ABUSING CERTIFICATE TRANSPARENCY

OR HOW TO HACK WEB APPLICATIONS BEFORE INSTALLATION.

Hanno Böck
https://hboeck.de/

HTTPS

CERTIFICATE AUTHORITIES (CAS)

CAN WE TRUST CERTIFICATE AUTHORITIES?

NO

Many cases of illegitimate certificates in the past.

IMPROVE OR REPLACE?

Popular Infosec opinion:

CAs are bad, we need to get rid of them.

HOW?

Reality: Nobody has a feasible plan how to replace CAs.

IMPROVING THE CA ECOSYSTEM

BASELINE REQUIREMENTS

HTTP PUBLIC KEY PINNING (HPKP)

CERTIFICATE AUTHORITY AUTHORIZATION (CAA)

CERTIFICATE TRANSPARENCY (CT)

PUBLIC LOGS

Let's put all certificates into public logs that everyone can read.

CT DETAILS

Merkle Hash Trees, Signed Certificate Timestamps (SCT), Signed Tree Head (STH), Precertificates, Monitors, Gossip, ...

It's complicated, but not relevant for this talk.

CERTIFICATE LOGGING

In the future logging will be required (April 2018).

CT TODAY

Most certificates already get logged.

WATCHING THE CAS

Certificate Transparency means everyone can check logs for suspicious activity.

https://crt.sh

```
Issuer:
    commonName                  = DigiCert SHA2 High Assurance Server CA
    organizationalUnitName      = www.digicert.com
    organizationName            = DigiCert Inc
    countryName                 = US
Validity
    Not Before: Oct  6 00:00:00 2014 GMT
    Not After : Dec 13 12:00:00 2017 GMT
Subject:
    commonName                  = *.adoftheyear.com
    organizationalUnitName      = IT
    organizationName            = Metrixlab B.V.
    localityName                = Rotterdam
    stateOrProvinceName         = California
    countryName                 = NL
```

CERTIFICATE TRANSPARENCY IS A DATA SOURCE

For researchers.

For search engines.

For attackers?

FEED OF NEW HOST NAMES

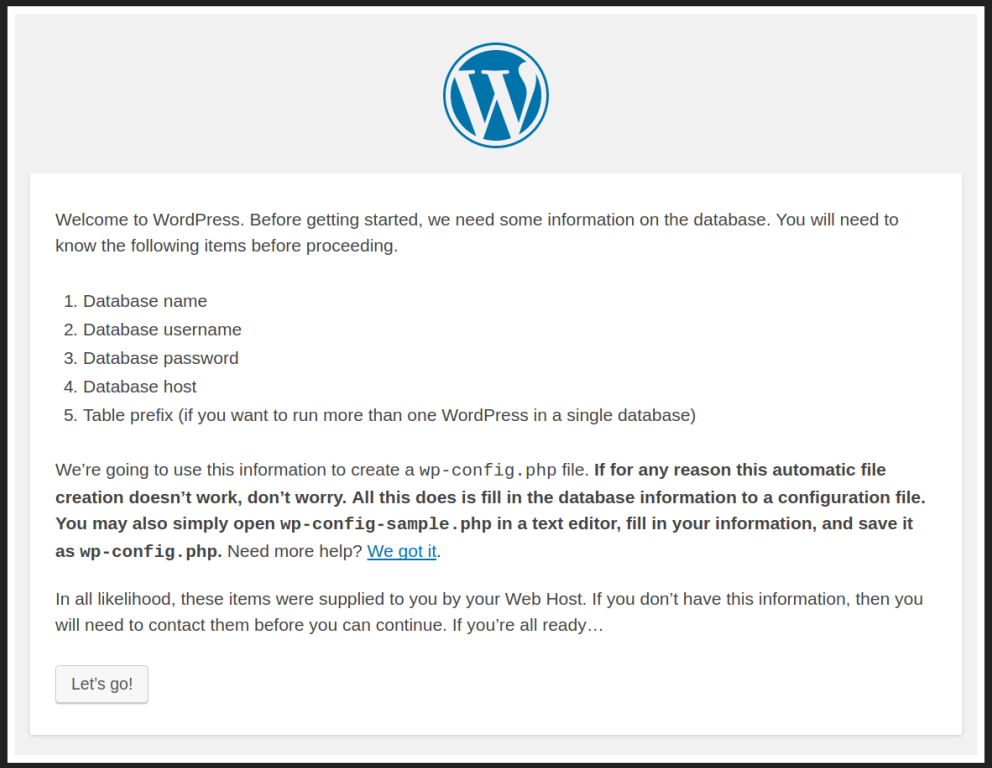
Certificates contain hostnames.

In other words: Certificate Transparency provides a feed of newly created HTTPS host names.

SELF-HOSTED WEB APPLICATIONS

Wordpress, Joomla, Drupal etc.

WEB APPLICATION INSTALLERS



Welcome to WordPress. Before getting started, we need some information on the database. You will need to know the following items before proceeding.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

We're going to use this information to create a `wp-config.php` file. **If for any reason this automatic file creation doesn't work, don't worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.** Need more help? We got it.

In all likelihood, these items were supplied to you by your Web Host. If you don't have this information, then you will need to contact them before you can continue. If you're all ready…

Let's go!

INSTALLERS

Upload files to hoster, open in browser.

Installer asks for some settings (initial user account, database credentials, ...).

INSTALLER (IN)SECURITY

Usually installing needs no authentication!

GOOGLE DORKING WEB INSTALLERS

Old idea: Use Google to find unprotected installers.

ATTACK IDEA

During installation there is a time window between uploading files and completing the installer without any protection.

Remember: We have a stream of newly created host names.

HTTPS AND CERTIFICATES

HTTPS is becoming more popular and many hosters automatically issue certificates.

ATTACK (1)

Attacker monitors CT logs, extracts host names.

Compares web pages with common installers.

ATTACK (2)

Insaller found: Install the application.

Upload a plugin with code execution backdoor.

Revert installation, leave backdoor.

DATABASE CREDENTIALS

Installers often require Database credentials for MySQL.

But the database server can be external.

https://www.freemysqlhosting.net/

DEMO

CHALLENGES

Logged certificates aren't immediately public. Lag around 30 minutes - 1 hour.

Still: You'll hit plenty of vulnerable installers.

OPTIMIZATIONS

Instead of checking sites once one could check them multiple times.

NUMBERS

I could've taken over around 4000 Wordpress installations within a month.

Much lower numbers for other apps, Joomla (~400) and Owncloud/Nextcloud (each ~100) may still be worth attacking.

PROTECTION

INSTALLERS NEED AUTHENTICATION

CHALLENGE

Application programmers want easy installations.

SECURITY TOKENS

Webapp creates token, writes it to file.

User has to read token to perform installation.

VENDORS

Drupal

Typo3

Owncloud

... no reaction at all.

VENDORS

Wordpress, Nextcloud, Serendipity participated in cross-vendor discussion, but no action.

JOOMLA

Security token if database server is not localhost.

WHITELISTING LOCALHOST

This was my idea, but I don't like it.

Attacker could already own database credentials for localhost.

WHAT CAN USERS DO?

BE FAST

However there's no way of knowing how fast CT logs are.

Future CT logs could be faster.

.HTACCESS

User can create .htaccess with password protection before installation.

Some webapps create .htaccess themselve.

TAKEAWAY (1)

Unauthenticated installers need to go away.

Majority of web application developers ignored the issue so far.

TAKEAWAY (2)

Certificate Transparency is a powerful tool to strengthen the HTTPS/TLS and PKI ecosystem.

As a side effect it provides an interesting data source.

Attackers have that data source, too.

TAKEAWAY (3)

Certificate Transparency means TLS host names are no longer secret.

People need to be aware of that.

THANKS FOR LISTENING!

Questions?

https://hboeck.de/