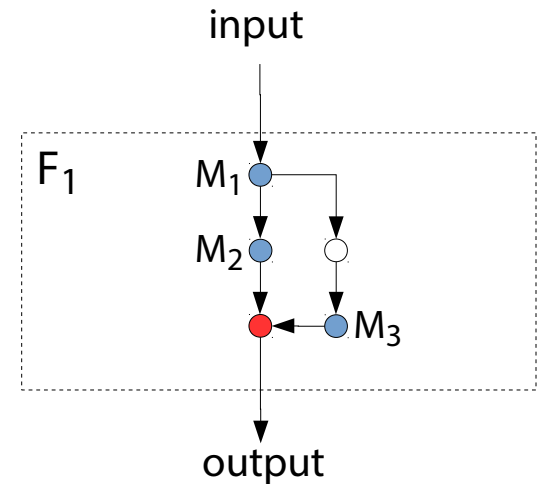


Punteggio = P<sub>1</sub>

# Un problema

(Come calcolare CWSS in presenza di più meccanismi di protezione?)

Come si calcola il punteggio della risposta in presenza di più meccanismi di protezione?

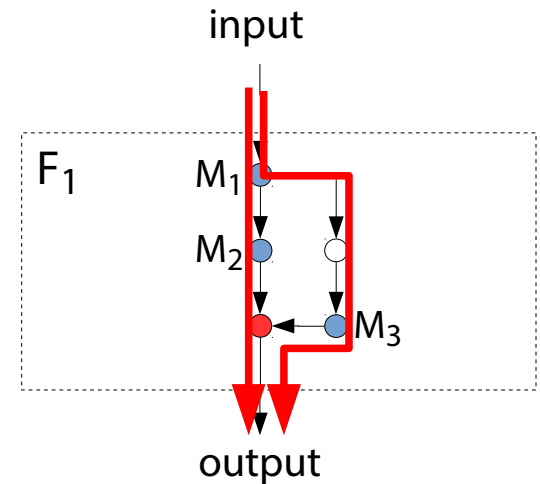


$P = ?$

# Soluzione: Passo 1

(Individuazione percorsi di codice con almeno un M e F)

Si individuano inizialmente tutti i percorsi di codice che:  
partono da *input* e terminano in *output*;  
contengono almeno un M e F.  
La debolezza può essere sfruttata solo tramite questi percorsi.



Punteggio = ?

# Soluzione: Passo 2

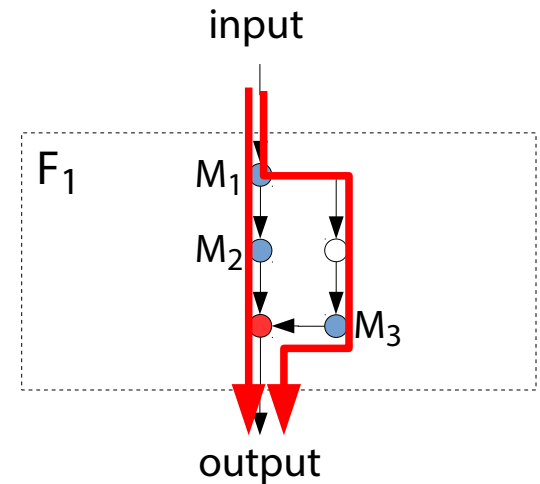
(Individuazione del meccanismo più robusto in ogni percorso)

Per ogni percorso, si individui il meccanismo più robusto.

Lo scoglio più arduo da aggirare. È sufficiente scegliere il meccanismo con il punteggio più basso ( $\rightarrow$  difficoltà di violazione più elevata).

$$\min(P_1, P_2).$$

$$\min(P_1, P_3).$$



Punteggio = ?



# Soluzione: Passo 3

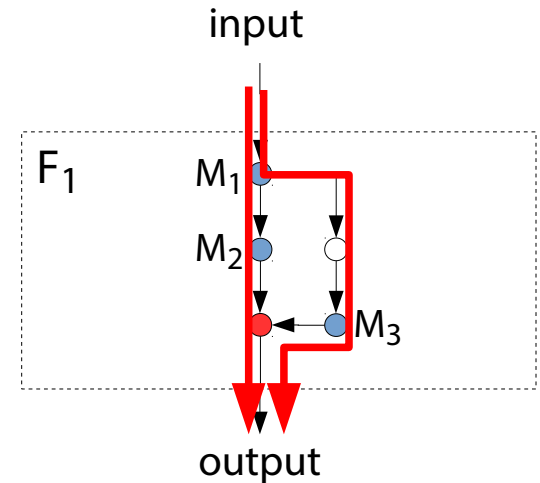
(Individuazione del percorso più debole)

Infine, si sceglie il percorso più debole.

Il più comodo per l'attaccante.

È sufficiente scegliere il percorso con il punteggio minimo più alto (→ difficoltà di violazione più bassa).

$$\text{Punteggio} = \max(\min(P_1, P_2), \min(P_1, P_3))$$



# Please remember this!

(There's no point in securing strong links; focus on the weak ones. Please.)

**“Security is a chain; it’s only as secure as the weakest link.”**

*Bruce Schneier (1963-)*

*Esperto internazionale di sicurezza*

*Apprezzato suonatore di bonghi*



# Base Finding: Finding Confidence (FC)

(Proven True, Proven Locally True, Proven False)

Quanto si è sicuri che il difetto/bug individuato sia una debolezza?  
possa essere usato da un attaccante?

Base Finding

Technical Impact

Acquired Privilege

Acquired Privilege Layer

Internal Control Effectiveness

Finding Confidence

Valore		Descrizione	Punt.
Proven True (T)	True	La debolezza esiste ed è raggiungibile da un attaccante.	1.0
Proven Locally True (LT)	True	La debolezza esiste, ma non è chiaro se sia o meno sfruttabile da un attaccante.	0.8
Proven False (F)	False	Il difetto/bug non costituisce una debolezza e/o non è sfruttabile da un attaccante.	0.0

# Calcolo del punteggio “Base Finding”

(Tramite un insieme perverso di formule)

**Il Punteggio Base Finding (Base Finding Score)** è calcolato nel modo seguente.

$$f(TI) = \begin{cases} 0 & \text{if } TI = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$BaseFindingScore = [(10 * TI + 5 * (AP + AL) + 5 * FC) * f(TI) * IC] * 4.0$$

**OSS.:** BaseFindingScore  $\in$  [0, 100].

# Attack Surface: Required Privilege (RP)

(None, Limited/Guest, Regular User, Partially-Privileged User, Administrator)

Quali privilegi deve già possedere un attaccante per sfruttare la debolezza?

## Attack Surface

Required Privilege

Required Privilege Layer

Access Vector

Authentication Strength

Level of Interaction

Deployment Scope

Valore	Descrizione	Punt.
None (N)	Non sono richiesti privilegi particolari.	1.0
Limited / Guest (L)	L'attaccante deve già avere i privilegi di un utente ristretto.	0.9
Regular User (RU)	L'attaccante deve già avere i privilegi di un utente normale.	0.7
Partially Privileged User (P)	L'attaccante deve già avere i privilegi di un utente speciale (con alcuni privilegi in più rispetto ad uno normale, ma non tutti quelli di un amministratore).	0.6
Administrator (A)	L'attaccante deve già avere i privilegi di un utente amministratore.	0.1

# Attack Surface: Required Privilege Layer (RL)

(Application, System, Network, Enterprise Infrastructure)

A quale livello operativo deve l'attaccante avere già privilegi per poter sfruttare la debolezza?

## Attack Surface

Required Privilege

Required Privilege Layer

Access Vector

Authentication Strength

Level of Interaction

Deployment Scope

Valore	Descrizione	Punt.
Application (A)	L'attaccante deve già avere privilegi applicativi.	1.0
System (S)	L'attaccante deve già avere privilegi a livello di sistema operativo.	0.9
Network (N)	L'attaccante deve già avere i privilegi di accesso alla rete.	0.7
Enterprise Infrastructure (E)	L'attaccante deve già avere i privilegi a livello di infrastruttura (router, switch, DNS, controller di dominio, firewall, ...).	1.0

# Attack Surface: Access Vector (AV)

(Internet, Intranet, Private Network, Adjacent Network, Local, Physical)

Attraverso quale canale deve comunicare l'attaccante per sfruttare la debolezza?

## Attack Surface

Required Privilege

Required Privilege Layer

Access Vector

Authentication Strength

Level of Interaction

Deployment Scope

Valore	Descrizione	Punt.
Internet (I)	L'attaccante deve avere accesso ad Internet.	1.0
Intranet (R)	L'attaccante deve avere accesso ad una Intranet schermata da un proxy Web.	0.8
Private Network (V)	L'attaccante deve avere accesso ad una rete privata disponibile solo ad alcuni utenti fidati.	0.8
Adjacent Network (A)	L'attaccante deve avere accesso fisico al dominio di broadcast o di collisione della rete.	0.7
Local (L)	L'attaccante deve avere accesso locale ad una shell.	0.5
Physical (P)	L'attaccante deve avere accesso fisico all'asset.	0.2

# Attack Surface: Authentication Strength (AS)

(None, Weak, Moderate, Strong)

Qual è la forza della procedura di autenticazione a protezione della debolezza?

## Attack Surface

Required Privilege

Required Privilege Layer

Access Vector

Authentication Strength

Level of Interaction

Deployment Scope

Valore	Descrizione	Punt.
None (N)	Non è prevista alcuna forma di autenticazione.	1.0
Weak (W)	È prevista una autenticazione debole (username e password).	0.9
Moderate (M)	È prevista una autenticazione moderatamente forte (uso di certificati, autenticazione basata su conoscenza, one-time password).	0.8
Strong (S)	È prevista una autenticazione forte (token hardware, multi-fattore).	0.7



# Attack Surface: Level of Interaction (IN)

(Automated, Typical/Limited, Moderate, Opportunistic, High, No interaction)

Quali azioni deve compiere la vittima per permettere lo svolgimento con successo di un attacco?

## Attack Surface

Required Privilege

Required Privilege Layer

Access Vector

Authentication Strength

Level of Interaction

Deployment Scope

Valore	Descrizione	Punt.
Automated (A)	Non è richiesta interazione umana.	1.0
Typical / Limited (T)	L'attaccante deve convincere l'utente a svolgere una azione normale nel contesto del software.	0.9
Moderate (M)	L'attaccante deve convincere l'utente a svolgere una azione sospetta per un conoscente della sicurezza.	0.8
Opportunistic (N)	L'attaccante non può controllare direttamente la vittima; può solo capitalizzare errori altrui.	0.3
High (H)	L'attaccante deve usare il social engineering.	0.1
No Interaction (NI)	Non è possibile alcuna interazione.	0.0

# Attack Surface: Deployment Scope (SC)

(All, Moderate, Rare, Potentially Reachable)

In quali piattaforme e/o configurazioni si manifesta la debolezza?

## Attack Surface

Required Privilege

Required Privilege Layer

Access Vector

Authentication Strength

Level of Interaction

Deployment Scope

Valore	Descrizione	Punt.
All (A)	La debolezza si manifesta in tutte le piattaforme ed in tutte le configurazioni.	1.0
Moderate (M)	La debolezza si manifesta nelle piattaforme e/o nelle configurazioni più comuni.	0.9
Rare (R)	La debolezza si manifesta solo in piattaforme e/o nelle configurazioni più rare.	0.5
Potentially Reachable (P)	La debolezza è potenzialmente sfruttabile. In questo specifico istante tutti i percorsi di codice sembrano sicuri e/o la debolezza è codice "morto" (non raggiungibile in pratica).	0.1

# Calcolo del punteggio "Attack Surface"

(Tramite un insieme perverso di formule)

**Il Punteggio Attack Surface (Attack Surface Score)** è calcolato nel modo seguente.

$$f(TI) = \begin{cases} 0 & \text{if } TI = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\text{AttackSurfaceScore} = [20 * (RP + RL + AV) + 20 * SC + 15 * IN * 5 * AS] * 100.0$$

**OSS.** AttackSurfaceScore  $\in [0, 1]$ .

# Environmental: Business Impact (BI)

(Critical, High, Medium, Low, None)

Qual è l'impatto aziendale di uno sfruttamento della debolezza?

Environmental

Business Impact

Likelihood of Discovery

Likelihood of Exploit

External Control Effectiveness

Prevalence

Valore	Descrizione	Punt.
Critical (C)	L'azienda può fallire.	1.0
High (H)	Le operazioni aziendali sono colpite gravemente.	0.9
Medium (M)	Alcune operazioni aziendali sono colpite, ma non quelle più comuni.	0.6
Low (L)	L'impatto aziendale è minimo.	0.3
None (N)	Non vi è impatto aziendale alcuno.	0.0

# Environmental: Likelihood of Discovery (DI)

(High, Medium, Low)

Qual è la probabilità che un attaccante riesca a scoprire la debolezza?

## Environmental

Business Impact

Likelihood of Discovery

Likelihood of Exploit

External Control Effectiveness

Prevalence

Valore	Descrizione	Punt .
High (H)	È molto probabile che un attaccante riesca a scoprire la debolezza usando tecniche semplici e senza accesso al codice sorgente del software.	1.0
Medium (M)	Un attaccante potrebbe riuscire a scoprire la debolezza, ma solo con accesso al codice sorgente del software e tanto tempo a disposizione.	0.6
Low (L)	È improbabile che un attaccante riesca a scoprire la debolezza senza avere capacità particolari, accesso al codice sorgente e tanto tempo a disposizione.	0.2

# Environmental: Likelihood of Exploit (EX)

(High, Medium, Low, None)

Qual è la probabilità che, una volta scoperta la debolezza, un attaccante con il giusto privilegio, autenticazione, accesso sia in grado di sfruttarla?

Environmental

Business Impact

Likelihood of Discovery

Likelihood of Exploit

External Control Effectiveness

Prevalence

Valore	Descrizione	Punt
High (H)	È molto probabile che un attaccante riesca a sfruttare la debolezza tramite un exploit di facile implementazione.	1.0
Medium (M)	Un attaccante potrebbe riuscire a sfruttare la debolezza. Le probabilità di successo variano; potrebbero essere necessari più tentativi.	0.6
Low (L)	È improbabile che un attaccante riesca a sfruttare la debolezza.	0.2
None (N)	L'attaccante non ha alcuna chance di successo.	0.0198

# Environmental: External Control Effectiveness (EC)

(None, Limited, Moderate, Indirect, Best-Available, Complete)

Qual è l'efficacia delle contromisure esterne (NON a livello di codice)?

Valore	Descrizione	Punt.
None (N)	Non esistono contromisure.	1.0
Limited (L)	Esiste un meccanismo semplice o fortuito, in grado di rintuzzare un attaccante occasionale.	0.9
Moderate (M)	Esiste un meccanismo standard con dei limiti, aggirabile con un po' di impegno da un esperto.	0.7
Indirect (I)	Un meccanismo non specifico per la debolezza ne riduce l'impatto in maniera indiretta.	0.5
Best-Available (B)	È implementato il meccanismo migliore noto. Un attaccante esperto e determinato potrebbe aggirarlo con l'aiuto di altre debolezze.	0.3
Complete (C)	Il meccanismo impedisce lo sfruttamento.	0.1

Environmental

Business Impact

Likelihood of Discovery

Likelihood of Exploit

External Control Effectiveness

Prevalence

# Environmental: Prevalence (P)

(Widespread, High, Common, Limited)

Qual è la frequenza di occorrenza della debolezza nel software in generale?

Environmental

Business Impact

Likelihood of Discovery

Likelihood of Exploit

External Control Effectiveness

Prevalence

Valore	Descrizione	Punt.
Widespread (W)	La debolezza è presente nella maggioranza (se non la totalità) dei software in esecuzione nella infrastruttura considerata.	1.0
High (H)	La debolezza si incontra spesso, ma non è diffusa su ampio spettro.	0.9
Common (C)	La debolezza si incontra di tanto in tanto.	0.8
Limited (L)	La debolezza si incontra raramente (oppure, mai).	0.7



# Calcolo del punteggio “Environmental”

(Tramite un insieme perverso di formule)

**Il Punteggio Environmental (Environmental Score)** è calcolato nel modo seguente.

$$f(BI) = \begin{cases} 0 & \text{if } BI = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\text{EnvironmentalScore} = [(10 * BI + 3 * DI + 4 * EX + 3 * P) * f(BI) * EC] * 20.0$$

**OSS.:** EnvironmentalScore  $\in [0, 1]$ .

# Il vettore CWSS

(Rappresenta sinteticamente le risposte al questionario)

Il **vettore CWSS (CWSS vector)** è una stringa che riassume sinteticamente le risposte alle domande del questionario.

A differenza del CVSS, nel CWSS non tutti i valori possono essere descritti con un numero discreto.

Si pensi alla possibilità di moltiplicare i valori delle risposte per un coefficiente arbitrario in  $[0, 1]$  (risposta di default "Quantified").

# Il vettore CWSS

(Presenta una leggera differenza rispetto al vettore CVSS)

Il formato di un vettore CWSS è una serie di terne *abbr\_domanda:abbr\_risposta:peso\_Q*, separate dal carattere /.

Ad es. (Base Finding):

TI:H,0.9/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0

# Il foglio di calcolo CWSS

(Permette di calcolare il punteggio CWSS con pochi click)

All'URL seguente:

<https://cwss.dcnws.com/>

è presente un foglio di calcolo Web (non ufficiale) per la stima dei punteggi CWSS (Base Finding, AttackSurface ed Environmental).

# Il foglio di calcolo CWSS

(Permette di calcolare il punteggio CWSS con pochi click)

All'URL seguente:

<https://cwss.cisecurity.com/>

è presente un foglio di calcolo Web (non ufficiale) per la stima dei punteggi CWSS (Base Finding, Attacksurface ed Environmental).

Il foglio è attualmente offline. È disponibile un foglio di calcolo alternativo all'URL seguente:

<https://github.com/g4xyk00/CWSS-Calculator>

# Un limite del foglio di calcolo

(Il foglio di calcolo non implementa benissimo i coefficienti Q)

Il foglio di calcolo proposto non gestisce benissimo i coefficienti moltiplicativi Quantified.

Se non si seleziona una risposta Quantified, si assume un coefficiente  $Q=1.0$  (non è detto che sia così nella realtà).

Se si seleziona una risposta Quantified, è il foglio di calcolo a scegliere un valore predefinito (e non l'utente).

# La (triste) conseguenza




(Il punteggio calcolato con il foglio può differire da quello teorico)

→ Il punteggio CWSS calcolato con il foglio di calcolo può essere radicalmente diverso da quello teorico (calcolato con carta e penna).

# Struttura del foglio di calcolo CWSS

(Tre tab espandibili mediante un semplice click)

Il foglio di calcolo consta di tre tab espandibili tramite l'icona ∨ sulla destra.

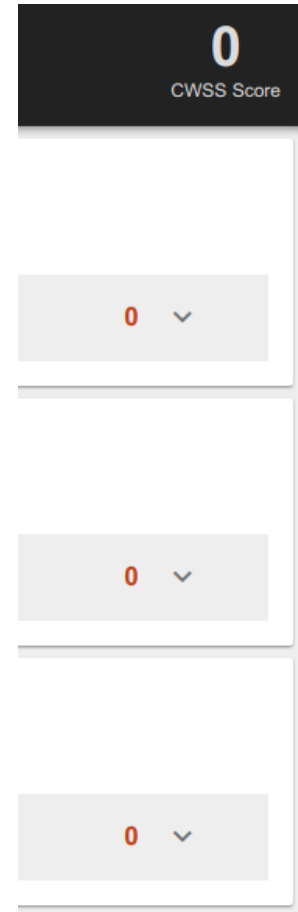
 <b>Base Finding</b> Inherent Risk, Confidence in Accuracy, Strength of Controls	<b>Score</b> 0 ∨
 <b>Attack Surface</b> Barriers that an Attacker Must Overcome	<b>Score</b> 0 ∨
 <b>Environmental</b> Environment Specific Characteristics of Weakness	<b>Score</b> 0 ∨



# I punteggi CWSS

(Tre numeri visibili sul lato destro dell'applicazione)

I tre sotto-punteggi ed il punteggio totale sono visibili sulla parte destra dell'applicazione Web.



# Immissione metriche

(Tramite una pulsantiera per le risposte)

Cliccando su una delle icone  $\nabla$ , la singola scheda si espande in una pulsantiera contenente le risposte possibili e le relative descrizioni.



Base Finding  
Inherent Risk, Confidence in Accuracy, Strength of Controls

Score				
Technical Impact	Acquired Privilege	Privilege Layer	Control Effectiveness	Finding Confidence
CRITICAL	ADMINISTRATOR	APPLICATION	NONE	PROVEN TRUE
HIGH	PRIVILEGED USER	SYSTEM	LIMITED	LOCALLY TRUE
MEDIUM	REGULAR USER	NETWORK	MODERATE	PROVEN FALSE
LOW	LIMITED / GUEST	INFRASTRUCTURE	INDIRECT	DEFAULT
NONE	NONE	DEFAULT	BEST-AVAILABLE	UNKNOWN
DEFAULT	DEFAULT	UNKNOWN	COMPLETE	NOT APPLICABLE
UNKNOWN	UNKNOWN	NOT APPLICABLE	DEFAULT	QUANTIFIED
NOT APPLICABLE	NOT APPLICABLE	QUANTIFIED	UNKNOWN	
QUANTIFIED	QUANTIFIED		NOT APPLICABLE	
			QUANTIFIED	

The Base Finding metric group captures the inherent risk of the weakness, confidence in the accuracy of the finding, and strength of controls.

An Internal Control is a control, protection mechanism, or mitigation that has been explicitly built into the software (whether through architecture, design, or implementation). Internal Control Effectiveness measures the ability of the control to render the weakness unable to be exploited by an attacker. For example, an input validation routine that restricts input length to 15 characters might be moderately effective against XSS attacks by reducing the size of the XSS exploit that can be attempted.

Significant control over the software being analyzed, or access to critical information can be obtained.

# Un esempio concreto

(Vale spesso più di 0x3e8 parole)

La A.C.M.E. SrL\* è una azienda che produce beni e servizi di ogni tipo per il pubblico.

A.C.M.E. espone al pubblico un server Web contenente:

- un catalogo dei prodotti;
- un negozio elettronico.

Una delle applicazioni coinvolte ha una debolezza.



\* A.C.M.E. SrL è un nome di fantasia e non ha alcuna attinenza con fatti e/o personaggi reali.

# Descrizione della debolezza 1/2

(Serve per poter rispondere al questionario)

L'applicazione è una fonte di guadagno primaria per la A.C.M.E. (ha valore di business critico).

L'applicazione permette ad un utente arbitrario di registrare un account usando solo un indirizzo di posta elettronica.

Un attaccante può sfruttare la debolezza ed ottenere i privilegi di amministratore nel contesto dell'applicazione.

# Descrizione della debolezza 2/2

(Serve per poter rispondere al questionario)

L'attacco non può avere successo fino a quando un amministratore non controlla le richieste di registrazione.

L'attaccante può solo cancellare utenti e dati, ma non ha il pieno controllo dell'applicazione.

L'applicazione non ha meccanismi di protezione della debolezza.

Il fix della debolezza è semplice (poche righe di codice).

# Punteggio Base Finding

(Il primo sotto-punteggio)

Metrica	Valore	Fattore
Technical Impact (TI)	High	0.9
Acquired Privilege (AP)	Administrator	1.0
Acquired Privilege Layer (AL)	Application	1.0
Internal Control Effectiveness (IC)	None	1.0
Finding Confidence (FC)	Proven True	1.0

Punteggio teorico = 92.6

Vettore teorico = TI:H,0.9/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0

Punteggio foglio calcolo = 96

Vettore foglio calcolo = TI:H,1.0/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0

# Punteggio Attack Surface

(Il secondo sotto-punteggio)

Metrica	Valore	Fattore
Required Privilege (RP)	Guest	0.9
Required Privilege Layer (RL)	Application	1.0
Access Vector (AV)	Internet	1.0
Authentication Strength (IC)	None	1.0
Level of Interaction (IN)	Typical/Limited	1.0
Deployment Scope (SC)	All	1.0

Punteggio teorico = 0.965

Vettore teorico = RP:L,0.9/RL:A,1.0/AV:I,1.0/AS:N,1.0/IN:T,0.9/SC:A,1.0

Punteggio foglio calcolo = 0.97

Vettore foglio calcolo = RP:L,1.0/RL:A,1.0/AV:I,1.0/AS:N,1.0/IN:T,1.0/SC:A,1.0 <sub>215</sub>

# Punteggio Environmental

(Il terzo sotto-punteggio)

Metrica	Valore	Fattore
Business Impact (BI)	Critical	0.9
Likelihood of Discovery (DI)	High	1.0
Likelihood of Exploit (EX)	High	1.0
External Control Effectiveness (EC)	None	1.0
Prevalence (P)	Not Applicable	1.0

Punteggio teorico = 1.0

Vettore teorico = BI:C,0.9/DI:H,1.0/EX:H,1.0/EC:N,1.0/P:NA,1.0

Punteggio foglio calcolo = 1.0

Vettore foglio calcolo = BI:C,1.0/DI:H,1.0/EX:H,1.0/EC:N,1.0/P:NA,1.0



# Il punteggio finale

(Il prodotto dei tre sotto-punteggi)

Punteggio finale teorico: 92.6

Vettore finale teorico:

TI:H,0.9/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0/  
RP:L,0.9/RL:A,1.0/AV:I,1.0/AS:N,1.0/IN:T,0.9/SC:A,1.0/  
BI:C,0.9/DI:H,1.0/EX:H,1.0/EC:N,1.0/P:NA,1.0

Punteggio finale foglio calcolo: 94.0

Vettore finale foglio calcolo:

TI:H,1.0/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0/  
RP:L,1.0/RL:A,1.0/AV:I,1.0/AS:N,1.0/IN:T,1.0/SC:A,1.0/  
BI:C,1.0/DI:H,1.0/EX:H,1.0/EC:N,1.0/P:NA,1.0

# Qualche osservazione 1/3

(Base Finding)

Il rischio legato alla debolezza è molto elevato.

Il punteggio Base Finding è 92.6 (vicino a 100.0).

La debolezza esiste per davvero.

La debolezza è sfruttabile facilmente.

La debolezza non è protetta a livello di codice.

# Qualche osservazione 2/3

(Attack Surface)

Le barriere poste ad un attaccante sono poche.  
Il punteggio Attack Surface è 0.965 (vicino a 1.0).  
Non sono necessarie particolari credenziali.  
Si può condurre l'attacco da remoto tramite Internet.  
È richiesta una scarsa interazione con gli umani.  
La debolezza è presente in tutti i sistemi analizzati.

# Qualche osservazione 3/3

(Environmental)

Le condizioni “ambientali” non sono favorevoli.

Il punteggio Environmental è 1.0 (il peggiore).

Se l'applicazione è violata, le conseguenze per il business sono fatali.

La debolezza è facile da scoprire.

La debolezza è facile da sfruttare.

La debolezza non è protetta a livello operativo (firewall, etc.).

# CVE Details

(Un motore di ricerca per CVE & Co.)

Il servizio CVE Details, disponibile all'URL:

<https://www.cvedetails.com>

è un motore di ricerca per CVE.

Interfaccia d'accesso semplice a dati collegati alle vulnerabilità.

Aggregatore di dati provenienti da sorgenti diverse.

Database navigabile per vendor, prodotto, versione, tipologia di vulnerabilità, data di segnalazione, ...

Visualizzazione di statistiche, trend, report.

Integrazione con OVAL (verifica di vulnerabilità e di patch).

# Sorgenti dati utilizzate

(Diverse)

