

Common Security Issues

國立陽明交通大學資工系資訊中心

Computer Center of Department of Computer Science, NYCU

Security Principles

- Network Security is a very very big issue, can not full covered in this course
- CSO: Chief Security Officer
 - Required position for an IT-related public offering company in Taiwan
- Security is time-consuming
 - One Time Password
 - Forced splitting internet and intranet
 - EC sites in Taiwan
 - Long password
- KISS: Keep it simple and stupid

Network Security Threats

- Virus / Trojans
- Adware / Spam
- Phishing / Fraud
- Hijacking
- Social Engineering / APT
- Denial of Service Attacks
-

Example

- [駭客挾持BGP，竄改加密錢包DNS，盜走價值17萬美元的以太幣 | iThome](#)
- [研究：中國電信長期挾持經過美國與加拿大的流量 | iThome](#)
- [微軟緊急修補 Exchange Server 漏洞，背後功臣是來自台灣資安團隊](#)
 - An exploit not found for 11 years (since Exchange 2010)

The ideas

- Multiple layers of protection
 - DMZ
 - Splitting permission of users
 - Principle of least privilege
 - Protected Network (e.g. VPN)
- Intrusion Detection
 - Firewalls
 - WAF
- Auditing
 - Logging

OWASP

- The Open Web Application Security Project
- <https://owasp.org/>

OWASP Top 10 Security Risks

- Published every 3 years, the latest is 2017 version (the 2020 version is on-going, may be updated in the near future)
 - <https://owasp.org/www-project-top-ten/>
 1. Injection
 2. Broken Authentication
 3. Sensitive Data Exposure
 4. XML External Entities (XXE)
 5. Broken Access Control
 6. Security Misconfiguration
 7. Cross-Site Scripting (XSS)
 8. Insecure Deserialization
 9. Using Components with Known Vulnerabilities
 10. Insufficient Logging & Monitoring

Injection

- Improper escaping of input
- Cause data leak or data loss
- Examples
 - SQL injection
 - Javascript injection
 - XSS (Cross-site Scripting)
- Question: How to avoid it?

CVE: Common Vulnerabilities and Exposures

- CVE® is a list of entries for publicly known cybersecurity vulnerabilities.
- Subscribe the RSS feed
- <https://cve.mitre.org/>

SSL Server Test

- Test the SSL configuration and potential risks
- <https://www.ssllabs.com/ssltest/>

The screenshot displays the Qualys SSL Labs report for the domain **www.cs.nycu.edu.tw**. The overall rating is **A+**. The report includes a navigation menu, a breadcrumb trail, and a summary section with a bar chart showing scores for Certificate, Protocol Support, Key Exchange, and Cipher Strength. Below the chart are three informational banners: a yellow one about documentation, a light blue one about SNI support, and a green one about HSTS.

Qualys. SSL Labs Home Projects Qualys Free Trial Contact

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > [www.cs.nycu.edu.tw](#)

SSL Report: www.cs.nycu.edu.tw (140.113.235.48)

Assessed on: Sun, 14 Mar 2021 19:10:46 UTC | [Hide](#) | [Clear cache](#) [Scan Another »](#)

Summary

Overall Rating

A+

Category	Score
Certificate	100
Protocol Support	100
Key Exchange	90
Cipher Strength	90

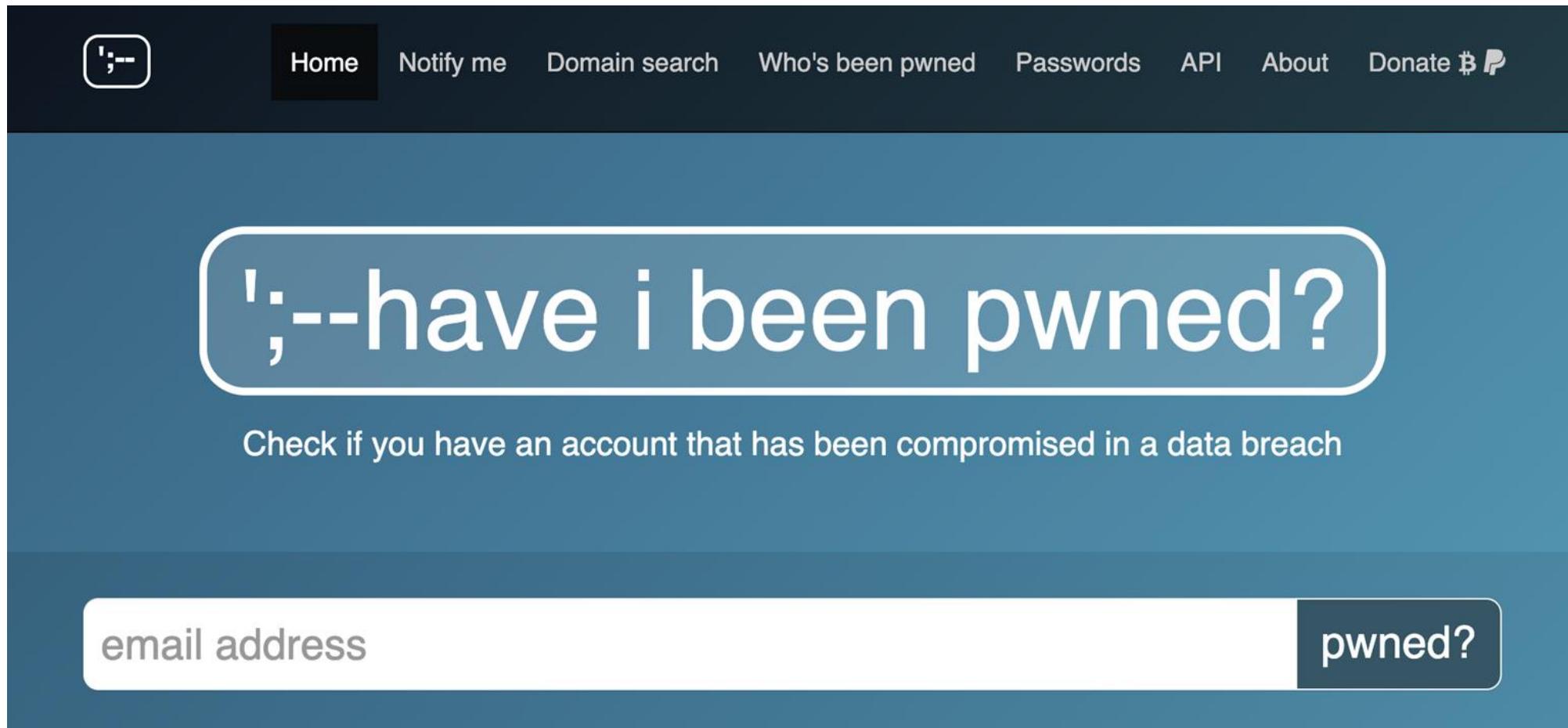
Visit our [documentation page](#) for more information, configuration guides, and books. Known issues are documented [here](#).

This site works only in browsers with SNI support.

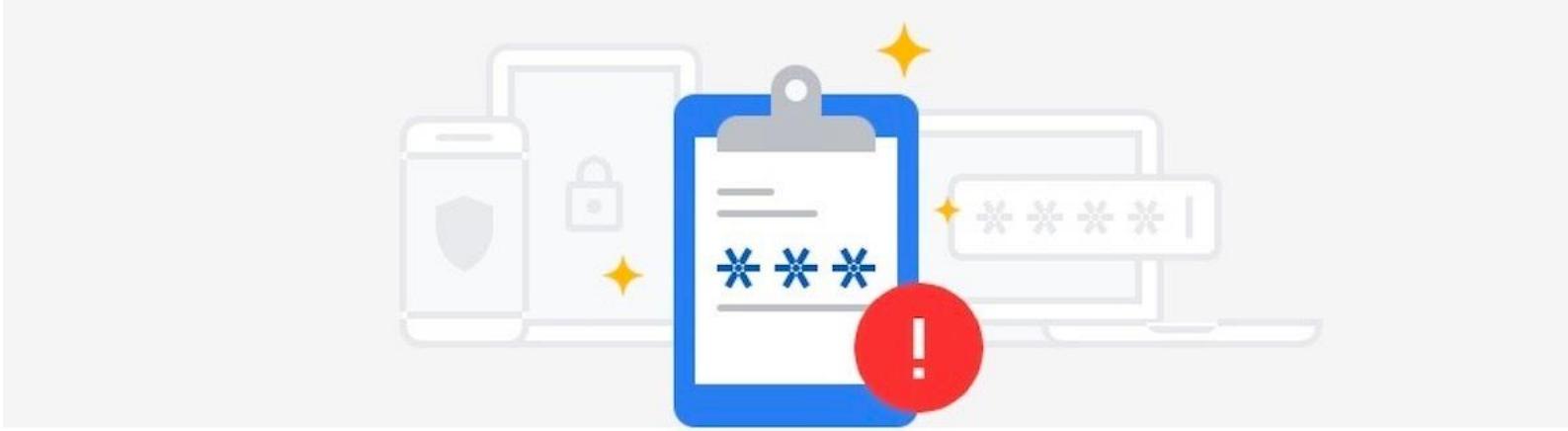
HTTP Strict Transport Security (HSTS) with long duration deployed on this server. [MORE INFO »](#)

Have I been pwned?

- Test if your email & password leaked
- <https://haveibeenpwned.com/>



The screenshot shows the homepage of the website 'Have I Been Pwned?'. The navigation bar at the top includes links for Home, Notify me, Domain search, Who's been pwned, Passwords, API, About, and Donate. The main content area features a large white rounded rectangle containing the text 'have i been pwned?' in a white, lowercase, sans-serif font. Below this, a smaller line of text reads 'Check if you have an account that has been compromised in a data breach'. At the bottom, there is a white input field with the placeholder text 'email address' and a dark blue button labeled 'pwned?'.



Check your passwords

A data breach on a site or app exposed your password. Chrome recommends checking your saved passwords now.



Close

Check Passwords

Static analysis

- A testing methodology that analyzes source code to find security vulnerabilities
- Human review needed
 - Can not find authentication problems, access control issues, insecure use of cryptography
 - Large number of false positives
 - Difficult to “prove” that an identified security issue is an actual vulnerability.
- https://owasp.org/www-community/Source_Code_Analysis_Tools/

Dependencies

Dependents

⚠ We found potential security vulnerabilities in your dependencies.

[Dismiss](#)

Some of the dependencies defined in your `package.json` have known security vulnerabilities and should be updated.

Only the owner of this repository can see this message.

[Learn more about vulnerability alerts](#)

These dependencies have been defined in `sarkartanzil.github.io`'s manifest files, such as `package.json`

 Dependencies defined in `package.json` **11**

>  [jquery / jquery](#)

⚠ Known security vulnerability in `^ 1.11.3`

>  [twbs / bootstrap](#)

`^ 3.3.7`

>  [BrowserSync / browser-sync](#)

`^ 2.13.0`

>  [FortAwesome / Font-Awesome](#) font-awesome

`^ 4.6.3`

>  [gulpjs / gulp](#)

`^ 3.9.1`

>  [scniro / gulp-clean-css](#)

`^ 2.0.10`

DoS: Denial of Service Attacks

- Make a network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet
- Methods
 - Software bugs
 - Bad designed / implemented
 - protocol
 - SYN Flooding
 - Amplification Attacks
- Exhaust all resources

UDP-based Amplification Attacks

Protocol	Bandwidth Amplification Factor
Memcached	50000 (fixed in version 1.5.6) ^[61]
NTP	556.9 (fixed in version 4.2.7p26) ^[62]
CharGen	358.8
DNS	up to 179 ^[63]
QOTD	140.3
Quake Network Protocol	63.9 (fixed in version 71)
BitTorrent	4.0 – 54.3 ^[64] (fixed in libuTP since 2015)
CoAP	10 – 50
ARMS	33.5
SSDP	30.8
Kad	16.3
SNMPv2	6.3
Steam Protocol	5.5
NetBIOS	3.8

DDoS: Distributed Denial of Service (1)

- The incoming traffic flooding the victim originates from many different sources
 - Maybe the sources have normal behavior
- Difficult to block all attacking source



DDoS (2)

- Botnet
 - Comprised computers with malicious software (zombies)
 - Launch attacks if the botnet owner gets paid
- Clean Pipe
 - Multiple layers of filter of DDoS traffic
 - Lots of IDSs & IPSs
 - “Wash” traffic and direct clean traffic to servers
 - False positive

Conclusion

- Safe design lowers the security risk
 - Multiple layers of protection
 - Encrypt matters
- Logging and auditing
- Keep systems updated
 - Vulnerabilities feed
 - Human review