Main Assessment
Instructions: Answer THREE Questions in total: ONE from Section A and TWO from Section B. [Q1, Q2, Q3 or Q1, Q2, Q4 or Q1,Q3,Q4]

Module: Penetration Testing & Ethical Hacking
Module Code: ELE8072
Module Owner: Dr Oluwafemi Olukoya
Module Moderator: Dr Chongyan Gu
Session: 2024/2025
Cohort: MSc Applied Cyber Security, PGCert Applied Cyber Security
Assessment Mode: Coursework

Instructions to Candidates
Please read all sections of the instructions before you start answering questions.

1. Format

You can use any word processor to produce your document, but the submission must be **PDF**. Use a simple document style and format:

- Font: Calibri, Arial, Times, Cambria or similar
- Font size: 11 or 12
- Your document should be clear, uncomplicated, and professional.

2. Submission

Your work should be submitted as a single PDF file via the Canvas *Assignments* page. The document must be named with your student number followed by your name. For example, 12345678_John_Doe. Your pdf file should have sections clearly marked for each question you are answering.

You must submit your document via the *TurnItIn* interface within Canvas, at the bottom of the Canvas Assignments page. The *TurnItIn* similarity score for your document must be below **20%**.
For further guidance about submissions via Canvas, see the following links:

- Student Canvas Orientation: https://canvas.qub.ac.uk/courses/145

All submissions must be made no later than 23:59pm on the required date. The standard penalties apply for late submission.

3. Plagiarism

A *TurnItIn* score below 20% does not mean the document is plagiarism-free, and the work will still be verified for authenticity when it is marked. Plagiarism is an academic offence. You should familiarise yourself with the university's regulations regarding academic offences such as plagiarism and collusion. See the link below for more information:
https://www.qub.ac.uk/directorates/AcademicStudentAffairs/AcademicAffairs/GeneralRegulations/Procedures/ProceduresforDealingwithAcademicOffences/
https://www.qub.ac.uk/directorates/AcademicStudentAffairs/AcademicAffairs/GeneralRegulations/GeneralProvisionsRelatingtoAcademicAppealsConductAcademicOffencesandStudentComplaints/

For information about plagiarism and academic writing, see the link below:
https://www.qub.ac.uk/directorates/sgc/learning/LearningResources/Referencing/

If plagiarism or collusion is suspected, the module owner is mandated by the University to report it. If you are found to have plagiarised any part of your work or colluded, penalties can include being awarded a mark of zero (for the entire module) or other disciplinary measures as deep fit by the panel.

4. Declaration of Academic Integrity

This **MUST** be included after the title page of your course work:

By submitting the work, I declare that:
I. I have read and understood the University regulations relating to academic offences, including collusion and plagiarism:
https://www.qub.ac.uk/directorates/AcademicStudentAffairs/AcademicAffairs/AppealsComplaintsandMisconduct/AcademicOffences/Student-Guide/
II. The submission is my own original work and no part of it has been submitted for any other assignments either by me or by anyone else, except as otherwise permitted.
III. The material submitted was not produced by an AI tool using GPT-2, GPT-J, GPT-NEO, GPT-GPT-3.5, ChatGPT or any other chatbot that produces human-like responses to user-generated prompts.
IV. All sources used, published or unpublished, have been acknowledged if appropriate.
IV. I give my consent for the work to be scanned using plagiarism detection and AI detection software.
Signed: [Student's Full Name] **Date:** [Date of Submission]

5. Collaboration Policy

You may discuss the coursework with other students **but do not share attack payloads, code, screenshots, attack scenarios, diagrams, attack chains, supporting output, documentation, appendices, or ANY part of your solution**. Designing an attack scenario and inputs to find where vulnerabilities lie, the impact these vulnerabilities may have and actions that can be taken to improve the security posture of the system; reviewing research works in penetration testing; understanding the role of security models in penetration testing are some of the learning objectives that the assignment fulfils, and with every question, there's a feeling of satisfaction when you have cracked it. However, if someone tells you the solution before you have figured it out yourself, you would have robbed yourself of the best part of this module. **If you discuss the coursework with another student, you must list their name(s) and student numbers(s) in the submission. Each student must write up their solutions independently.**

6. Late Submission of Continuous Assessment/Coursework

Coursework submitted after the deadline will be penalised at the rate of 5% of the total marks available for each calendar day late up to a maximum of five calendar days, after which a mark of zero shall be awarded. More information is available in Section 3.2 here:
https://www.qub.ac.uk/directorates/AcademicStudentAffairs/AcademicAffairs/GeneralRegulations/StudyRegulations/StudyRegulationsforPostgraduateTaughtProgrammes/#d.en.918560

6. Component Weight

- **100%** of Module Mark

7. Assessment Criteria for Postgraduate Taught Programs

The individual questions will be assessed against the postgraduate conceptual equivalents scale, as shown in the following table.

Mark Band	Criteria	Determinator within grade band
80-100	• Thorough and systematic knowledge and understanding of module content. • Clear grasp of issues involved, with evidence of innovative and original use of learning resources • Knowledge beyond module content • Clear evidence of independence of thought and originality • Methodological rigour • High critical judgement and confident grasp of complex issues	Originality of Argument.

70-79	• Methodological rigour • Originality • Critical judgement • Use of additional learning resources.	Methodological rigour
60-69	• Very good knowledge and understanding of module content • Well-argued answer • Some evidence of originality and critical judgement • Sound methodology • Critical judgement and some grasp of complex issues	Extent of use of additional or non-core learning resources
50-59	• Good knowledge and understanding of the module content • Reasonably well argued • Largely descriptive or narrative in focus • Methodological application is not consistent or thorough • Some evidence of critical judgement	Understanding of the main issues
40-49	• Lacking methodological application • Adequately argued • Basic understanding and knowledge • Gaps or inaccuracies but not damaging	Relevance of knowledge displayed
0-39	• Little relevant material and/or inaccurate answer or incomplete • Disorganised • Largely irrelevant material and misunderstanding • No evidence of methodology • Minimal or no relevant material	Weakness of argument

Question 1 (60%) - Penetration Testing & Report Writing

Assessment is made based on reproducible steps and evidence of completion according to the following criteria:

- **Quality of the Report (15%)**
  - **Report Flow and Process**: Organization, presentation, style, clarity, and quality of writing; use of relevant figures, tables, examples (threats, attacks, mitigations etc.) and purposeful diagrams ((security model, application mapping for tracked testing methodology using vertical and/or horizontal documentation tools like text editors or mind maps, application architecture diagrams etc); clear, concise, and relevant content in the report; relevant sources properly and appropriately cited & listed.
  - **Executive Summary** – Analysis (Findings, Business Impact etc.), Recommended Actions (Severity, Urgency etc.)
  - **Recommendations** – meaningful conclusions and remediation.
  - **Findings and Technical Details** – Methodology, Objectives, Scope of Work, Details, Attack Chains, Metrics and Measures, Risk Rating (Likelihood vs Impact), Security Posture, Traceability Matrix.
  - **Appendices:** Relevant supporting output, screenshots, and documentation that demonstrates proof of actions and potential impact of attack path.

- **Rigour of Penetration Testing Techniques (45%)**
  - Identification of the paths to get users and escalate the privileges with the login credentials.
  - The different penetration testing techniques used in the report such as Enumeration, Exploitation, Post Enumeration, Privilege Escalation, persistence etc will be assessed based on the reproducibility of the steps, dexterity of tools, quality of exploits (basic/medium/advanced, well-known/recent etc.), methods used in the different penetration testing techniques, demonstratable proofs and appropriate supporting materials, providing context to scanner results, finding vulnerabilities beyond scanners etc

- Clear evidence of complex problem-solving, with creative and imaginative skills being adopted in the research.
- A critical account of the tools and methods used and the way in which they were selected.

Question 2 (20%)**: **Evaluating Research-Informed Penetration Testing Tools

Assessment is made against the following criteria:

- Clear criteria for evaluation
- Well-defined metrics for achieving the criteria
- Consideration of state-of-the-art benchmarks for evaluating research-informed penetration testing and ethical hacking tools.
- Technical communication of the main ideas presented in the tool.
- An in-depth systematic and critical analysis that demonstrates a thorough knowledge of the tool and offers comprehensive independent insight.
- Use of relevant and appropriate references

Question 3 (20%)**: **Risk-based Vulnerability Management I (Severity and Standards) – CWE, OWASP Top 10, OWASP MASVS etc

Assessment is made against the following criteria:

- Attack surface analysis for mobile applications
- Proposal of a consistent risk scoring system based on MobSF's current severity and standards.
- Additional insights into risk scoring factors beyond what is available in MobSF.
- Clear evidence of independence of thought and significant originality
- Comparative evaluation of the existing app security scoring system with the new proposed scoring system.
- Use of relevant and appropriate references

Question 4 (20%)**: **Risk-based Vulnerability Management II – Reachability Analysis and Blast Radius

Assessment is made against the following criteria:

- Sound methodology for addressing blast radius in reachability analysis.
- Well-argued approach for blast radius calculation.
- Use of relevant and appropriate sample data
- Results and Evaluation of the different components of the blast radius analysis
- A critical analysis using a case-based scenario to demonstrate how reachability score and blast radius can complement risk-based vulnerability management.
- Use of relevant and appropriate references

[End Of Instructions]

Section A – Practical Component (60%)

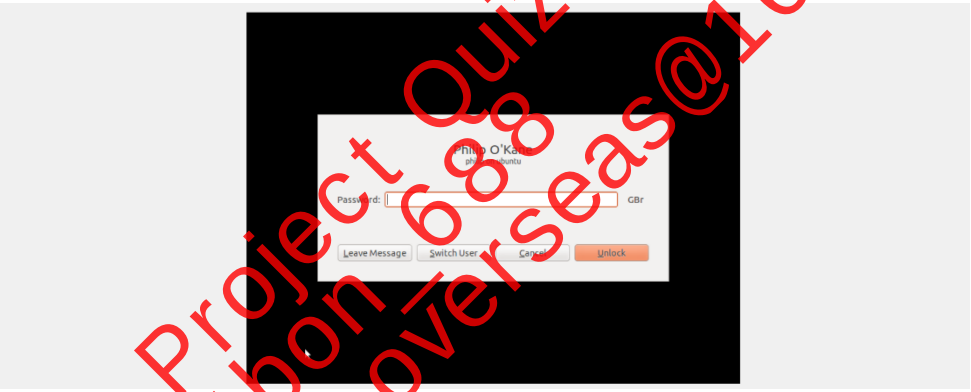
Question 1: Penetration Testing & Report Writing

The Scenario

CyberColony is a cybersecurity start-up company. Before using their services, a potential client has asked for an internal penetration testing report of the CyberColony network environment as part of their due diligence. CyberColony has tasked you with assessing the security of its internal infrastructure and producing a penetration testing report. However, to eliminate the risk of downtime due to penetration testing and to allow for easier reproduction of the found exploits without compromising data, CyberColony has made available a virtual machine for penetration testing. The report should present the result of the penetration testing and vulnerability assessment of the virtual machine and its underlying services that encompasses common penetration techniques such as port scanning, service enumeration, exploitation, post-enumeration, privilege escalation, persistence etc. Amongst other requirements of a penetration testing report, your report should be clear about i) the paths to get users and paths to escalate the privileges ii) how the exploits were conducted iii) steps to reproduce iv) impact/severity of the exploit v) remediation for discovered issues (patching, configuration changes etc.). The CyberColony VM can be accessed here: CyberColony.ova

[NOTE: The length of the penetration testing report **MUST NOT EXCEED 35 pages** – Title pages, and references excluded]

This is what should appear when you power on the CyberColony VM



Section B – Theoretical Component (40%)

Question 2: Evaluating Research-Informed Penetration Testing Tools

The emergence of LLM has led to the development of various cybersecurity applications that utilize this technology for different tasks [5]. Some repositories have curated these tools for cybersecurity research [1,2], including tools for penetration testing. One notable tool is ***PentestGPT***, which is an automated penetration testing tool powered by GPT. Details about the tool are outlined in a research paper published in USENIX Security 2024 [3], and the code repository is hosted on GitHub [4]. You are tasked with conducting an independent study of the research paper and the tool to present an evidence-based critical analysis of ***PentestGPT***, using relevant criteria and metrics for evaluating penetration testing tools.

References

1. Awesome Large Language Model Tools for Cybersecurity Research: https://github.com/tenable/awesome-llm-cybersecurity-tools
2. Awesome GPTs (Agents) for Cybersecurity: https://github.com/fr0gger/Awesome-GPT-Agents
3. Deng, G., Liu, Y., Mayoral-Vilches, V., Liu, P., Li, Y., Xu, Y., Zhang, T., Liu, Y., Pinzger, M. and Rass, S., 2024, August. PentestGPT: Evaluating and Harnessing Large Language Models for Automated Penetration Testing. In *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association.
4. A GPT-empowered penetration testing tool: https://github.com/GreyDGL/PentestGPT
5. Chris Madden (2024). Language Models for Cybersecurity – An Applied Guide: https://cybersecai.github.io/

Question 3: Risk-based Vulnerability Management I (Severity and Standards)– CWE, OWASP Top 10, OWASP MASVS etc

The Mobile Security Framework (MobSF) [1] is a research platform for mobile applications in Android, iOS, and Windows Mobile. MobSF can be used for various purposes such as mobile application security, penetration testing, malware analysis, and privacy analysis using static and dynamic analysis. Additionally, MobSF generates a detailed static analysis report that includes findings from the application evaluation, such as the app security score and grade, CWE, OWASP Top 10, OWASP MASVS tags from the code analysis report, and more. MobSF uses "High," "Warning," and "Secure" severity findings to calculate an app security score. Below is the historical logic used for calculating the app security score.

App Security Score Calculation

Version #1

```
avg_cvss = round(sum(cvss_scores) / len(cvss_scores), 1)
app_score = int((10 - avg_cvss) * 10)
```

Version #2

Every app is given an ideal score of 100 to begin with.
For every finding with severity high we reduce 15 from the score.
For every finding with severity warning, we reduce 10 from the score.
For every finding with severity good we add 5 to the score.
If the calculated score is greater than 100, then the app security score is considered as 100.
And if the calculated score is less than 0, then the app security score is considered as 10.

Risk Calculation

APP SECURITY SCORE	RISK
0-15	CRITICAL
16-40	HIGH
41-70	MEDIUM

71-100	LOW

Some discussion on the App Security Score inconsistencies can be found here [2.3].

 # Current Logic

App Score Logic[4]

```
    high = len(findings.get('high'))
    warn = len(findings.get('warning'))
    sec = len(findings.get('secure'))
    total = high + warn + sec
    score = 0
    if total > 0:
        score = int(100 - (((high * 1) + (warn * .5) - (sec * .2)) / total)
* 100)
    if score > 100:
        score = 100
   findings['security_score'] = score
```

MobSF also grades an application based on its risk level [5].

APP SECURITY SCORE	GRADE
0-29	F
30-39	C
40-59	B
60+	A

Design an app security scoring system that accurately measures the risk posed by vulnerabilities in mobile applications, utilizing the MobSF static analysis report and vulnerability risk scores assessment. Compare and contrast your system with the app security score and grade currently used by MobSF, while justifying your proposed app security scoring system.

References

1. Mobile Security Framework (MobSF): https://github.com/MobSF/Mobile-Security-Framework-MobSF
2. [FEATURE] Improve security scoring of apps: https://github.com/MobSF/Mobile-Security-Framework-MobSF/issues/1069
3. App Security Score inconsistencies: https://github.com/MobSF/Mobile-Security-Framework-MobSF/issues/1940
4. appsec.py: https://github.com/MobSF/Mobile-Security-Framework-MobSF/blob/master/mobsf/StaticAnalyzer/views/common/appsec.py
5. appsec_dashboard.html: https://github.com/MobSF/Mobile-Security-Framework-MobSF/blob/master/mobsf/templates/static_analysis/appsec_dashboard.html

Question 4: Risk-based Vulnerability Management II – Reachability Analysis and Blast Radius

As a penetration tester, you have been tasked with analyzing and adapting a graph theory-based network reachability [3] analysis solution for prioritizing vulnerability remediation prioritization as part of your organization's maturity model.

Description: The solution uses in-degree centrality in graph theory to develop a reachability metric for vulnerability remediation prioritization. The implementation uses the *Networkx* library to calculate the reachability of network-based vulnerabilities on IT assets, using sample data from firewall (management) solutions, vulnerability management solutions and DHCP servers, and trust values for different zones/subnets. Additional details about the description [1] and implementation [2] of the solution have

been provided. The starting code in Python and the output have also been provided as part of the assessment files.

Tasks: Adapt the code to determine the '*blast radius*' of an exploited vulnerability. Take reachability from the same zone/subnet into account. You can apply exploit chaining (https://cyberhoot.com/cybrary/exploit-chain/) or any other relevant methodology to address this task. You are also permitted to generate additional appropriate sample data for your analysis. Using an evidence-based scenario, discuss the integration of the reachability score and blast radius of an exploited vulnerability into risk-based vulnerability management

Deliverables: Updated Code, Example result (table or graph) of the reachability analysis that includes the blast radius, visited nodes, exploitable links and any other relevant output, and the discussion of streamlining reachability score and blast radius with existing vulnerability prioritization metrics. This information should be appended to the pdf document.

Note: This question requires the ability to understand and write code in Python, Basic TCP/IP knowledge, graph theory and relevant design of the IT network.

References

1. Albert-Jan Talsma (2024): Data-driven Vulnerability Management: Graph Theory based Reachability Analysis (1/2): https://medium.com/@a.talsma/data-driven-vulnerability-management-graph-theory-based-reachability-analysis-1-2-61f2fe185339
2. Albert-Jan Talsma (2024): Data-driven Vulnerability Management: Graph Theory based Reachability Analysis (2/2): https://medium.com/@a.talsma/data-driven-vulnerability-management-graph-theory-based-reachability-analysis-2-2-4378e6a3de9c
3. Khakpour, A.R. and Liu, A.X., 2010, June. Quantifying and querying network reachability. In *2010 IEEE 30th International Conference on Distributed Computing Systems* (pp. 817-826). IEEE.

[End Of Questions]