NAPIER UNIVERSITY 2021
GROUP PROJECT
PENETRATION TESTING SCENARIO
FINAL REPORT

Project Manager: Kenneth Brown

PM Support/ Server Administrator: Davide Pisanu

Security Team Manager: Tom Neil

Security Team: Luis Loaysa

Web Team Manager/ Back-End Developer: Connor Grattan

Web Team / Front-End Developer: Jake Salt

Project Client: Robert Ludwiniak

Project Sponsor: Andrew Partridge

Contents

Section 1 - Executive Summary 2
Section 2 - Delivered Product 3
2.1 – Must-Have Deliverables 3
2.2 – Should-Have Deliverables 3
2.3 – Could-Have Deliverables 3
Section 3 - Client Approval 4
3.1 - Client sign-off 4
Section 4 - Closing Audit 5
4.1 – Team Satisfaction 5
Figure 1: Team Satisfaction Survey 5
4.2 – Customer Satisfaction 5
Figure 2: Customer Satisfaction Survey 5
Section 5 - Lessons Learned 6
5.1 – Methodology 6
5.2 – Teamwork 6
5.3 – Communication 6
5.4 – Measuring Team Contributions 7
Appendix 8
1.0 - MoSCoW Prioritisation Scheme 8
2.0 - Deliverables 9
3.0 - Schedule Model 10
Figure 3: Schuduled Tasks 10
Figure 4: Gantt Chart 10

Approved by the Project Sponsor

Section 1 - Executive Summary

The team was tasked with creating a virtual cybersecurity platform for carrying out penetration testing. Our client wants to use this platform for teaching and to create labs for fourth-year students. The security-based scenario we were asked to create involves hosting an E-commerce style website inside of an unsecured server. This website was a requirement set by the client; however, the team was given creative freedom to explore implementing other ideas for security exploits within the test environment. Several main goals were set at the beginning of the project: -

- **Create a pre-configured virtual environment or set of environments, which can be used to carry out Cybersecurity penetration testing.**
- **Create an E-commerce style website within the virtualized environment, designed for simulating and testing web-based cyber-attacks.**
- **Create documentation that explains each step of all successful pen-tests in such a way that they can be easily recreated later for lab purposes.**
-

In addition to these main goals, the client also required that the platform makes use of modern Operating Systems, the vulnerabilities present could be exploited realistically by the students, and the exploitation of said vulnerabilities can be performed in a logical initiative manner that conveys the idea of privilege escalation. The client stated that they hope to use this work to improve upon the current standard of student labs within pen-testing by creating scenarios that closely resemble situations found within real-world IT environments. Whilst these additional requirements are desirable, it was acknowledged both by the client and the team during the planning stages that these are potentially unachievable within the scope of the project and were given a lower MOSCOW priority (**Appendix 1.0**). Some of the major challenges presented to the group during the development phase of this project were as follows: -

- **The scenario was developed by a group of security students who are not yet studying at the level that these labs are intended to eventually teach. This required that the team carry out extensive individual research into many areas that are relatively unfamiliar to their current studies.**
- **Cybersecurity & penetration testing are vast subject areas and the security team had to act quickly to decide which tests were most likely to yield positive results within our environment and avoid scope creep.**
- **Hosting an E-commerce website realistically and also offline is a challenge due to the nature of online payment systems and their integration into modern websites. The team was not expected to be able to safely crack modern security systems or reveal any undiscovered vulnerabilities in this type of secure technology.**

Upon the project's closure, the team has successfully developed several pen-test exploits within the virtual environment and created extensive documentation to support each of these. The individual exploits are intended to be used as separate labs, but also have a logical exploit structure that allows students to move through each test hierarchically. In addition to these exploits, there are still several in progress such as the SQL injection, FTP server exploits, and development of rainbow tables. Future improvements on this scenario, outside of the project timeframe, have been discussed with the client and are currently in development.

Section 2 - Delivered Product

This section of the report refers to the MOSCOW prioritization scheme (**Appendix 1.0**). When comparing the delivered product to the original plan this scheme has been used to show which deliverables were successfully completed for the client (**Appendix 2.0**).

2.1 – Must-Have Deliverables

- A Virtual-Network was **successfully created and delivered**.
- An unsecured, pre-configured server VM **was successfully deployed and delivered**.
- An E-Commerce style website designed to be intentionally vulnerable **was successfully implemented and delivered**.
- **AT LEAST** one penetration test which exploited vulnerabilities using user-configurations, hardware/software versions, or the website, **was successfully created and delivered.**
- Documentation which clearly demonstrated to the reader in a step-by-step format how to recreate our pen-test in a student lab environment **was successfully created and delivered.**
- OVSM (VMWare) Virtual Machine files, a cloned copy of the final websites GitHub Repository and all relevant project documentation **were successfully created and delivered.**

2.2 – Should-Have Deliverables

- **AT LEAST** three penetration tests, which demonstrate the exploitation of vulnerabilities across all the configured network elements (Server, Users, Website) **were successfully created and delivered.**
- Network should contain vulnerabilities but should not be left without security entirely (no firewall, no user security, outdated protocols/OS Versions) to maintain a realistic and challenging scenario for student labs. **This was partially delivered. Whilst some layers of security are in place, many of the current basic security standards had to be disregarded to allow most of our exploits to function appropriately**.
- Wherever possible, exploits should be shown to be utilising the project website. **All of the six completed exploits were shown to be exploiting or utilising the projects website in a variety of ways.**

2.3 – Could-Have Deliverables

- **AT LEAST** five penetration tests, demonstrating multiple exploits of vulnerabilities across all configured network elements. **Six exploits were successfully delivered, with three more still currently in progress outside the scope of the project.**
- A realistic network which adheres to many of the modern security standards that are currently used in real business environments. Vulnerabilities on this network will be carefully tailored and well hidden, to simulate scenarios that could logically occur outside of a lab environment. **This was not delivered successfully. Whilst the group made efforts to maintain realism throughout the network, the final product still requires many unrealistic configuration changes for all exploits to properly function.**
- Self-directed tasks for students, such as locating hidden information within the network architecture. **This was partially delivered. The Cross-Site scripting exploit encourages students to explore online for various payloads upon completion of the step by step that was provided and the unfinished SQLi exploit provides similar opportunities.**

Approved by the Project Sponsor

Section 3 - Client Approval

This section is a confirmation of the client's approval of the delivered product as it is described in this report. The document shown below was signed by the client (**Appendix 2.0**) and provided here as evidence of the work that was completed and delivered by the team.

3.1 - Client sign-off

Penetration Testing Scenario

Materials checked against PMIS

- **A pre-configured virtual environment or set of environments, which can be used to carry out one or more types of penetration tests.**
- **An E-commerce style website within the virtualized environment, designed for simulating and testing web-based cyber-attacks.**
- **Documentation which explains each step of all *successful* pen-tests in such a way that they can be easily recreated later for lab purposes.**

I confirm that the results from the project have been completed and delivered as agreed.

Signed: Robert Ludwiniak

Date: 29/04/21

Approved by the Project Sponsor

Section 4 - Closing Audit

4.1 – Team Satisfaction

Team satisafaction was measured using the Project Health Register. It was decided that the greatest level of accuracy could be provided by conducting a single survey in the final week of the project schedule. This was decided due the project managers failure to recognise the importance of the health register until halfway into the project. So, not wishing to potentially fabricate any previous results, only one survey was conducted The Project Health Register can be viewed via the Github Link provided (**Appendix 2.0**).

The overall satisafaction rating is based on a scale of 0 to 100 across a total of 10 questions and six team members. Question five is not included due to there being no supplier for the project. The overall score for Team Satisfaction was 90%. The following table shows the average results for each question asked in the survey: -

Item	M01
Do you know what is expected from you, and what you can expect from others?	84
How effective is the communication between you and the person you directly report to?	91
How effective is the communication between you and your teammates?	90
How effective is the communication between you and your peers in the customer side?	77
How effective is the communication between you and your peers in the supplier side?	
Are the project targets set realistically?	90
Does the project management system protect you, and help you work on the project comfortably?	93
Is the Project Manager supportive?	96
Do you have a clear image of the project as a whole? Do you know your role in this mission?	90
Overall, are you happy working in this project?	96
Overall, how do you rate the project management system?	85
Score:	90

Figure 1: Team Satisfaction Survey

4.2 – Customer Satisfaction

As with the Team Satisafaction survey, one late-stage Customer Satisfaction survey was carried out. To support this decision it should be stated that regular meetings with the client were held throughout the project and at no point was any dissatisfaction expressed with the teams progress. This negated any immediate need to take the time to guage satisfaction levels when progress on the project was of a much higher priority in the early to mid stages of the project. The overall score for customer satisfaction was 83%. Individual questions and their responses are shown : -

Item	M01
Is it easy to communicate with us and let yourself understood?	80
Are we responsive enough in our communications?	80
Is our project management process clear and transparent enough for you?	70
Do you have a clear image of what we're doing at any time?	85
Do you have a clear image of what we're going to do in the future?	70
Overall, how do you rate our project management process?	80
Score:	78

Figure 2: Customer Satisfaction Survey

Approved by the Project Sponsor

Section 5 - Lessons Learned

5.1 – Methodology

Using an Agile project management framework to arrange the scheduled tasks (**Appendix 3.0**), was extremely beneficial to the final products overall quality. The exact nature of the tasks was unknown to the team during the planning stages of the project. Using this methodology meant that the implementation time could be split into three, fort-nightly iteration cycles. Within a single cycle the group was given five working days to explore an individual members ideas and find out which were possible. The next three week days would be spent taking these ideas into our own virtual environment which had been deployed early in the planning stages, and test them here for definitive results. The remainder of the time was spent analysing results, making suggestions for improvements and making plans for the next iteration.

The Agile Framework was of great benefit here because if tasks had been strictly scheduled in advance, it is highly likely that much of our time would have been lost when a particular idea was unsuccessful, as this would have delayed subsequent tasks, sometimes indefinitely. An example of this is the SQL injection. One member of the group was assigned to this particular exploit but ultimately the final product does not contain a comprehensive overview and is still currently in development. Had the whole team been dedicated to this task, it may have been completed, but if it had not then other exploits that *were* discovered in that time may not have been developed enough for our final hand-in. Agile allowed the team the necessary time to carry out individual research alongside one another to negate this potential loss of time.

5.2 – Teamwork

The team worked together extremely well throughout the duration of the project. This can is mainly attributed to the fact that everyone on the team had expressed enthusiasm for this specific project at an early stage. This fact is reflected in the 90% Team Satisfaction rating shown in the previous section. Despite this there were still issues throughout that had to be addressed such as avoiding conflicts with multi-tasking when other members were working towards a similar goal. This was primarily seen in the deployment of the E-commerce website when code was either not being pushed on time to the GitHub repository or being pushed without all members of the web teams neccesary aknowledgements. This led to a situation that was resolved quickly by acknowledging the issue privately with the project manager and having them address the team directly, without conflict. The importance of avoiding conflict here cannot be understated.

5.3 – Communication

Our group used Discord and Microsoft Teams for communication throughout the project. Due to the Covid 19 pandemic the team was not able to conduct any meetings in person so all activity was online. Teams was used for live meetings which took place roughly twice a week and once a week with the group sponsor. The rest of our communication was via Discord. The use of separate channels for the relevant area of the project was useful here such as the separation between research carried out by the web team and security team. Work for the final report, PMIS, presentation and final hand-in were all carried out in separate channels which made navigating the related information much more accessible after the fact.

5.4 – Measuring Team Contributions

The group members individual contributions to the project were measured using a weekly spreadsheet and percentage system. Each of the six members start with an equal weekly contribution share of 16.7%. At the beginning of each week all members were asked to place two votes into a contributions Discord channel. One vote is always for the member themselves and this is to encourage each member to speak up and recognise the work they themselves had contributed. The other vote asked each member to vote for another team member that they had recognised as having made a significant contribution that week. This system was effective in regard to avoiding conflicts as it focusses soley on the positive aspects of the working week. Group members were not encourage to use this as an opportunity to point out who had NOT been contributing, this rare type of situation was addressed privately with the members involved.

One area this system is flawed in was if a group member was not voted for that week but had still made a significant contribution to the project. This would result in that member receiving a negative value for the week, making it appear as though they had not worked as hard as the rest. In this situation the project manager made it clear that the final results after voting were not finalised at this point. More in depth discussion than a single vote would be needed here to determine if that member should receive a higher score that week.

On two separate weeks throughout the project it was decided that due to the whole group being present and contributing equally, the voting system would not be used for that week and every member would receive an equal 16.7%. This system helped keep the team motivated throughout to perform well individually. This is reflected in the fact that the final contributions saw every member of the group fall within the same 4% margin (14% - 18% overall contribution). The group contribution spreadsheet has been provided separately, alongside this report. The weekly voting calculations are also available in the ‘Kenneth’ Github folder (**Appendix 2.0.**)

Appendix

1.0 - MoSCoW Prioritisation Scheme

To allow flexibility in the scope, DSDM Atern uses the MoSCoW prioritization technique. This is a four-level scheme that is applied to the requirements that will be attempted during a timebox. The four priority levels are described in the table below.

Label Interpretation

M - Must-have items are essential for the product or for the business case of the project.

S - Should-have items are not essential but are nevertheless important for the quality of the finished product.

C - Could-have items are features that would be nice to have, but which would not compromise the overall quality if they were missing.

W - Won't-have items are not included in the current scope - this final category is more important than it first appears.

[M]ust Have

- A Virtual-Network
- An unsecured, pre-configured server VM.
- An E-Commerce style website designed to be intentionally vulnerable.
- **AT LEAST** one penetration test which exploits vulnerabilities using user-configurations, hardware/software versions, or the website.
- Documentation which clearly demonstrates to the reader in a step-by-step format how to recreate the pen-test in a student lab environment.
- Final Product hand should contain OVSM (VMWare) Virtual Machine files, a cloned copy of the final websites GitHub Repository and all relevant project documentation in PDF format.

[S]hould Have

- **AT LEAST** three penetration tests, which demonstrate the exploitation of vulnerabilities across all the configured network elements (Server, Users, Website).
- Network should contain vulnerabilities but should not be left without security entirely (no firewall, no user security, outdated protocols/OS Versions) to maintain a realistic and challenging scenario for student labs.
- The website should be of particular focus due to the time investment required for web development when designing and deploying an effective web strategy. Wherever possible, exploits should be shown to be utilising the project website.

[C]ould Have

- **AT LEAST** five penetration tests, which demonstrate multiple exploits of advanced vulnerabilities across all the configured network elements (Server, Users, Website).
- A realistic network which adheres to many of the modern security standards that are currently used in real business environments. Vulnerabilities on this network will be carefully tailored and well hidden, to simulate scenarios that could logically occur outside of a lab environment.
- Self-directed tasks for students, such as locating hidden information within the network architecture.

[W]on't Have!

- Most up-to-date Server/Host OS for exploit targeting. The team cannot, reasonably, expect to uncover new exploits, given a modern OS's high levels of security. This is the type of work currently being carried out by professional pen-testers and is well outside the scope of this university project. The team is focussing on OS models from 2016 onwards.
- A fully functioning payment system for the E-commerce website. Payment systems through banks and browsers are heavily regulated and attempting to break these systems without proper authorization to do so would most likely result in legal action against the individuals within the group or against the school of computing at Napier university.

2.0 - Deliverables

The following link is for the group GitHub repository. All the the deliverables and documentation described in the Delivered Product section of this report are stored here. The reason for this being that there is too much documentation to reasonably include within this appendix.

To view all deliverable related documentation please access the 'Penetration Testing Scenario – Exploit Documentation' PDF file. For individual members work please access the relevant folder. The rest of the project documentation is also stored in the main directory and is available for viewing. Documentation relating to project management, such as the Health Register & PMIS documents, is stored in the '**Kenneth**' folder as it was produced by the project manager.

Documentation & Deliverables available at: - https://github.com/Sly-Lamp/GRP14_PMIS

3.0 - Schedule Model

	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**Research & Planning Stage**	**20.88 days?**	**Mon 18/01/21**	**Mon 15/02/21**	
2		Virtual-Server Configuration	20.88 days	Mon 18/01/21	Sun 14/02/21	
3		Kick-Off Meeting	0 days	Mon 25/01/21	Mon 25/01/21	
4		Penetration-Testing Research	10.88 days	Mon 25/01/21	Mon 08/02/21	3
5		Web Vulnerabilities Research	10.88 days	Mon 25/01/21	Mon 08/02/21	3
6		OS Vulnerabilities Research	10.88 days	Mon 25/01/21	Mon 08/02/21	3
7		Sponsor Meeting	0 days	Thu 04/02/21	Thu 04/02/21	
8		Website Development	5 days	Tue 09/02/21	Sun 14/02/21	5
9		PMIS Submission	0 days	Fri 12/02/21	Fri 12/02/21	
10		Virtual Environment Deployment	0 days	Mon 15/02/21	Mon 15/02/21	2
11		Website Deployment	0 days	Mon 15/02/21	Mon 15/02/21	8
12		**Implementation and Development Stage**	**30.88 days?**	**Mon 15/02/21**	**Mon 29/03/21**	
13		**Iteration 1**	**10.88 days?**	**Mon 15/02/21**	**Mon 01/03/21**	
14		Deployment	5.88 days	Mon 15/02/21	Mon 22/02/21	
15		Testing	3.88 days	Mon 22/02/21	Thu 25/02/21	
16		Analysis	1.88 days	Fri 26/02/21	Mon 01/03/21	
17		**Iteration 2**	**10.88 days**	**Mon 01/03/21**	**Mon 15/03/21**	
18		Deployment	5.88 days	Mon 01/03/21	Mon 08/03/21	
19		STARL Submission	0 days	Mon 08/03/21	Mon 08/03/21	
20		Midpoint Progress Review	0 days	Mon 08/03/21	Mon 08/03/21	
21		Testing	3.88 days	Mon 08/03/21	Thu 11/03/21	
22		Analysis	1.88 days	Fri 12/03/21	Mon 15/03/21	
23		**Iteration 3**	**10.88 days**	**Mon 15/03/21**	**Mon 29/03/21**	
24		Deployment	5.88 days	Mon 15/03/21	Mon 22/03/21	
25		Testing	3.88 days	Mon 22/03/21	Thu 25/03/21	
26		Analysis	1.88 days	Fri 26/03/21	Mon 29/03/21	
27		**Evaluation & Aanalysis Stage**	**20.88 days?**	**Mon 29/03/21**	**Mon 10/05/21**	
28		Presentation	0 days	Mon 19/04/21	Mon 19/04/21	
29		Delivery	0 days	Mon 26/04/21	Mon 26/04/21	
30		Report	0 days	Mon 26/04/21	Mon 26/04/21	
31		Presentation Self Evaluation	0 days	Mon 03/05/21	Mon 03/05/21	

Figure 3: Schuduled Tasks

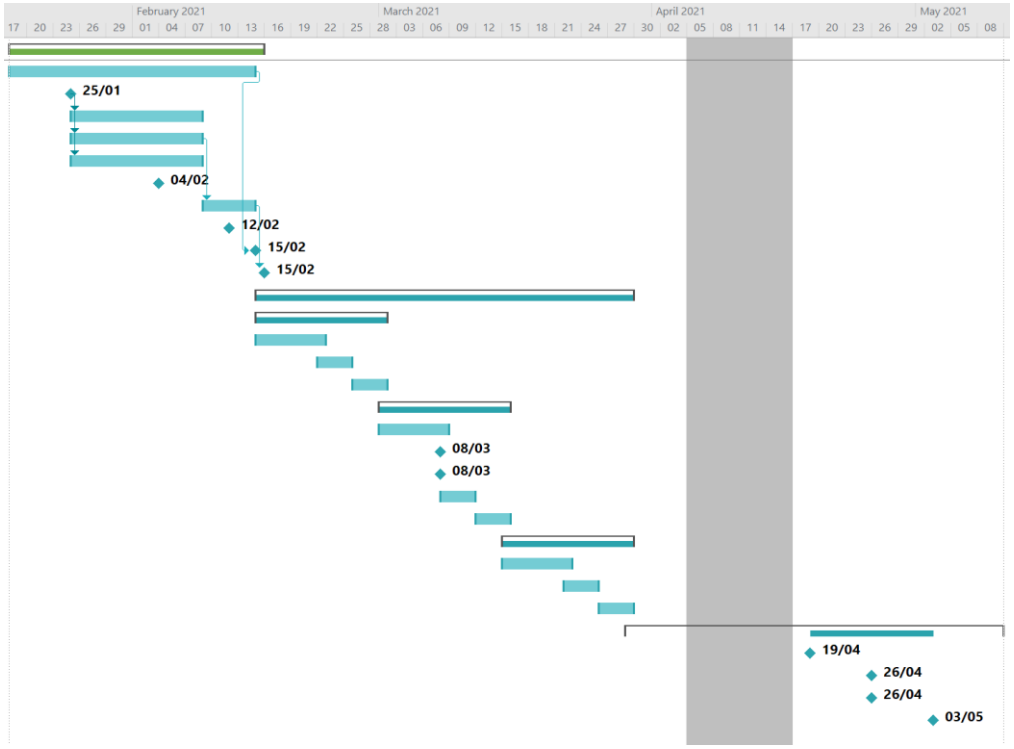

Figure 4: Gantt Chart

Approved by the Project Sponsor