AN TOÀN PHẦN MỀM

1. Mục tiêu khóa học

Lỗ hổng phần mềm

Kỹ thuật phát hiện, khai thác lỗ hổng phần mềm

- Kiểm thử tĩnh
- Kiểm thử động
- Kiểm thử fuzzing
- Khai thác lỗ hổng

Phát triển phần mềm an toàn

- Thiết kế an toàn
- Lập trình an toàn
- Chống can thiệp

2. Phương pháp học tập

Giảng viên cung cấp (gợi ý!!!) tài liệu để học viên nghiên cứu. Học viên thảo luận cùng giảng viên các nội dung đã nghiên cứu.

Học viên thực hiện các bài tập cá nhân 1 và 2 cùng hai bài Mini Test để củng cố kiến thức căn bản về an toàn phần mềm.

Học viên thực hiện hai bài tập nhóm 1 và 2 để phát triển kiến thức, kỹ năng theo một hướng chuyên biệt về an toàn phần mềm.

3. Lịch làm việc

Buổi học	Nội dung
1	Giới thiệu môn học
2	– Bài tập cá nhân số 1 – Thống nhất chủ đề Bài tập nhóm số 1 (Lớp trưởng tổng hợp và gửi danh sách trước buổi học số 2) – Thảo luận nội dung “Lỗ hổng phần mềm”
3	– Thảo luận nội dung “Lỗ hổng phần mềm” – Mini Test 1
4	– Thống nhất chủ đề Bài tập nhóm số 2 (Lớp trưởng tổng hợp và gửi danh sách trước buổi học số 4). – Bài tập cá nhân số 2
5	Bài tập nhóm số 1
6	– Bài tập nhóm số 1 – Mini Test 2
7	Bài tập nhóm số 2
8	Bài tập nhóm số 2
9	Tổng kết

4. Bài tập

4.1. Danh sách bài tập

Bài tập	Yêu cầu
Bài tập cá nhân số 1	- Nộp trước buổi học số 2. - Trình bày trong tối đa 2 trang A4 - Chủ đề "An toàn phần mềm là gì? Cần làm gì để đảm bảo an toàn phần mềm?"
Bài tập cá nhân số 2	- Nộp trước buổi học số 4 - Trình bày tối đa trong 10 trang A4 - Chủ đề 1: "Các dạng lỗ hổng phần mềm do lỗi lập trình. Bản chất và khả năng ảnh hưởng tới tính an toàn của hệ thống đối với mỗi dạng lỗ hổng" - Chủ đề 2: "Trình bày chi tiết 2 dạng lỗ hổng phần mềm do lập trình". "Chi tiết" có nghĩa là với mỗi dạng lỗ hổng cần trình bày được: bản chất, (các) đoạn mã minh họa, giải thích lỗ hổng trên (các) đoạn mã, cách thức khai thác, khả năng tác động, cách thức phát hiện, cách thức phòng tránh".
Bài tập nhóm số 1	- Nộp trước buổi học số 5. - Lựa chọn và trình bày một chủ đề HẸP về an toàn phần mềm. Ví dụ: + Threat modeling + Security Pattern in Software Designing + Code Coverage Problem in Fuzzing Test + Code Obfuscation Techniques + …
Bài tập nhóm số 2	- Nộp trước buổi học số 7 - Lựa chọn và trình bày một chủ đề mang tính THỰC NGHIỆM về an toàn phần mềm Ví dụ: + Kiểm thử mã nguồn với Fortify Staitic Code Analyzer + Kiểm thử fuzzing với AFL + Kiểm thử fuzzing với libfuzzer + Lập trình phát triển một fuzzer + Phân tích, xác nhận lỗ hổng phần mềm sau kết quả fuzzing + Phát triển mô-đun mở rộng cho Metasploit Framework + Khai thác lỗ hổng phần mềm với Metasploit Pro + …

4.2. Yêu cầu về nội dung

– Không dịch máy.

– Bài tập cá nhân

+ Học viên trình bày theo cách hiểu của mình, không sao chép y nguyên từ tài liệu hoặc sao chép của người khác.

+ Kết quả cần nộp: Bản báo cáo đầy đủ.

– Bài tập nhóm:

+ Mỗi nhóm tối đa 5 học viên, trong đó có 1 trưởng nhóm.

+ Mỗi nhóm đề xuất chủ đề cùng với đề cương, thống nhất với giảng viên trước khi thực hiện.

+ Không quá 2 nhóm làm chung một chủ đề. Trong trường hợp này, các nhóm phải làm việc độc lập với nhau.

+ Phải có phân công nhiệm vụ trong nhóm; các buổi làm việc nhóm phải được thực hiện qua Microsoft Teams, được ghi hình và lập thành biên bản (với mỗi nhóm sẽ tạo một Channel riêng).

+ Hàm lượng kiến thức, nội dung phải đủ lớn.

+ Kết quả gồm: Bản báo cáo đầy đủ, Báo cáo dạng slide để thuyết trình, Các biên bản họp nhóm, Tài liệu khác (nếu có).

4.3. Yêu cầu về hình thức (bản báo cáo đầy đủ)

– Định dạng văn bản:

+ Căn lề: trên 2cm, dưới 2cm, trái 3cm, phải 2cm

+ Nội dung chính: phông Times New Roman; cỡ 14pt; giãn dòng 1,3; căn lề đều 2 biên

+ Mã chương trình: phông Consolas; cỡ 10pt; giãn dòng 1,0; căn lề trái

+ Đánh số trang ở cuối trang, căng giữa; không đánh số trang bìa.

– Cấu trúc gồm:

+ Trang bìa

+ Mục lục

+ Danh mục từ viết tắt

+ <Phần nội dung báo cáo>

+ Tài liệu tham khảo

– Giới hạn về số lượng trang chỉ áp dụng cho <Phần nội dung báo cáo>

5. Tài liệu tham khảo

5.1. Slides

5.2. Books

[1] James Helfrich, **Security for Software Engineers**, CRC Press, 2019

[2] Michael Howard, David LeBlanc, and John Viega, **24 Deadly Sins of Software Security. Programming Flaws and How to Fix Them**, Mc Graw Hill, 2010

[3] Brian Chess, Jacob West, **Secure Programming with Static Analysis**, Addison-Wesley, 2007

[4] Michael Howard, David LeBlanc, **Writing Secure Code**, Microsoft Press, 2003

[5] Adam Shostack, **Threat Modeling. Designing for Security**, Wiley, 2014

[6] Tony Hsu, **Hands-On Security in DevOps**, Packt Publishing, 2018

[7] Phil Martin, **Essential CSSLP Exam Guide**, Nonce Corp, 2018

[8] Jame Ransom, Anmol Misra, **Core Software Security**, CRC Press, 2014

[9] Jan Cappaert, **Code Obfuscation Techniques for Software Protection** (Doctoral Thesis), 2012

5.3. Video courses

[10] CISSP Cert Prep 8 Software Development Security (Lynda)

[11] Techniques for Developing Secure Software (Lynda)