ARE YOU A TARGET?

(WEB) SECURITY

FOR NONPROFITS AND EVERYBODY ELSE

SECURITY PRINCIPLES APPLY EVERYWHERE

HOW DOES IT RELATE TO WEB SECURITY?
WHAT QUESTIONS SHOULD I ASK?
WHAT ACTIONS CAN I TAKE?

SECURITY DOMAINS

- Authentication
- Authorization
- Confidentiality
- Integrity
- Availability
- Non-Repudiation

AUTHENTICATION

WHO ARE YOU?

Passwords	Passphrases
Hard to remember	Easy to remember
Easy to guess	Hard to guess
Easy to brute-force	Hard to brute-force
Quite likely to be public	Quite **un**likely to be public
Giv3C@mp2o16	thistablecannotcontainmyAwesomeness!!!

AUTHENTICATION

Who are you?

Two-factor authentication

Pick two from the right

Usually known + item

passphrase + smartphone

AUTHENTICATION

Who are you?

Known information

password

Physical characteristic

fingerprint

Physical item

ID card

Password managers

Great for seldom-used accounts and security questions;

however, single point of failure

pw mgr + 2fa greatly helps

AUTHENTICATION

- Password manager tips
  - Restrict access
  - Notifications
  - Rotate passwords

Authentication is paramount for security in the web, where everyone can be anonymous.

Definitely not limited to the web!

Desktops, office domain/net, devices, software, databases...

AUTHENTICATION

How does it relate to web security?

How does it not?

It's everywhere.

If a service provider knows your password, run.

2FA is already fairly common – request it.

AUTHENTICATION

What questions should I ask?

What is my password?

Do you offer 2FA?

What are the complexity requirements?

Is there a maximum length limit?

With stronger passphrases, your password change schedule can be longer (6 months should do).

AUTHENTICATION

What actions can I take?

Passphrases. Yesterday.

Rotation schedule

Consider pw managers

Apply 2FA wherever possible

DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT

AUTHENTICATION

What actions can I take?

DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT
DO NOT STORE IN PLAINTEXT

AUTHORIZATION

HERE'S WHAT YOU CAN DO.

What do you need to do <action>?

Grant the least possible privileges

Multiple admin levels

Getting to superuser should be annoying!

AUTHORIZATION

Here's what you can do.

If possible, authenticate first

Role-based is most common

Be extremely stingy

Defense-in-depth:

multiple, independent layers

Require authorization between privilege layers in each environment (network, physical, software, services...)

Just because they used to have access, doesn't mean they still do

AUTHORIZATION

Here's what you can do.

Authorization gates

Independent gates

Check every time

Unrelated layers should be kept separate

e.g., does everyone (every role) need access to every shared folder across the entire network?

Major pain point for attackers.

AUTHORIZATION

Here's what you can do.

Compartmentalization

Split | things | up

Web applications nowadays usually have back-ends (databases)

Every action, even logging in, should require appropriate permissions.

AUTHORIZATION

How does it relate to web security?

How does it not?

Should Joe Schmuck have all access?

Remember: wide net.

Security-conscious service providers will have zero trouble answering all of these.

While you're at it, ask about incident response and remediation.

AUTHORIZATION

What questions should I ask?

How granular is access control?

How far could someone get if they break in?

Is my stuff in a shared environment?

If so, how is it protected from a breach in other applications?

Authentication may be done differently in different kinds of systems, but these principles apply everywhere.

AUTHORIZATION

What actions can I take?

Segregate access based on roles (network, applications, shares, etc.).

Grant the least possible privileges.

Defense-in-depth

AVAILABILITY

IF YOU CAN'T USE IT, IT'S USELESS.

If there weren't enough reasons before...

Ransomware.

Also disaster recovery, incident recovery, business continuity, etc...

AVAILABILITY

If you can't use it, it's useless.

Backups...

Do you have them?

Do you test them?

Are they protected?

Denial of service has many forms, many of which apply to web applications.

Besides network flooding, back-end attacks or data tampering may lead to loss of availability.

AVAILABILITY

How does it relate to web security?

Again, back-ends

99.999% uptime is expected

Have concrete plans to:

- prevent down-time
- quickly recover from incidents
- keep your business running during an incident

This applies to networks, devices, applications, databases...

AVAILABILITY

What questions should I ask?

DoS protection?

Can I see the SLA?

Do we have backups?

Are the backups tested?
Are they secure from unauthorized access?

These practices could be considered mature.

Take care of higher-risk items first

(like plaintext password spreadsheets in your open file shares accessible from the Internet)

AVAILABILITY

What actions can I take?

Back your stuff up

Test the backups

Test access to the backups

Simulate incidents

INTEGRITY

TRUST NO ONE.

Create a regular patch schedule – test, then deploy.

Avoid unpatched or old software if possible.

INTEGRITY

Trust no one.

All software is faulty. Patch.

Every patch will break something. Test first.

Public databases exist for automatically exploiting unpatched software – including web apps and frameworks – and even network devices.

Injection attacks against web applications are very prevalent.

INTEGRITY

How does it relate to web security?

Security-conscious service providers should have zero problems answering all of these.

INTEGRITY

What questions should I ask?

How often do you apply patches? Are the patches tested?

How do you protect against...

- SQL injection
- XSS
- CSRF
- Other tampering attacks
- (see OWASP top 10)

Whitelists are the best way to protect against most of the nastiest web application attacks.

Accept only what you expect.

INTEGRITY

What actions can I take?

Patch after testing

For a web app...

- whitelists
- then blacklists

CONFIDENTIALITY

NEED-TO-KNOW.

Most email is not encrypted.

Keep that in mind.

TLS encrypts HTTP traffic. Use it in your internal network as well, where applicable.

VPNs encrypt traffic and allow remote access to the office network.

Be diligent when setting up these things!

CONFIDENTIALITY

Need-to-know.

Email is not confidential!

Secure protocols yet again.

If a service provider can tell you your password, run.

If you can read passwords in your database, people should run from you.

Attackers will create havoc on purpose – web apps should display minimally useful error messages and fail securely.

CONFIDENTIALITY

How does it relate to web security?

Protect sensitive data

Avoid giving out free information

You can google the names of the algorithms and easily learn which ones are good.

CONFIDENTIALITY

What questions should I ask?

What encryption algorithms do you use, and to protect what data?

Is TLS available?

Are passwords hashed and salted? With what algorithm?

Protect your internal network with TLS where applicable.

WiFi: use WPA2 and disable WPS. Separate guest network.

Any failure should be followed by shutting all the appropriate gates.

CONFIDENTIALITY

What actions can I take?

Enable TLS everywhere

letsencrypt.org

VPNs

Remember: email is not confidential

Minimally useful error messages

Fail securely

NON-REPUDIATION

CAUGHT IN THE ACT!

Make it impossible for an attacker to remain undetected.

Enable notifications, alerts, alarms, bells – to a point.

If you start ignoring your alarms, they're useless.

NON-REPUDIATION

Caught in the act!

Notifications

Alerts

Logs

Knowing who (authentication) did what (authorization) when allows you to better defend your assets.

Consider alerts for failed actions.

NON-REPUDIATION

How does it relate to web security?

Quicker defensive reaction

Quicker recovery

Prevent from happening again

When too much noise from logs or alerts is unavoidable, aggregate reports could help.

NON-REPUDIATION

What questions should I ask?

What notification options are available?

What information is logged?

Reports?

Attackers have gone unnoticed for years on networks because nobody was watching...

NON-REPUDIATION

What actions can I take?

Enable notifications for sensitive actions.

Look at your logs (or aggregate reports) now and then.

YOU

YOU ARE THE BEST DEFENCE

People love to help.

Nonprofits love to employ people who particularly love to help.

Attackers know this, so help all you want, just be smart (and a little paranoid) about it.

YOU

Security is everybody's job

Create small, ongoing awareness programs...

- creating good passphrases
- spotting suspicious emails
- ask for details (authenticate) before helping
- password managers and avoiding password re-use

SUMMARY

FOR THE PRACTICAL RISK MANAGER

SUMMARY

- Security awareness
- Passphrases
- 2FA
- No readable passwords stored anywhere (files, databases, etc.)
- Consider password managers (they encrypt your passwords)
- Compartmentalize your network

SUMMARY

- Apply role-based least privilege
- Backups
- Patches
- TLS (internal too if applicable)
- WiFi: WPA2, no WPS, separate guest network
- Enable notifications for sensitive actions

RESOURCES

FINALLY SOME GOODIES

RESOURCES

- "lifehacker best password managers"
- letsencrypt.org
  - free TLS certificates
- pfsense.org
  - free and excellent firewall (also sells appliances)
- untangle.com
  - mostly free next-gen (modular) firewall (also sells appliances)
- freenas.org
  - free storage system with integrity checks (requires some expertise)
- owasp.org
  - for web app developers and testers

BECOME A FORMIDABLE TARGET!