Executive Summary

I was contracted to conduct a penetration test in order to determine its exposure to a targeted attack. All activities were conducted in a manner that simulated a malicious actor engaged in a targeted attack against **this login page** with the goals of:

1. Identifying if a remote attacker could 83.212.174.87 defenses.
2. Determine the impact of a security breach on:
   a) Confidentiality of the its private data
   b) Internal infrastructure and availability of 83.212.174.87 information system

Efforts were placed on the identification and exploitation of security weakness that could allow a remote attacker to gain unauthorized access to the database. The attacks were conducted with the level of access that a general internet user would have.

All tests and actions being conducted under controlled conditions.

SCOPE

Activity performed a Web Application Security Assessment of web portal (83.212.174.87)

The application is internet facing and requires password identity for secure access.

The landing page to the application under review was at the following address:

URL : http://83.212.174.87/login.php

Client (milkatos7) defined the following application URL and web server host as in scope:

URL : http://83.212.174.87

My testing included both unauthenticated as well as authenticated testing.

Attack Narrative

Remote System Discovery

This section provides details on the open ports and remote system discovery

This table shows the open port on the system, not each open port is a security threat, but open ports on the system are invitations to the attackers. In general, the number of open ports should be kept to a minimum and only the mission-critical ports should be open.

PORT NUMBER	Services
22(tcp)	ssh
23(tcp)	telnet
25(tcp)	smtp
80 (tcp)	HTTP
2222 (tcp)	ssh
2323/tcp	3d - nfsd

Screenshots:

```
root@kali:~# nmap -sV 83.212.174.87
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-06 12:1
Nmap scan report for cs-unipi-sec.vm.grnet.gr (83.212.174.
Host is up (0.19s latency).
Not shown: 994 closed ports
PORT     STATE    SERVICE VERSION
22/tcp   open     ssh     OpenSSH 6.0p1 Debian 4+deb7u2 (p
23/tcp   filtered telnet
25/tcp   open     smtp?
80/tcp   open     http    Apache httpd 2.4.38 ((Debian))
2222/tcp open     ssh     OpenSSH 6.0p1 Debian 4+deb7u2 (p
2323/tcp filtered 3d-nfsd
```

Remote operating system : Linux Kernel 3.2 on Debian 7.0 (wheezy)

Banner Grabbing & Version Detection

This table provides general details of Banner and Version Detection.

Target Banner - 80	Apache HTTP Server 2.4.38 (Port 80)

Screenshot:

Port	Protocol	State	Name	
22	tcp	open	ssh	OpenSSH 6.0p1 De
80	tcp	open	http	Apache httpd 2.4.3
2222	tcp	open	ssh	OpenSSH 6.0p1 De

Load Balancer and Firewall Detection:

I found that the IP : 83.212.174.87 has no Load balancer and no Firewall in place.

But the landing URL has no protection whatsoever. Check the Below screenshot :

```
                    ,,
          .-. .-. 80        tcp          open     ) (_
         ()``; |==|_______)                        ( |__| http
         / ('              tcp          open     .)|__|
        (  /  )          /|\                      ( |__| ssh
         \(_)_))        / | \                      . |__|
                       /  |  \                      |__|

    WAFW00F - Web Application Firewall Detection Tool


Checking http://83.212.174.87
Generic Detection results:
No WAF detected by the generic detection
Number of requests: 7
```

DNS Penetration Test

For the purposes of this assessment, **milkatos7** provided minimal information of the organizational domain name: https://83.212.174.87/login.php

The name of this machine either does not resolve or resolves to a different IP address.

IP Analysis

I have found the IP address of **the login page** named is 83.212.174.87

IP = 383.212.174.87**.**
IP registrar is Greek Research and Technology Network (GRNET) S.A.

Web App Built with Following technologies:

See the table below:

Sr. No	Technology Used
1	Apache 2.4
2	**Iphone/Mobile Compatible**
3	**HTML5**
4	**php**

File Guessing Attack

Risk: High

It is sometimes possible to find interesting contents on a web site simply by "snooping" around.

Sometimes there are backup of files or older versions of live code, or perhaps vulnerable sample application pages on the web site. When accessing sensitive patient data, application relies on dynamic tokens that change with each request.

Conclusion: I attempted various URL brute-forcing for common file names and found no file which has to be hidden.

Intruder attack

Attack Save Columns

Results | Target | Positions | Payloads | Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Leng
3602	login.php	200			944
5152		200			1863
3438	javascript	301			551
5	%EXT%	400			487
496	2257.%EXT%	400			487
564	ASPSamp/AdvWorks/equipmen...	400			487
565	AccessDenied.%EXT%	400			487
580	Adm.%EXT%	400			487
591	Admin.%EXT%	400			487
601	Admin/knowledge/dsmgr/users...	400			487
602	Admin/knowledge/dsmgr/users...	400			487
603	Admin/login.%EXT%	400			487
616	Administracao.%EXT%	400			487
617	Administracion.%EXT%	400			487
618	Administrateur.%EXT%	400			487
619	Administration.%EXT%	400			487
623	Administrator.%EXT%	400			487

Request | Response

Raw | Headers | Hex

```
GET /javascript HTTP/1.1
Host: 83.212.174.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefo
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
```

? < + > Type a search term

Finished

Password Brute Force

Risk: High

A **brute force attack** is a trial-and-error method used to obtain information such as a user **password** or personal identification number (PIN). In a **brute force attack**, automated software is used to generate a large number of consecutive guesses as to the value of the desired data.

I performed a rigorous brute force attack on the login page with a wordlist of around 80,000 most commonly used passwords around the world and found no success. This simply means that the password set for the portal is either strong or not common.

Intruder attack

Attack Save Columns

Results | Target | Positions | Payloads | Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Leng
3281	antidisestablishmentarianism	200			972
8862	bostreep-ghieliemientjie	200			968
16712	counter-revolutionaries	200			967
23859	electroencephalographic	200			967
23860	electroencephalographs	200			966
23861	electroencephalography	200			966
53335	non-representationally	200			966
16706	counter-intelligences	200			965
16713	counter-revolutionary	200			965
23858	electroencephalograph	200			965
23880	electromyographically	200			965
37841	indistinguishableness	200			965
38585	institutionalisations	200			965
38768	interdenominationally	200			965
46064	magnetohydrodynamical	200			965
53135	non-deterministically	200			965
56624	palaeoanthropological	200			965

Request | Response

Raw | Params | Headers | Hex

```
POST /login.php HTTP/1.1
Host: 83.212.174.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefo
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://83.212.174.87/
```

? < + > Type a search term

Finished

Conclusion: I found no success with the brute force attack.

Directory Browsing

Risk: Medium

Directory Browsing is an information gathering attack which leverages an administrative misconfiguration in a web server which allows listing of directory contents.

This is a very bad practice as it provides a would-be attack far too much information. Most web servers are configured out-of-the box with directory browsing turned on. As a result, this vulnerability is still often found in the wild.

Conclusion: Directory browsing is disabled fro

URL Injection

Risk: High

URL injections take place when an individual attempts to manipulate your online database through the commands sent by the URL.

Often, this form of hacking involves the creation of new pages throughout your website by hackers- often dangerous bits of code or spam links that can make your site a security risk to visitors.

Often, new pages that are created are packed full of code that re-directs your visitors to dangerous locations, or allow your webserver to participate in attacks that you may not even be aware of.

Conclusion: Being a website with little or no parameters, I didn'

Cross-Side Scripting

Risk: Medium

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.

NOTE- **The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS. This is a serious security issue.**

Conclusion: Because of no inputs methods in the web a

Other vulnerabilities:

1) SSH Server CBC Mode Ciphers Enabled

Severity : Low

Description

The SSH server is configured to support Cipher Block Chaining (CBC) encryption. This may allow an attacker to recover the plaintext message from the ciphertext.

Solution

Contact the vendor or consult product documentation to disable CBC mode cipher encryption, and enable CTR or GCM cipher mode encryption.

```
The following client-to-server Cipher Block Chaining (CBC) a
are supported :

  3des-cbc
  aes128-cbc
  aes192-cbc
  aes256-cbc
  blowfish-cbc
  cast128-cbc

The following server-to-client Cipher Block Chaining (CBC) a
are supported :

  3des-cbc
  aes128-cbc
  aes192-cbc
  aes256-cbc
  blowfish-cbc
  cast128-cbc
```

2) *Back-end code disclosure*

Severity : High

Description

Source code disclosure issues occur when the code of the backend environment of a web application is exposed to the public. Source code disclosure enables attackers to understand how the application behaves by simply reading the code and checking for logical flaws, or hardcoded username/password pairs, or API secret keys. The severity here depends on how much of the code is exposed, and how critical the leaked lines of code are for the security of the web application. In short, source code disclosure turns a black box testing process into more of a white box testing approach since attackers get access to the code.

Affected URL : http://83.212.174.87/util.sh

When a wrong password is entered, ideally a login page must only show alerts like 'Wrong Password' or 'Wrong input, Try again' etc. But here in this case, the application shows the entire database variable name and syntax which is an attack surface for any hacker or attacker.

Wrong password!

Executed SQL query: SELECT * FROM form_passwords WHERE
'/dev/random@localhost')=-1, REVERSE(" \'"), ISNULL(NULLIF(N
'/dev/null@localhost')=0, CHAR(40*2-POWER(1, LOG(2)),(4*10+1
MariaDB-0+deb10u1')=0,IF(CONNECTION_ID()=1337,""="","-_-'

util.sh

3) **Default Password Found**

Severity: High

I found that the open port in the server Port: 25 (ssh) uses default password (admin).
This vulnerability can be lethal and any attacker can get root priviliges on the server and do whatever he wants to.

Below is the screenshot:

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to
permitted by applicable law.
root@svr04:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 9d:71:07:11:28:5
          inet addr:83.212.174.87  Bcast:83.212.174.25
          inet6 addr: fe53::25b:8bff:fe21:ea01/64 Scop
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Me
          RX packets:537334 errors:0 dropped:0 overrun
          TX packets:405245 errors:0 dropped:0 overrun
          collisions:0 txqueuelen:1000
          RX bytes:317989090 (318.0 MB)  TX bytes:3567

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:65536  Metric:1
          RX packets:110 errors:0 dropped:0 overruns:0
          TX packets:110 errors:0 dropped:0 overruns:0
          collisions:0 txqueuelen:0
          RX bytes:26943982 (26.9 MB)  TX bytes:269439
root@svr04:~# whoami
root
root@svr04:~#
```

I found there were configuration files inside the system. I could have edited it to make it unusable for the owner. This is a critical flaw.

3) Web Application Potentially Vulnerable to Clickjacking

Severity : Medium

Description
The remote web server does not set an X-Frame-Options response header or a Content-Security-Policy 'frame-ancestors' response header in all content responses. This could potentially expose the site to a

clickjacking or UI redress attack, in which an attacker can trick a user into clicking an area of the vulnerable page that is different than what the user perceives the page to be. This can result in a user performing fraudulent or malicious transactions.

X-Frame-Options has been proposed by Microsoft as a way to mitigate clickjacking attacks and is currently supported by all major browser vendors.

Solution
Return the X-Frame-Options or Content-Security-Policy (with the 'frame-ancestors' directive) HTTP header with the page's response. This prevents the page's content from being rendered by another site when using the frame or iframe HTML tags.

The following pages do not use a clickjacking mitigation response header and contain a clickable event :

- http://83.212.174.87/

FUNCTIONALITY and USABILTY TEST:

Performance/ Load Test:

- Page Size :36.9kb
- Fully Load Time: 688ms

The website took around 36.9 milli seconds to load with 5 requests.

Summary:

Serious flaw found in the system. It needs to be addressed as soon as possible.