WHAT IT TAKES TO

SECURE A WEB APPLICATION

ANANT SHRIVASTAVA

NULL BHOPAL - SEP 2016

ANANT SHRIVASTAVA

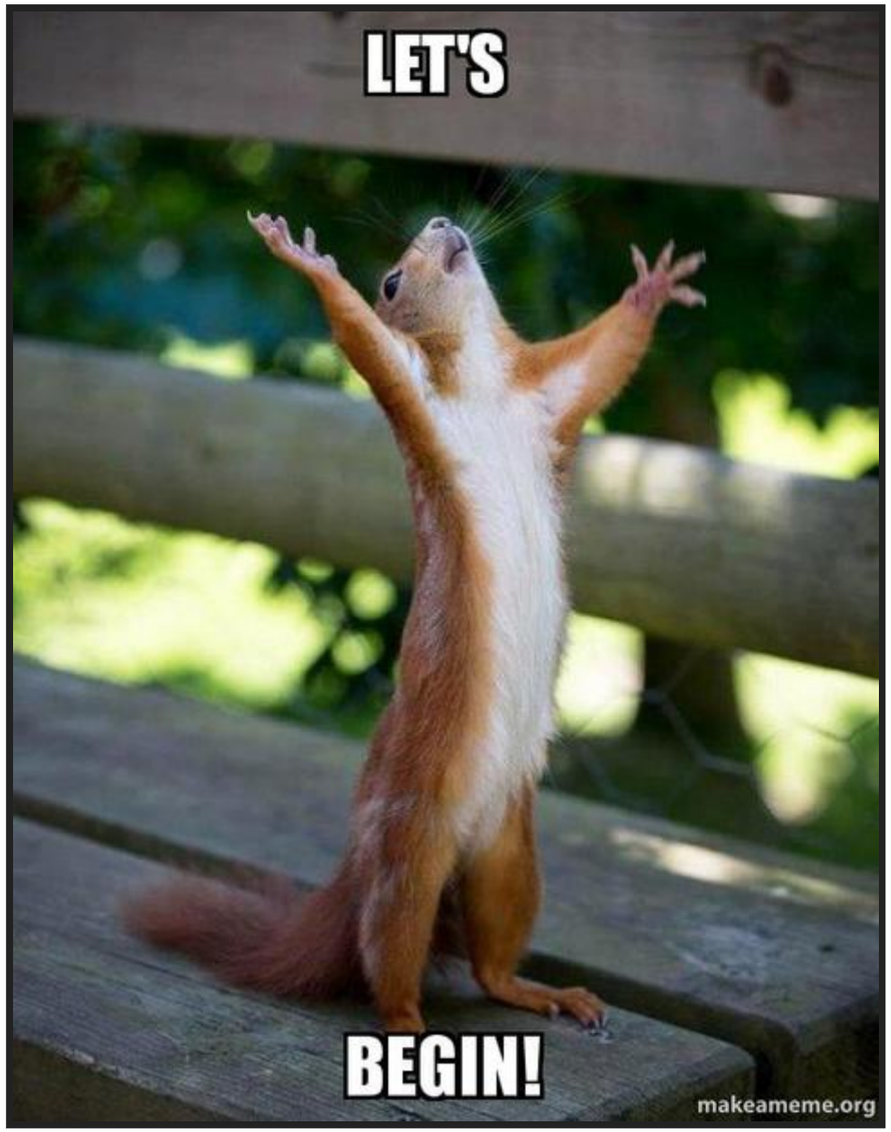
- Information Security Consultant
- Admin - Dev - Security
- null + OWASP + G4H
- http://anantshri.info and @anantshri
- Trainer : Blackhat, RuxCon, NullCon, g0s, c0c0n
- Speaker : Nullcon, c0c0n, ClubHack, RootConf

Android Tamer

Code Vigilant

SO YOU WANT TO HOST A WEB APPLICATION

LETS BEGIN



Ref: https://makeameme.org/meme/lets-begin-ms712r

DEFINING SCOPE

A Financial Organizations Public Blog

- To be kept separate from internal/company network
- Only PR team and one Administrator to have access.
- Should be accessible across the globe (admin interface too)
- Easy to use, WYSIWYG
- Cost effective
- Expected Traffic (1L hits a day)
- Budget not to exceed 20$ a month excluding admin's salary

LETS SELECT THE SOFTWARE

1. Linux OS (coz its "secure" [pun intended])
2. Wordpress (coz its simpler)
3. PHP (coz WP needs it)
4. NGINX (coz its hip-hop)
5. mysql (coz that's what is generally used)

LET BOOK OUR DOMAIN

1. how cheap they are
2. do they offer freebies (discount / free whois protection / free ssl etc)
3. usability

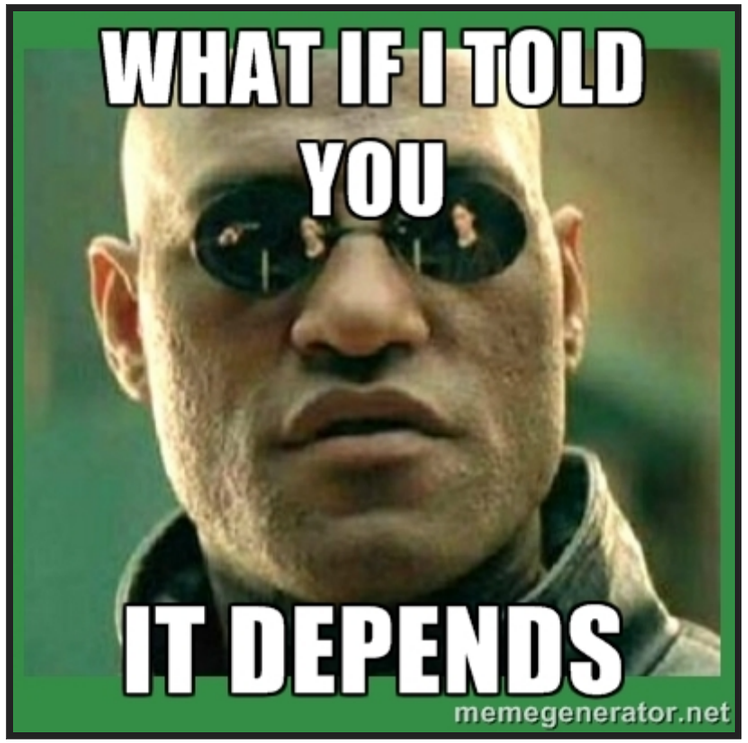
Let me add

1. how quickly does the dns propagates
2. do they prevent domain ownership transfer

example : gandi, namecheap or more

P.S.: More on this later

SO YOU WANT TO SECURE IT



ref:http://meta.security.stackexchange.com/questions/880/the-memes-of-information-security

WHAT ARE YOU PROTECTING

- Data on blog (reputation damage)
- Defacement (reputation damage)
- Financial data, year end review etc (money damage)
- Site downtime (reputation damage)
- Reverse hack on users and back-end users. (leading to potential internal network compromise when they connect to internal network)

WHO IS YOUR ADVERSARY

- Stray hacker / attacker
- Disgruntled employee
- Rival or determined attacker (for hire)
- State Agencies

LETS GET STARTED

PROTECTION AND DEPLOYMENT

Ref: http://www.mememaker.net/meme/well-lets-begin-then/

WHERE WOULD YOU HOST IT

1. Shared hosting (worst in terms of security)
2. VPS
3. Cloud Services (GCloud, Azure, AWS etc)
4. Dedicated server

P.S. 2,3,4 basically means you take care of your own security different models

PAAS, IAAS, SAAS

LETS FINALIZE VPS / CLOUD SERVER

virtual server instance

1. 1 Core
2. 1-2 GB RAM
3. 1 Static IP
4. 1 management console

Cost: 10$ or so.

Source: DO, Linode, AWS and more

BASIC SECURITY HYGIENE

- Don't use pirated software
- Verify OS installers before using them
- Keep OS and components update

OS SECURITY

- What ports need to be opened
  - 21
  - 22
  - 80
  - 443
  - 25
  - what else
- http port 80 should be opened or not.
- ftp 21 should be used or not
- https 443 to be or not to be

OS SECURITY

- Ports to be kept open
  - 80 (seemless redirection to 443)
  - 443
  - 22 (sftp over ssh)
- Services to be configured
  - nginx with 80 marked as auto forward to 443
  - ssh to allow sftp access
  - 443 with ssl
  - 25 exim4 or sendmail for outbound emails not to be exposed publicly

OS SECURITY

- Automatic updates
- Encrypted partitions (Should we use them? what do they prevent)
- why not to use ppa's or why not to use third party channels

WEB SERVER SECURITY

- Port 80 redirect to 443 seamlessly

```
server {
        listen 80;
        listen [::]:80;
        server_name abc.com ;
        location / {
            return 301 https://$host$request_uri;
        }
}
```

- Production config: header update

```
server_tokens off;
autoindex off;
more_set_headers "X-Frame-Options:
SAMEORIGIN";
more_set_headers "X-Content-Type-Options:
nosniff";
more_set_headers "X-XSS-Protection: 1;
mode=block";
more_set_headers "X-Download-Options:
noopen";
more_clear_headers 'Link';
more_clear_headers 'Server';
more_clear_headers 'X-CF-Powered-By';
error_page 403 = 404;
```

WEB LANGUAGE (PHP) SECURITY

- fastcgi mode
- use `GET` or `POST` and not `REQUEST`
- don't rely on `mysql_real_escape_string`

Common basic prod settings

```
allow_url_fopen          = Off
allow_url_include        = Off
allow_webdav_methods     = Off
expose_php               = Off
error_reporting          = E_ALL
display_errors           = Off
display_startup_errors   = Off
log_errors               = On
error_log                = /valid_path/PHP-logs/php_error.log
ignore_repeated_errors   = Off
```

MORE

```
enable_dl                  = On
disable_functions          = system, exec, shell_exec, passthru, phpinfo, show_source, popen,
proc_open
disable_functions          = fopen_with_path, dbmopen, dbase_open, putenv, move_uploaded_file
disable_functions          = chdir, mkdir, rmdir, chmod, rename
disable_functions          = filepro, filepro_rowcount, filepro_retrieve, posix_mkfifo
   # see also: http://ir.php.net/features.safe-mode
disable_classes            =
```

Ref: https://www.owasp.org/index.php/PHP_Configuration_Cheat_Sheet

DATABASE SERVER SECURITY

- Don't use root user for everything
- Don't use blank password for any account
- Separate accounts via there role.
- WP user should not have access to other users db.
- Never expose server port to public internet

WEB APPLICATION SECURITY

- Login mechanism protection (wp-admin)
- Information disclosure protection (username / attachment enumeration)
- Basic Web application level attacks

self-plug https://github.com/anantshri/wp-security *self-plug*

Ref: https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents

HTTPS TRANSPORT SECURITY

- What certificate to be used (Self signed, DV, EV or more)
- What ciphers to be used
- What is Perfect Forward Secrecy and why to use it
- How to validate config

References

- SSLlabs (http://ssllabs.com/)
- Mozilla (https://wiki.mozilla.org/Security/Server_Side_TLS)
- Config generator (https://mozilla.github.io/server-side-tls/ssl-config-generator/)
- SSLScan (https://github.com/rbsec/sslscan)

RECOMMENDATION

- no CBC, RC4
- no 40 or 56 bit cipher. prefer > 128 bit
- SSL Certificate signed by trusted authority
- sha-256 signed cert
- Public key modulus >= 2048 bits
- Prefer PFS (Perfect forward secrecy)

CONFIG

- certs sent to the client in SERVER HELLO are concatenated in ssl_certificate

```
ssl_certificate /path/to/signed_cert_plus_intermediates;
ssl_certificate_key /path/to/private_key;
ssl_session_timeout 1d;
ssl_session_cache shared:SSL:50m;
ssl_session_tickets off;
```

- HSTS (ngx_http_headers_module is required) (15768000 seconds = 6 months)

```
add_header Strict-Transport-Security max-age=15768000;
```

- OCSP Stapling ---fetch OCSP records from URL in ssl_certificate and cache them

```
ssl_stapling on;
ssl_stapling_verify on;
```

MORE

- modern configuration. tweak to your needs.

```
ssl_protocols TLSv1.2;
ssl_ciphers 'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-
POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-
AES128-SHA256';
ssl_prefer_server_ciphers on;
```

- verify chain of trust of OCSP response using Root CA and Intermediate certs

```
ssl_trusted_certificate /path/to/root_CA_cert_plus_intermediates;
resolver ;
```

SSH / REMOTE ADMIN SECURITY

- Key based auth
- no password allowed
- no root login allowed
- no sharing of X clients

Reference

- ssh_scan (https://github.com/mozilla/ssh_scan)
- Guidelines (https://wiki.mozilla.org/Security/Guidelines/OpenSSH)

ADDITIONAL PROTECTIONS

- WAF
- iptables
- fail2ban
- docker

WAF

NAXSI

- Default config is a good starter
- enable in learning mode
- understand if it works for you
- might also be blocking legitimate traffic initially so be careful.
- more powerful alternative ModSecurity

IPTABLES

- Default drop

```
# Setting default filter policy
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

- Default port

```
-A INPUT -p tcp -m multiport --dports 80,443 -j fail2ban-
wordpress
-A INPUT -p tcp -m multiport --dports 22 -j fail2ban-ssh
-A INPUT -p tcp -m multiport --dports 22 -j fail2ban-ssh
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
```

- Fail2ban config

```
-A fail2ban-ssh -j RETURN
-A fail2ban-ssh -j RETURN
-A fail2ban-wordpress -j RETURN
```

MOAR IPTABLES

- minimal DoS mitigation

```
-A INPUT -p tcp -m tcp --tcp-flags RST RST -m limit --limit 2/sec --limit-burst 2 -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -m limit --limit 50/sec --limit-burst 50
-j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -m limit --limit 50/min --limit-burst 200 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -m limit --limit 50/min --limit-burst 200 -j ACCEPT
-A INPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --state NEW -j DROP
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j DROP
-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG FIN,SYN,RST,PSH,ACK,URG -j
DROP
-A INPUT -m conntrack --ctstate INVALID -j DROP
```

FAIL2BAN

```
destemail = EMAIL_ADDRESS

[wordpress]
enabled = true
filter = wordpress
logpath = /var/log/auth.log
maxretry = 3
port = http,https


[ssh]
enabled  = true
port     = ssh
filter   = sshd
logpath  = /var/log/auth.log
maxretry = 3
```

- need a wordpress plugin to provide auth log : such as wp-fail2ban

DOCKER

- new kid in the block
- Allows for fine grained privilege seperation
- runs everything in its own containers.
- Web application will be divided
  - web server "nginx"
  - app server "php"
  - database server "mysql"
- Useful but not necessary

ARE WE DONE YET

NOPE NOT YET

WELL WE COVERED EVERYTHING

DID WE??

WHAT ABOUT

- Server provider access control
- domain registrar account security
- DNS account security
- administrator laptop
- What about backup's
- did we missed anything?

SERVER PROVIDER ACCESS CONTROL

VPS / Cloud or any service provider also need to be in good shape

1. Use 2 factor authentication
2. Google authenticator / freeotp etc
3. Complex password

DOMAIN / DNS REGISTRAR

1. 2 factor authentication
2. do they allow easy transfer
3. do they get attacked by social engineering try calling them up giving them fake details or public details and see if they accept it to reset your account password or give you details.

ADMINISTRATOR LAPTOP

- privilege separation
  - 1 root user
  - 1 admin user
  - 1 non sudo/ admin user for daily operations.
  - switch to admin only when you need to install software
- full disk encryption
- encrypted storage to be used only for work
- password manager

BACKUP PLANS

Think of following:

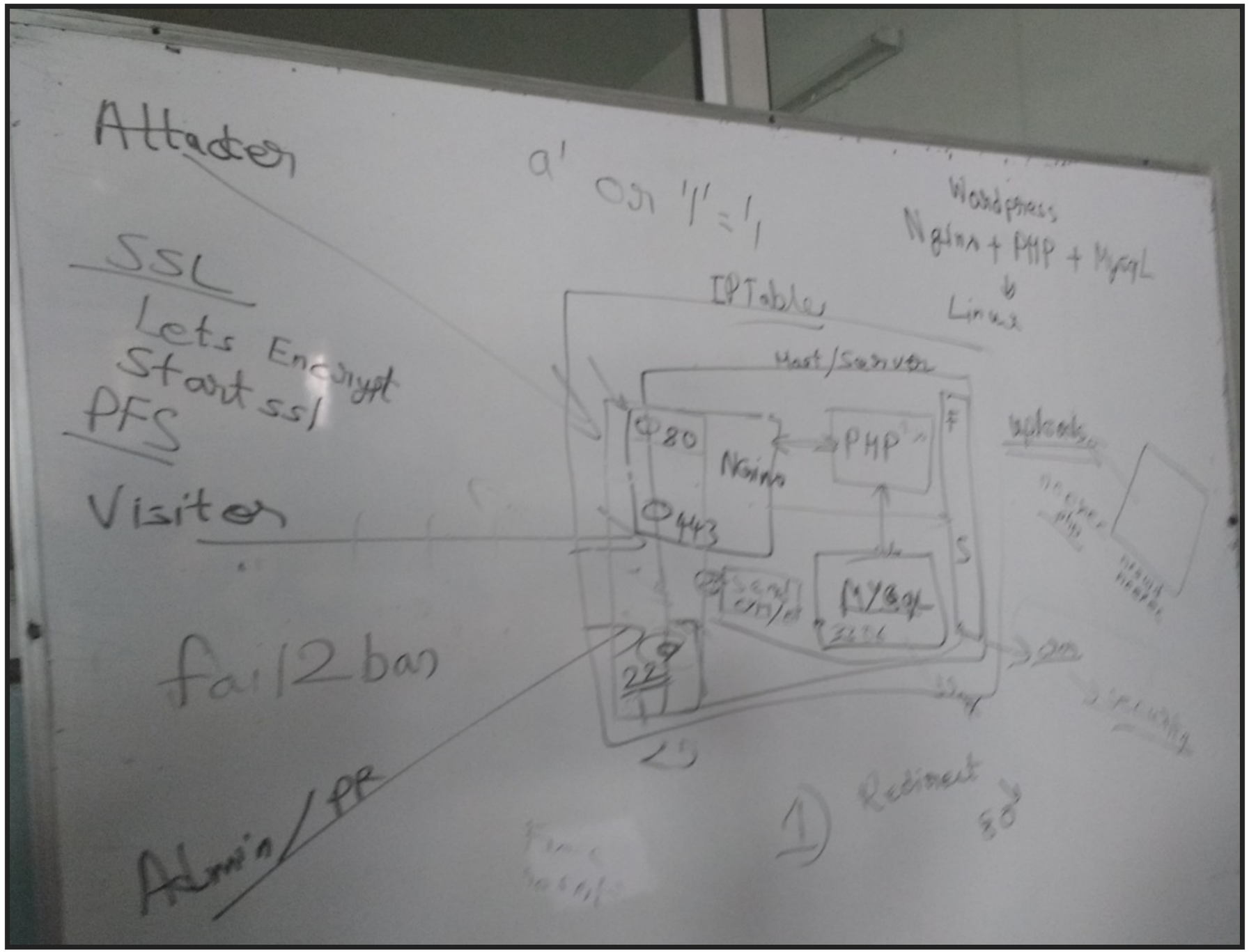
1. What if the server goes down entirely.
2. What if service provider goes down entirely (Linode Dec 2015)
3. What is the backups are only avialable with service provider.
4. what if backup don't restore properly.

BACKUP PLANS

1. multiple *encrypted* backup copies
2. onsite, offsite, cold storage
3. don't just backup try periodically restoring them also
4. Ensure security of these backup accounts are again maintained. Complex passwords, end to end encryption, encrypted storage etc.

MORE QUESTIONS

1. how to keep all these secure / complex passwords
2. how to ensure people dont use weak passwords
3. Should we use something like cloudflare (Hosted DNS + traffic proxy) or not


Attacker
a' or '1'='1
SSL
Lets Encrypt
Start ssl
PFS
Visiter
fail2ban
Admin / PP
Wordpress
Nginx + PHP + MySQL
Linux
IPTables
Host/Server
80
443
Nginx
PHP
MySQL
22
1) Redirect

THATS ALL