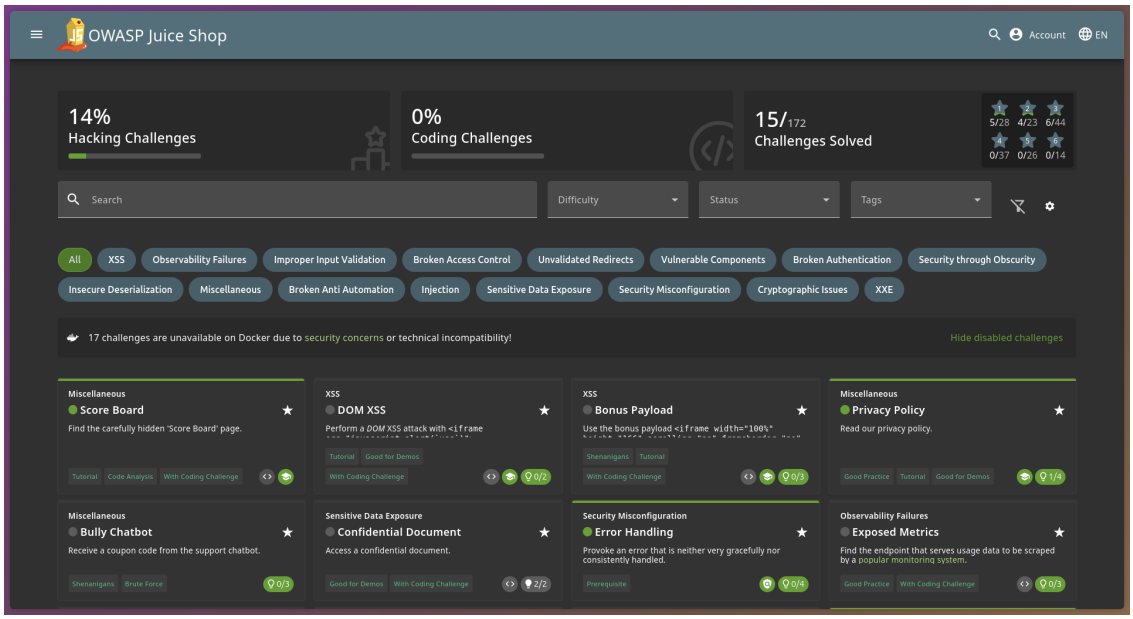
Network Administration/System Administration Homework #11

B10202012 劉仲楷

1 三角準則的侵略者!?

1.
   - 成績查詢系統在出成績當日容易因大量連線，而無法正常使用，違反 Availability。
   - Mixpanel、OpenAI 資安時間流出使用者資訊，違反 Confidentiality。
2. Assumption：筆電功能正常。Attacker 沒有太 powerful 的 resources。
   - Threat model: Attacker 在 victim 離開電腦但沒登出時繞過登入認證；Countermeasure: 離開電腦即手動登出、不要讓電腦離開自己。
   - Threat model: Attacker 猜到密碼，通過登入認證；Countermeasure: 用困難的密碼、不要被 shouder surfing、不要在鍵盤上留下痕跡、改用其他認證不要用密碼。
3. Assumption：手機、網路通訊功能正常，手機主人不主動允許他人，用他的手機使用此服務。
   - Threat model: 手機可能被偷走或掉了；Countermeasure: 手機上鎖且不在未解鎖的狀態下用此服務。Sim 卡轉移到另一隻手機要輸入密碼才能使用。
4.

2 果汁店也有洞！

1. 

(a) DOM XSS

(b) Confidential Document

(c) Login MC SafeSearch

(d) Five-Star Feedback:
登入頁面輸入 User: `admin' OR 1=1 --`、Password: any。Login admin 後，到 `http://localhost:3000/#/administration` 刪掉五星評價。

(e) View Basket

(f) Password Strength

(g) Meta Geo Stalking

(h) Missing Encoding

(i) Repetitive Registration:
註冊帳號，得到 api request，更改 password repeat field，其他不更動再送一次即可。

(j) Exposed Credentials

2.

3 R-SA！破密部

Ref: CNS 2024 slides 這題 code 詳細見 `code/rsa/solve.py`

1. flag: `NASA_HW11{blind_signing_is_dangerous}`
利用 RSA 的 Malleability，我們可以簽 x, y s.t.

$$\begin{aligned} c_1 &= x^e && \mod n \\ c_1 &= x^e && \mod n \\ \Rightarrow c_1 c_2 &= (xy)^e && \mod n \end{aligned}$$

因此做法是將要簽的明文 `name=soyo` 轉換成數字後，拿到 factordb 得到質因數分解得到 c_1, c_2 給 soyo 簽完後，再相乘轉回 bytes 傳給 anon。

2. flag: `NASA_HW11{W0w_y0u_kNow_h@st@d'5_bro4dc@s7_47t@cK}`
利用 Hastad's broadcast attack，也就是找到 e 組 (c_i, n_i) s.t.

$$c_i \leftarrow m^e \mod n_i$$

Let $N = \prod n_i$ 找到 $C \leftarrow m^3 \mod N$ 由於 $m^e < N$ 所以 $m = C^{1/e}$ 至於怎麼找到 C 就需要用到 CRT（中國剩餘定理） Let $b_i = N/n_i$

$$C = \sum c_i b_i b_i^{-1}$$

所以我們只要連線 e 次，得到 e 組密文和公鑰，就能恢復明文。

4 TESTING in the FUZZ

1.
   - 測資來源方式不同：Mutation-based fuzzing 從現有的 seed input 出發，透過 bit-flip、byte-swap等方式突變產生新測資。Generation-based fuzzing 則根據 spec 或 grammar 從零生成結構化的測資。
   - 單次測資「有效性」與覆蓋率不同：Mutation-based 常產生大量無效或無意義輸入，容易卡在 parser 前段。Generation-based 產生的輸入通常更有結構，較容易通過 parsing，達到 deeper coverage。
   - 對輸入格式的知識需求不同：Mutation-based 不需要了解輸入格式，對黑箱測試非常方便。Generation-based 需要提供格式或語法規則，設定成本較高。

2. (a) fuzzer 從使用者提供的 seed 開始測試。
   (b) fuzzer 對 seed 做大量突變
   (c) 將新測資餵給目標程式，收集 coverage feedback（可透過 coverage 篩選 seed）。
   (d) 如果出發 crash 或 timeout，則成功發現潛在漏洞

3.

4.

5 敗北協定太多了！

ref: Fortinent: SPF、Fortinent: DKIM、Fortinent: DMARC、Wiki: downgrade attack

5.1 DNS Security

1. Attacker 做 (victim 的) IP spoofing，利用開放式的 DNS Resolver 發送一個小 Query（例如 ANY、TXT）。DNS response 封包通常比 Query 大非常多。DNS Server 就會將巨大 responce 送到 victim's IP，造成流量放大與 DDoS。
   防禦方法：
   - 關閉 open resolver：DNS 伺服器只允許內部網段做遞迴查詢。
   - 啓用 Response Rate Limiting (RRL)：限制相同來源/查詢的回應速度。

2. DNS Cache Poisoning 是攻擊者向 DNS resolver 送出 query 的同時，大量回傳偽造的 DNS response，希望在合法 response 前成功猜中交易識別碼與 UDP Port，使 DNS Cache 存入偽造的 domain, IP mapping。使用者之後 query 便會被導向假網站。防禦方法：
   - DNSSEC：加入數位簽章確保 DNS 回應不可偽造。
   - 縮短 Cache TTL、限制可疑回應

5.2 SMTP Security

3. SPF（Sender Policy Framework）原理是：網域（e.g. example.com）在 DNS TXT 記錄中宣告：「哪些 IP 位址有權替此網域寄信」。SMTP 收信端在收到 mail 時，會根據 MAIL FROM 的 domain 去查詢 SPF TXT，若發信 IP 不在白名單內，就視為 spoof。防止 spoofing 的方式：這樣攻擊者就無法假裝從合法 IP 代替該 domain 發送郵件，因此降低偽造寄件人網域的可能性。

4. DKIM（DomainKeys Identified Mail）原理：發信端使用一把 domain 專屬的 private key 對郵件內容做簽章；收信端根據公布的 public key 驗證簽章正確性。防止 spoofing 的方式：攻擊者無法偽造 DKIM 簽章。

5. 不能，SPF、DKIM、DMARC 只能保證網域的正確性，但 hint 這篇 paper 發現：10 個主流 email 供應商與 19 種郵件客戶端，都至少對某些攻擊類型存在弱點。舉 Intra-server攻擊 為例：
即使 SPF/DKIM/DMARC 設定正確，有些 email server 本身其內部模組對同一封信所採用的「哪個 header／domain 做為驗證依據」不一致，也可能錯判通過驗證，但實際從別的 domain 發來。

5.3 TLS Security

6. TLS Certificate 通常包含：

   - Subject：伺服器的 domain name（CN 或 SAN）
   - Public Key
   - 有效期限
   - 使用的演算法
   - Issuer（簽發該憑證的 CA）
   - CA 的數位簽章（chain of trust）

   CA 是被信任的第三方機構，負責驗證網站身份、簽發 Cert。瀏覽器與 OS 內建 CA 的 root cert，透過這些 root cert 可信任由 CA 簽發的網站憑證。也就是藉由 hierachy 的結構 reduce 各網站 cert 到信任 root CA，再 reduce 到相信瀏覽器/OS。

7. HTTPS Downgrade Attack：攻擊者位於 MITM 位置，攔截使用者對 https://example.com 的請求，强迫將連線降級成 http://，或更低有漏洞的版本，而攻擊者就能利用該漏洞攻擊/竊聽。可以利用啓用 HSTS（HTTP Strict-Transport-Security），强制使用 HTTPS。或將 Domain 加入 HSTS Preload List，讓瀏覽器在第一次連線前就知道此站永遠必須 HTTPS。

6 貓物語（赤）

這題 codes 詳細見 `code/more-ctf/*.py`

1. flag: `NASA_HW11{pseudorandomness_does_not_guarantee_unpredictability}`
   這題利用解 linear congruential system 即可獲得 a, c，由於 state 就是輸出 random number 本身，所以獲得 state 後，即可復現 rng。即可猜到下一個數字。 解 lcs 方式如下：

$$a = (s_2 - s_1) * (s_1 - s_0)^{-1} \mod m$$
$$c = (s_1 - a * s_0) \mod m$$

   其中 s_0, s_1 是連續兩個 states。 詳細 code 見 `game.py`。

2. flag: `NASA_HW11{07p_k3y_mu57_b3_47_l3457_45_l0n6_45_pl41n73x7}`
   由於我們知道明文裏面有 `NASA_HW11{` 剛好 10 個 bytes 符合 keylen，所以我們可以暴力找每個 offset 當作 `NASA_HW11{` 的 key，再利用 key 還原剩下明文。複雜度會是 $O(\text{len}(cip))$ 詳細見 `otp.py`。

3. flag: `NASA_HW11{https://youtu.be/1GxwDuV5JMc}`
   這題我們可以先預先算好 $0 \sim 2^{24}$ 的 md5 hash 結果做成字典，被 challenge 的時候直接回傳預算好的就可以了。詳見 `pow.py`。

4. flag: `NASA_HW11{y0u_KN0w_r3F13C710n_4774cK}`
   這題我們需要同時和 prover 及 verifier 各開一個連線。我們首先問 verifier nonce，再把 nonce 回傳給 prover，接把 prover 給得 prove 回傳給 verifier，就能拿到 flag 了。詳見 `refl.py`。