Homework #11

Due Time: 2025/12/02 (Tue.) 23:59

Contact TAs: `vegetable@csie.ntu.edu.tw / nasa@csie.ntu.edu.tw`

Instructions and Announcements

- **NO LATE SUBMISSION OR PLAGIARISM IS ALLOWED.**
- Discussions with others are encouraged. However, you should write down your solutions **in your own words**. In addition, for **each and every** problem you have to specify the references (the URL of the web page you consulted or the people you discussed with) on the first page of your solution to that problem.
- Some problems below may not have standard solutions. We will give you the points if your answer is followed by reasonable explanations.

Submission

- Please place your answers in the same order as the problem sheet and do not repeat problem descriptions, just organize them by problem number in a tidy manner.
- Please zip all the files, including one pdf file (your report), the `code/` folder, and the `poc/` folder. Name the zip file "`{your_student_id}.zip`", and submit it via NTU COOL. The directory layout should be the same as listed below:

```
{your_student_id}/
+-- {your_student_id}.pdf
+-- code/
+---- {security scripts}
+---- ...
+-- poc/
+---- {POCs for P4}
+---- ...
```

Grading

- The raw total score for the correctness and completeness of your answer is 175 points. However, the total score will be capped at 100 points if it exceeds 100.
- It's possible you don't get full credits even if you have the correct answer. You should show how you get the answers step by step and list the references.
- Tidiness score: 3 bonus points, graded by TA.
- Final score = $\min(100, \text{correctness score}) + \text{tidiness score}$.

Security Part I

0. More Instructions

- 請不要用任何形式干擾其他人作答，或不是以解題為目的來攻擊本作業的各項設施。經舉發查證屬實者，將會受到非常嚴厲的懲罰。
- 請先下載作業要用到的檔案。
- 所有題目分數加總是 175 分，但超過 100 分會以 100 分計。你可以斟酌不作答某些題目。
- 標有 (CTF) 的題目都會有 flag，flag 的格式為 `NASA_HW11{[0-9A-Za-z_':/@]+}`。請務必在你的作答中明寫出你所找到的 flag。
- 可以使用生成式 AI (包含 ChatGPT 及其他類似工具) 來分析 CTF 題的程式碼與釐清問答題相關概念，但嚴禁使用生成式 AI 完整作答題目。完全用生成式 AI 回答的助教會斟酌扣分。
- 如果你有寫了 script 或程式來進行解題，請在作業的 zip 中附上檔案，放在 `code/` 資料夾底下，並在 report 中提及之。
- 對於動手操作的題目 (包括但不限於標有 (CTF) 的題目)，你都需要在 report 寫一份 write-up ，即詳細說明自己是如何做到的。請說服批改助教「你是真的有自己想過」還有「你是真的懂」。
- 即便沒解出來也請儘量作答，可以寫下錯誤的嘗試或是網路上搜尋到的資料。批改的助教將會依照方向與距離答案多遠來給予部份分數。
- 只要不是抄襲或作弊，非常歡迎你嘗試非預期解。

1. 三角準則的侵略者!? (16 points)

課堂上有提到 CIA 一般用來當作資訊安全的準則，其中 C, I, A 三個字母分別為 Confidentiality, Integrity 和 Availability，其實也就是一個「正常的服務」所應具備的要素。

(a) (4 points) 請舉出兩個現實生活中的資安事件，說明其違反 CIA 的哪幾項，並說明原因。

為了達成 CIA，我們會透過 threat modeling 來搞清楚我們可能會面對的攻擊手法，並針對攻擊做出相應的防禦。以下的題目會提出許多不同的系統 (system) 與安全需求 (security requirement)。你需要提出**不超過 4 個**合理的假設 (assumption) 與 **2 種**不同的 threat model，每種 threat model 都需要提供 **1 個**應對措施。不同題目間的 threat model 不能太相似，否則批改者會認定你是偷懶而斟酌扣分。

例題

- System: 系上網路列印服務
- Security requirement: 同學們可以使用網路列印功能，在送出請求的三分鐘之內取得列印完成的印刷品

參考解答

- Assumption:
  1. 電子設備的電子元件皆狀態良好
- Threat model:

Threat Model	Countermeasure
有人嘗試利用網路列印頁面的網頁漏洞來攻擊服務	定期將 server 更新至最新版本
有人透過大量列印來耗盡印表機的資源（紙張或碳粉匣）	在資源剩餘量低落時，限制每個人的使用量，並通知管理員補充列印資源

題目

(b) (4 points)

- System: 個人筆電
- Security requirement: 沒有被擁有者允許的人不能使用

(c) (4 points)

- System: 簡訊實聯制
- Security requirement: 任何人皆以自己的真實身份進行實聯制掃描並傳送簡訊

(d) (4 points)

- System: Nasa 線上期末考
- Security requirement: 考試期間，各組不得以任何方式與非同組的人類進行交流

2. 果汁店也有洞！(26 points)

`OWASP Juice Shop`: https://github.com/juice-shop/juice-shop

`OWASP Juice Shop` 是一個相當**不安全**的網頁服務，一般用於資安相關的訓練或競賽。其中包含了 OWASP Top Ten 以及其他現實生活中的資安漏洞。本題希望同學們能夠透過 OWASP Juice Shop 上的題目來學習到 web security 相關的知識。

請參考上方連結自行架設一個 `OWASP Juice Shop` 伺服器 (建議用 Docker 架設以便保存進度)，並完成以下要求。你可以在 `/#/score-board` 找到 Scoreboard，而在 Scoreboard 中可以找到題目、要解開題目需要做哪些操作以及提示。

(a) (20 points) 請在以下 10 個 `OWASP Juice Shop` 的 hacking challenge 中挑選 5 題作答，並附上做完 5 題後的 Scoreboard 截圖 (不需要做 coding challenge)。對於各題，請同學們附上解題過程，並說明題目介紹的漏洞類型以及原理，並給出解決漏洞的方法。
Hint: Scoreboard 提供的 Hint 很有用

(i) DOM XSS
(ii) Confidential Document
(iii) Login MC SafeSearch
(iv) Five-Star Feedback
(v) View Basket
(vi) Password Strength
(vii) Meta Geo Stalking
(viii) Missing Encoding
(ix) Repetitive Registration
(x) Exposed Credentials

(b) (6 points) 簡單介紹 SSRF、CSRF、XSS 的原理，並比較三者之間的差異。

3. R-SA！破密部 (30 points)

Act II: 為什麼要破解 RSA

數個月前，在做 NASA HW0 的你收到了當紅樂團主唱——三角初華——的祕密委託：有不肖分子告訴人稱「東京阿農」的千早愛音關於 RSA 的錯誤資訊，導致她用了糟糕的 RSA public key 來加密她的祕密日記，讓攻擊者有機可乘！「主謀身份的線索就藏在她的祕密日記中。請盡快解開她的日記，並把它傳給我，由我來解開主謀的身份。」初華如此寫道。然而，你卻發現愛音正陷入低潮期，只願意與室友爽世進行對話。

身為社牛的你透過進一步交流，得知她們利用最基本的 RSA signature 來驗證彼此的身份，以確保外人 (包含你) 無法假冒她們任一人去跟對方聊天，以竊取只有她們兩位知道的祕密。同時，對於只有自己能知道的隱密資訊，她們會各自用最基本的 RSA encryption 來加密它們，確保只有自己能解開被加密的內容。在本題中，你將試著假冒爽世，以竊取並解開愛音的祕密日記。

愛音與爽世執行的程式原始碼分別為 `rsa/anon.py` 與 `rsa/soyo.py`。你可以透過 `nc 140.112.187.51 11451`、`nc 140.112.187.51 11452` 分別聯繫愛音與爽世。
Note: 若你使用 Python 解題，本題推薦使用 pwntools 或 socket 來與 server 建立連線。

(a) (15 points)(CTF) 愛音在確認與她連線的人就是爽世後，會給對方一個 flag。請取得該 flag，並解釋你的攻擊過程與原理。
Hint: 爽世不幫我簽 `name=soyo`。但似乎可以讓爽世簽其他訊息，再還原回我要的簽章...？

好不容易讓取得愛音信任的你驚訝地發現：即使是爽世，愛音也不願告訴對方自己的日記內容！幸好，她很樂意告訴你他所使用的 public key 與用其加密後的資料，因為她覺得 RSA 既然是基於大數因數分解問題，你不可能解出 private key，當然更不可能解出這個祕密日記的，對…對吧？

(b) (15 points)(CTF) 愛音在她的祕密日記裡藏了一個 flag。請取得該 flag，並解釋你的攻擊過程與原理。
Hint: 這個 public exponent 看起來有點小，會影響 RSA 的安全性嗎？

4. TESTING in the FUZZ (41 points)

儘管現今已有許多框架與函式庫，我們仍時常在軟體實作中發現重大漏洞。許多開發者依賴「Vibe Coding」的設計模式來寫程式：開發者只負責向 LLM 描述需求，而程式碼則由 LLM (e.g. ChatGPT, Copilot) 來自動生成。由於這些 LLM 生成的程式碼通常不經修改或測試就能運作，許多 Vibe Coding 開發者不會、也無須去理解、檢查、修改程式碼，這進而大幅降低了程式設計的門檻，使非程式設計師也能參與開發。[1] 然而，LLM 生成的程式碼往往存在潛在的漏洞，例如 buffer overflow 、double free 、infinite loop 等等。這些缺陷可能被惡意攻擊者利用，對系統造成威脅。

預防軟體實作中的缺陷的其中一個方式就是軟體測試 (software testing)。然而，手動測試 (manual testing) 需要測試者手動生成各種測資作為輸入，來找出足以讓程式崩潰 (crash) 或當機 (hang) 的輸入 (即 proof of concept，PoC)，以證明漏洞的存在。這往往費力又費時，也無法用來測試大型專案。

本題中將介紹模糊測試 (fuzzing/fuzz testing)。這是一種自動化的軟體測試技術，透過向電腦程式提供意外、無效或隨機的資料作為輸入，並監控是否出現異常情況，例如程式崩潰 (crash) 、內建斷言 (assertion) 失敗或潛在的記憶體洩漏 (memory leak)，提供了我們有效率地找出軟體中的缺陷和漏洞的途徑。

(a) (3 points) 請比較 mutation-based fuzzing 與 generation-based fuzzing 之間的差異，至少列出三點。

(b) (4 points) 請簡述 greybox fuzzing 的運作流程 (你可以使用 AFL 作為例子，也可以加上流程圖輔助說明)。

我們將使用知名的 coverage-guided mutation-based greybox fuzzer ——American Fuzzy Lop (AFL) ——(或它的 fork AFL++) 來檢查指定的 C 程式是否有漏洞。要檢查的程式為 `fuzzing/fuzz-me.c`。

(c) (4 points) 使用 AFL 或 AFL++ 對此程式進行 fuzz testing。列出你建置環境、編譯程式、執行 fuzzer 的步驟。

(d) (30 points) 此程式至少包含三個不同種類、能讓程式崩潰 (crash) 或當機 (hang ，即進入無窮迴圈) 的漏洞。承 (c) 小題，請找到**三個**能觸發**不同種類**的漏洞的 PoC (即能觸發漏洞的 input file)，並將它們放在你所提交的 zip 中的 `poc/` 資料夾底下。對於每個你所提供的 PoC ，請解釋對應的漏洞類型，並說明程式哪裡有缺陷以致產生該漏洞。每個漏洞各占 10 分。`fuzzing/example.md` 為作答範例，請盡量依照這個模板來作答 (中英文不拘)。
Hint1: 非使用 AFL 或 AFL++ 解題者將拿不到本小題的任何分數。
Hint2: 建議在虛擬環境 (VM 或 Docker container) 中執行 fuzzer。
Hint3: Compiler optimization 可能會直接移除漏洞，記得關掉它。
Hint4: 若你的 fuzzer 跑了很久都沒有找出其他種類的漏洞，可以試試重跑一次 fuzzer。
Hint5: 在拿到 PoC 後，AddressSanitizer (ASan) 與 GDB 可以幫助你快速定位程式中的漏洞。
Hint6: 若你提供了兩個觸發同一種類漏洞的相異 PoC ，批改者會視這兩個 PoC 為同一個。

**The problem is adapted from CNS 2024 HW3 P1, designed by 楊偉倫.*

[1] 上文摘錄自https://www.bnext.com.tw/article/82704/how-to-vibe-coding-2025

5. 敗北協定太多了！(28 points)

DNS Security

(a) (3 points) 請解釋什麼是 DNS amplification attack，並舉出兩個防止這種攻擊的方法。

(b) (3 points) 請解釋什麼是 DNS cache poisoning，並舉出兩個防止這種攻擊 (或降低攻擊成功的機會) 的方法。

SMTP Security

(c) (4 points) 請解釋什麼是 SPF ，以及這個技術如何防止 email spoofing。

(d) (4 points) 請解釋什麼是 DKIM ，以及這個技術如何防止 email spoofing。

(e) (6 points) 在 SMTP mail receiver 上設定了 SPF、DKIM 以及 DMARC 就能完全過濾掉 spoofed email、一勞永逸了嗎？若是，請提供原因。若否，請舉出一個實際的 email spoofing 手法。
Hint: 這裡有張酷酷的紙

TLS Security

(f) (4 points) 請簡述一個 TLS 的 certificate 裡會有什麼內容跟什麼是 CA (Certificate Authority)。

(g) (4 points) 請解釋什麼是 HTTPS downgrade attack。若你是網站管理員，你會如何防止瀏覽你的網頁的使用者遭受這種攻擊？

6. 猫物語（赤）(34 points)

完美的高材生，王肥貓。身為在各大 CTF 競賽破臺的資安高手，在 NASA HW11 發佈的第一天，竟將本題的所有 flag 給偷走了！經過 NASA 課程眾多磨練與考驗的你，為了同學們的分數的將來，自然是嚥不下這口氣。為了刺探敵情以取回這些 flag，你決定拉近與肥貓的關係，試圖博取他對你的信任，好讓你在他疏於防備時，用你高超的密碼學知識破解出被他藏起來的 flag ——這是爾虞我詐的遊戲..... 信任與背叛的物語。你在 13 週裡所淬煉出的網路攻防知識，如今終於要展露其身姿了！這才是網路世界裡的攻防！攻防！攻防！

肥貓執行的程式原始碼為 `more-ctf/fatcat.py`。你可以透過 `nc 140.112.187.51 1234` 聯繫他。
Note: 若你使用 Python 解題，本題推薦使用 pwntools 或 socket 來與 server 建立連線。

(a) (8 points)(CTF) 臥底中的你得知肥貓只會向他的麻吉透露其中一個被偷走的 flag (`FLAG1`)。請你藉由跟他玩猜數字遊戲，來成為他的麻吉，進而取得 `FLAG1`。
Hint: 你有辦法預測肥貓選的下一個數字嗎？

(b) (10 points)(CTF) 你偶然從同學的口中得知：線代期中考時，肥貓由於早早就寫完所有題目而閒得發慌，竟把另一個 flag (`FLAG2`) 直接寫在答案卷裡！請你破解出肥貓的考試答案卷內容，並取出裡面的 `FLAG2`。
Hint: 你知道 flag 在答案卷裡，也知道 flag 的 prefix，有辦法用這些資訊找出明文嗎？

(c) (8 points)(CTF) 肥貓的一位好麻吉在做研究的空檔不小心被反鎖在廁所裡，暫時無法脫身，於是肥貓請你替他承擔教授指派的工作！為了與肥貓打好關係，你當然不會拒絕。請你在聯繫上肥貓後，輸入選項 4，快速地解決 10 份 PoW (Proof of Work)，並取得肥貓的謝禮——`FLAG3`。

(d) (8 points)(CTF) 喜愛楓之谷的肥貓成立了台大學園楓之谷同好會。由於同好會的活動採會員制，會員們共享了一份密鑰，以此驗證彼此的會員身份。不知道密鑰的你，為了取得會員專屬的 flag (`FLAG4`)，決定冒充為同好會會員。請你讓肥貓成功驗證你的會員身份，並取得 `FLAG4`。