Annals of Emerging Technologies in Computing (AETiC)
Vol. 4, No. 2, 2020

Research Article

Automating the Assessment of Networking and Security in Higher Education

Neville Palmer

School of Media Arts and Technology, Solent University, UK
neville.palmer@solent.ac.uk
*Correspondence: neville.palmer@solent.ac.uk

Received: 18th January 2020; Accepted: 19th February 2020; Published: 1st April 2020

Abstract: In the subject of computer networking students must understand the concepts and be confident in configuring and securing a range of network devices and systems. Challenges are faced in assessing practical networking and security exercises in a laboratory environment utilising multivariate devices and operating systems that involve real or simulated systems operating in real or virtualized environments. The working environment for a practical exercise must be pre-configured and once an exercise has been completed the results extracted to enable formative or summative assessment of the outcomes. Using manual methods this process can be time consuming and it would therefore be beneficial to implement some form of automation. To facilitate this a new application has been developed that automates the configuration management and assessment processes within a computer networking laboratory. The new application has been successfully used to assess the extent to which students have been able to configure secure networks using case-study based exercises. The development challenges of the application and rationale for the implementation of the prototype application are discussed. A test methodology is presented in which the application is used to assess the outcomes of a computer networking exercise, comparing the results obtained from the application with those of a proprietary simulator. This demonstrates that the prototype application is able to successfully and accurately automate the assessment of the practice of networking and security in the context of defined parameters. Further work is suggested to improve the assessment mechanism employed by the application, seek enhancements and to conduct additional tests.

Keywords: ***Computer Networking; Security; Laboratory; Automation; Assessment***

---

1. Introduction

Computer Network engineers must be skilled in configuring and managing networks, but they must also be competent in ensuring that they are secure. A common strategy for network security involves a Defence in Depth approach, in which a number of technologies work together to secure various parts of a network from access to edge [1], or rather Defence in Breadth [2], where

Neville Palmer, "Automating the Assessment of Networking and Security in Higher Education", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 1-13, Vol. 4, No. 2, 1st April 2020, Published by International Association of Educators and Researchers (IAER), DOI: 10.33166/AETiC.2020.02.001, Available: http://aetic.theiaer.org/archive/v4/v4n2/p1.html.

overlapping technologies can be better used to protect networks. Computer network engineering graduates may be involved in managing and securing systems incorporating multivariate technologies. Feisel and Rosa [3] highlight the importance of practical discipline in engineering and emphasise that students should use an instructional laboratory to learn things that practicing engineers need to know. Employers need to be confident that graduates possess the skills that are useful to them [4]. This means that practice is just as important as theory in meeting the expectations of their profession. In the example of computer networking and security students should be able to apply networking principles to practice. In security terms they should not only understand attack vectors, but also the principles and practice in terms of defence.

Laboratories play a key role in building the necessary technical and practice-based skills. In the case of computer networking the management of shared laboratory resources and the assessment of the practical outcomes of laboratory work can be challenging. These challenges were discussed in previous work [5]. It is also important to understand that teaching, learning and assessment should encourage students to design solutions to problems for themselves rather than merely following or repeating instructions [6].

The outcomes of a network security exercise should allow the assessor to judge the extent to which the student has applied security techniques to a computer network. Students could practice techniques and then report on the outcomes of their work, but it may be desirable to measure the outcomes of this work directly. This is the approach advocated by Cisco in particular, utilising practiced based skills tests in which students are expected to implement networking principles and security using simulators or real networking equipment. The main focus of these exercises are routers, switches or security appliances. The Cisco Certified Network Associate (CCNA) Security course covers many of the fundamental network security systems essential for a modern business [7], [8]. For example access to the network could be protected by a zone based firewall. Communication between offices would need to be encrypted by a Virtual Private Network (VPN) tunnel. It would be wise to incorporate an Intrusion Detection (IDS) or Intrusion Prevention System (IPS) in the network. Some form of secure authentication of users and routing protocols could be implemented, along with authorization, accounting and logging functions that record activity on the network. There would need to be additional layer 2 security on switches. Dedicated security appliances could better support the Defence in Depth approach. However students should initially understand the principles of networking. This requires competency in IP addressing, configuration of network devices, including routing and switching protocols, hosts, including services, and so on. Employers expect students to possess knowledge of these concepts and expect them to be able to apply them in a professional context. Case studies are more likely to form the basis of a focused application based approach [4]. At the author's institution some educational modules are based on the contents of the Cisco CCNA range of courses. Case studies involving typical business scenarios are often used as the basis for teaching the modules through a series of laboratory-based sessions in which students implement networking and security strategies based on given requirements. To align with the Cisco assessment strategy Time Constrained Assignments (TCA) are often used that involve similar case studies in which students apply these techniques. Whilst simulators are available it is also useful for students to be able to configure real networking devices. In the laboratory real computers running Linux or Windows virtual machines can be connected to these devices and it is then possible to configure them using graphical interfaces and not just a command line interface. Cisco's Packet Tracer network

simulator has sophisticated assessment automation features in the form of the Assessment Wizard in which the assessor can select items to include in an assessment rubric [9]. However it cannot assess the configuration of real equipment, the output of other simulators, such as Graphical Network Simulator 3 (GNS3) [10], multi-vendor systems or operating systems based on real or virtual machines. Nevertheless it could be used as a suitable tool for comparison purposes to test the validity of results from a prototype assessment application. It would be useful to assess whether students have successfully configured networking and security features on a range of network devices and computers attached to them. Another challenge when setting practical exercises involves the need for configuration management in which pre-configuration of some devices is required. For example in a practical exercise students might assume that items like the IP address of interfaces and routing protocols are already configured at the start of the exercise if these attributes have already been covered in pre-requisite modules. The laboratory environment, including configurations and other files should be uploaded to the laboratory network at the start of an exercise. Prior to a practical exercise the tutor will work out solutions in the form of configurations that should be applied to the devices. When marking the practical exercise without the benefit of automation they might compare the output of each student, in the form of completed configurations, against this known solution. To assess the work of a group of students in this way may take many hours, ignoring the extra overhead of pre and post configuration duties. In previous work an application was proposed that would automate the configuration and assessment of practical exercises in the laboratory so that it follows the current manual practices in order to save some of the time involved in assessment [5].

2. Design and Implementation

The requirements of the new application were identified from the characteristics of current laboratory assessment practices and processes. The new application must be capable of assessing practical exercises in the laboratory and be capable of working with real or simulated multi-vendor network devices, and Windows or Linux operating systems based on real or virtual machines. It would need to compare a working solution to a given exercise against the outcomes of student work and it would be useful to provide feedback on progress to students. There would need to be some mechanism for configuring the laboratory environment in preparation for a practical exercise and then extracting completed configuration files from the laboratory once the exercise has been completed. It would also be useful to monitor the exercise whilst in progress.

The configuration management aspects of the requirements could be met by a third party application such as Red Hat Ansible [11] or other similar systems like Chef or Puppet [12]. Nevertheless, an application had been developed from previous work that provided configuration management mechanisms suitable for the laboratory network [13]. This used PowerShell scripts to communicate with and configure the Windows laboratory computers, whereas a new application would need to communicate with multi-vendor network devices and operating systems. A new application could adapt some of the principles of the previous system and use Secure Shell (SSH), which can provide a secure connection to any network device or computer operating system. Scripts could then be executed in the native device language once a communication channel has been established. This would also avoid the need to translate configuration information into YAML Ain't Markup Language (YAML) syntax used by systems such as Ansible [14]. The existing application was developed in Visual Studio using the C# language since it allows rapid application development.

An assessment application has also been developed using Visual C# into an Open Source GitHub project called Ultramarker that enables an assessor to define various types of assessment consisting of criteria and grading schemas [15]. It can automatically generate a report for students that provides detailed feedback on assessed work, including suggestions for improvement. Since the production of manually written feedback can be time consuming it has been found that this application is able to provide essential feedback in a timely manner. The assessment application is able to calculate a grade from weighted criteria. Some of the principles of marking and automated feedback generation can be incorporated in the new application, which could re-use code from the earlier applications if it were developed in Visual C#.

A mechanism was required for comparing the template solution to the output of student work. Network device configurations can be saved as text files, for example "startup_config" files in the case of Cisco devices. Server systems either save or can output configuration information in text form, for example Domain Name System (DNS) servers store their configuration information in text-based zone files. The Linux operating system stores most of its configuration information in text-based "conf" files. Prior to the start of a practical exercise, following any pre-configuration work, the configuration files from each laboratory station, consisting of their component network devices and computers, can be extracted. This information would form an initial baseline configuration. In order to produce a baseline representing the solutions the assessor would need to complete the exercise themselves by entering the correct solution into all devices and computers in the exercise. The new application would need to conduct a comparison between the initial and the solution baselines. If the system is working perfectly and the assessor has entered all of the required solutions correctly the application should indicate a rating of 100%. Then students would be tasked to work on the exercise and once completed the configuration information could be extracted from each station and assessed using the same mechanism.

The configuration files that are extracted from the devices or systems must be assessed, as suggested against a solution baseline in the form of a template consisting of one or more configuration files. It is therefore necessary to investigate a mechanism to achieve this. Software testing can use automation to test software systems and web sites by means of scripts using templates [16]. This technique might also be useful for testing the output of student work against a template, although without careful design the template may not allow sufficient flexibility in accounting for variations in possible solutions. If a large dataset, perhaps based on the outcomes of the marking of a large cohort of students undertaking the same exercise, were available then a more sophisticated system might use the principles of machine learning to facilitate the marking process. In this case a solution template could be refined by the process of supervised machine learning. This method is used widely in various fields, including education [17]. However this would depend on the output from a large number of students for a particular exercise across a wide range of outcomes [18]. The sample size available for the prototype on the other hand is relatively small, though this mechanism may be viable once a large dataset has been established. An alternative mechanism might involve an expert system. The successful deployment of a system or service might depend on a number of dependent commands or entries in a configuration file that can be pre-defined in an expert system [19]. In a medical expert system a certain diagnosis can be deduced based on a wide collection of data. In this case the diagnosis might be that a computer network is correctly or securely configured to a particular

standard. An expert system might take some time to setup. A large knowledge base is required that would need to cover a wide range of systems and services or multi-vendor network devices.

Whilst more sophisticated methods are possible a more straightforward solution presented itself for the prototype application. As we have seen a solution can be configured based on the requirements of a practical exercise and this can be used as a solution template. It should then be possible to compare the student output from this exercise against this template. The template may contain patterns that one might expect to see in device or system configuration files relating to a particular task. If we expect students to configure a firewall on a router, for example, we would expect to see certain command patterns in the router configuration file. However there may be variations in the information entered between different systems and these are variables that may differ between students. For example each system may have different IP addresses or firewall zone names chosen by students may vary. The names of variables could be specifically stated in the instructions for a practical exercise, but students may use a different one or simply misspell the name. One could argue that students should use the exact name specified, but if the system still works then they have still configured the system correctly. If we consider a firewall example, unless there is some flexibility in the application we would need to specify specific names for firewall zones in a practical exercise, but in enforcing the names we are suggesting part of the solution to the student in order to account for potential deficiencies in the application. If the application is more flexible we can simply ask students to "Configure a firewall on Router1". We would then expect the student to configure a firewall with zone names of their own choosing, which might improve their analytical skills. To improve flexibility the application could accept wildcards for variables like zone names. In the prototype this system might be satisfactory, though it might be difficult to see whether the student has applied the zone to the right interface. Wildcard values chosen in the new application prototype were three asterisks.

3. Test Methodology

3.1. Framework for Testing

One of the primary aims of the application is to assess the outcomes of practical exercises in a laboratory by establishing the extent to which laboratory objectives have been achieved. The prototype was given a working title of Network Assessor. A generic test mechanism was developed that was used to conduct tests. The methodology to facilitate testing of networking and security exercises was developed as follows:

i. Construct topology and apply basic configuration,
ii. Record baseline template from topology,
iii. Apply networking and security solutions to the topology,
iv. Record solution template,
v. 1st iteration: remove items from solution template that appear in baseline template.
vi. 2nd iteration: replace variables with wildcards,
vii. Test final template against solution template,
viii. Live iteration: test final template against 3rd party solutions

Once an initial configuration is complete (eg. if the IP addresses of devices are to be pre-configured) initial baseline templates are extracted from the devices in the topology. Solutions to the exercise are then applied to the relevant devices in the topology and once completed the solution baseline template extracted. This is compared with the initial template and any lines that appear in

the initial baseline either removed or commented out. It is important to ensure that only commands that are part of the exercise are included in the template files and uncommented. The modified solution template is then saved as the first iteration. Any variable names are then substituted for a wildcard in the form of three asterisks ("***") or removed from the end of a line so that students do not need to use exactly the same variable names as in the solution template. This is then saved as a second iteration of the solution template similar to figure 1. At this stage the new application is used to test the latest iteration of the solution template against the original solution and here one would expect an assessment rating of 100%. In the next step a third party should complete the exercise using different variable names to the original and the output of their work assessed by the application. If the user completed the exercise entirely correctly the result should still be 100%. The application records (by comparison with the final template) the number of correct lines in the solution files being assessed from a total number of lines, ignoring comments and blank lines. The application allows some lines to be weighted differently to account for variations in importance of some portions of the exercise. This is then converted to a percentage value, which will indicate the extent to which the network is correctly configured within the expectations of the exercise.

```
security passwords min-length 5

login block-for 60 attempts 2 within 30

enable secret

aaa new-model
!
aaa authentication login *** local
aaa authentication login default group radius local none

username admin12345 secret

#license boot module c2900 technology-package securityk9

crypto isakmp policy
 encr aes 256
 authentication pre-share
# group 5

crypto isakmp key *** address 10.20.20.1

crypto ipsec transform-set *** esp-sha-hmac
!
crypto map *** ipsec-isakmp
 set peer 10.20.20.1
 set transform-set
 match address

ip ssh version 2
ip domain-name security.com
```

Figure 1. Sample of Solution Template

3.2. Comparative Test

Initial tests, as previously reported [20], were performed on a network security exercise involving a small number of students. This exercise was run in the Cisco Packet Tracer simulator rather than real equipment since it was easy to reliably configure the necessary environment. Since Cisco Packet Tracer has its own in built assessment mechanism it is possible to compare the results generated by this with those from the new application. Due to the small sample size available in the initial network security exercise another one was chosen for this purpose, in which 25 students from a first year undergraduate computer networking module participated. The outcomes of the TCA were first measured using the inherent facilities of the Packet Tracer simulator. The exercise was then

assessed using the new application and the results compared. A typical network topology suitable for testing is shown in figure 2.

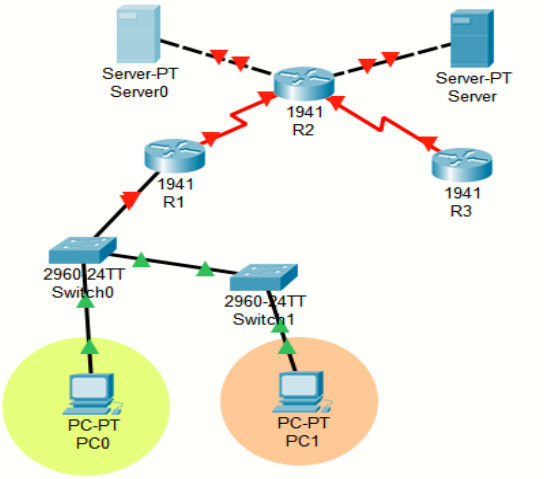

Figure 2. Typical network topology

4. Results

4.1. Background

The time constrained exercise required students to apply network and security configurations to each device in the Packet Tracer simulation topology. A solution was implemented and the saved configuration files from each network device extracted to produce the solution template files for the exercise. The remote script feature of the new application was used to run configuration management tasks. It was able to copy the necessary exercise environment from a server to the laboratory computers on which the simulation exercise would be run. Access permissions for the folder containing the exercise were disabled until the start of the TCA. During the exercise student folders were remotely monitored by a script to ensure that they had exported the device configurations into the correct folder. At the end of the exercise the application successfully extracted the configurations from each computer. Student work was manually checked as well as using the automated assessment features in the application. The assessment console is shown in figure 3. Packet Tracer also provided assessment results that could be used for comparison purposes. Instructions in the exercise were quite specific, which enabled a realistic comparison to be conducted. In the time constrained assignment exercise students were expected to configure tasks such as the IP addresses of router interfaces, routing protocols and the application of basic security.

4.2. Comparative Results

The results for each student from both Packet Tracer and the new application were recorded in a spreadsheet for comparison, with the results shown in table 1. Some corrections had to be made to the table after the exercise had been completed. It was noticed that at least 10 of the results from Packet Tracer showed exceptionally low marks in comparison to those from the new application. Column 2 in table 1 shows the original Packet Tracer scores for each student. When the Packet Tracer topology of each of these students was inspected it was found that those students had renamed either their routers or switches, or both, and in one case the servers too. The default device names given by Packet Tracer to routers is “Router0” and so on. Whilst students were expected to change the device

host names, for example to "R1", if they didn't also change the names in the topology to ones that Packet Tracer recognized it wasn't able to allocate marks to those devices. The problem was identified as being due to a conflict between the instructions given to students in the TCA and the initial network topology diagram. The solution involved manually changing the names to something that Packet Tracer recognized and recording the corrected mark as shown in column 3 of table 1.

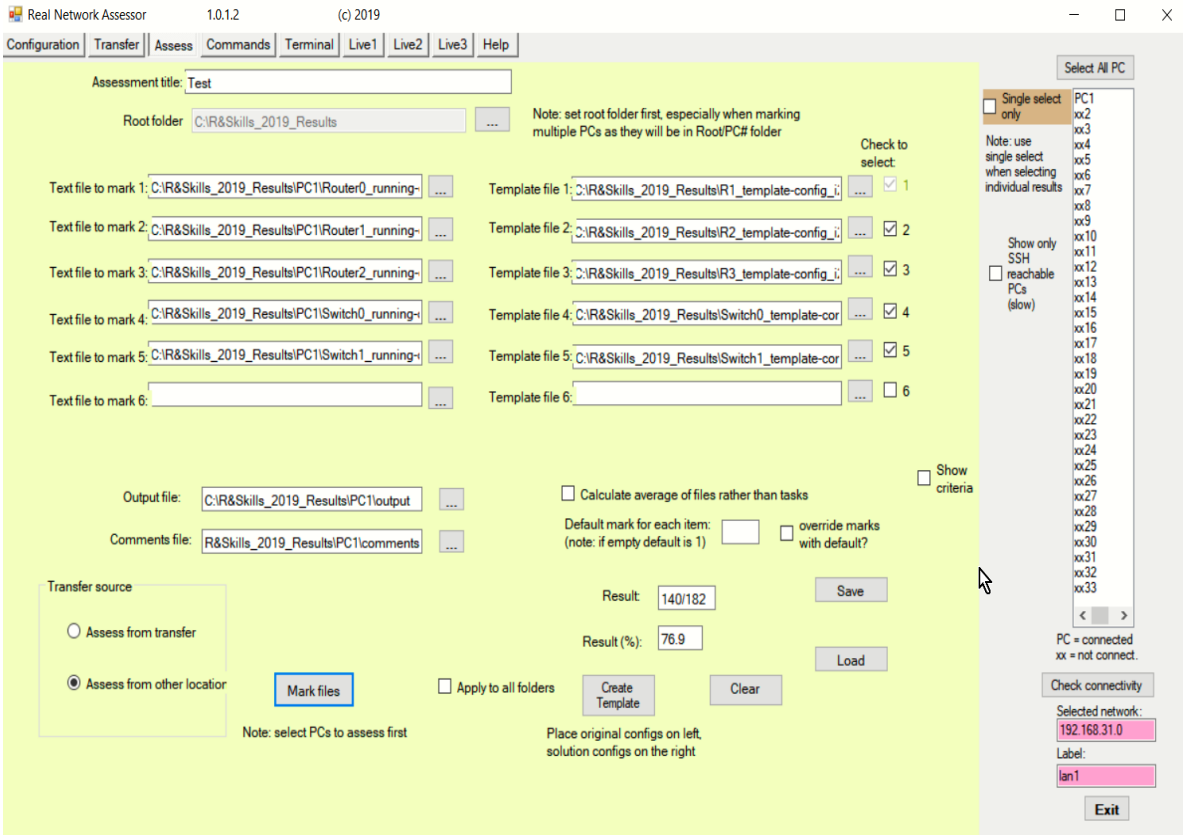

Figure 3. Assessment Console of the New Application

Since this experiment effectively piggy-backed on an existing exercise it wasn't designed with testing the new application in mind. Whilst it was important to configure the assessment mechanisms of each system (Packet Tracer and the new application) so that they assessed the same things it should be borne in mind that the mechanisms used by each are subtly different. In order for the new application to be able to assess the running configurations of network devices students had to export them to a file. The exercise also required students to undertake a few tasks on simulated servers, however there is no mechanism in Packet Tracer to export the server configurations to a file. Therefore, whilst Packet Tracer assessed the server configurations, the new application could not since the server configurations were not available to it. As it was not possible to remove the server tasks from the assessment prior to the TCA it was necessary to apply an adjustment to the results from Packet Tracer to remove marks allocated to the server tasks in order to ensure like for like comparison. In Packet Tracer 200 marks were available for the whole exercise, with server tasks contributing 15 marks (7.5% of the total marks available). For each student the number of marks that they had achieved in configuring the servers were recorded from Packet Tracer (see table 1 column 4) and then removed using a formula implemented in the spreadsheet:

$$pt\ score\ \% = \left(\frac{(p*2)-\left(\left(\frac{s}{15}\right)*15\right)}{185}\right) * 100 \quad (1)$$

where p is the Packet Tracer overall percentage score for a student (multiplied by 2 as the marks are from 200), s is their score for server configuration from 15 and the modified marks are from 185 once the server scores have been removed. This means that if a student scored 100% for the whole exercise once an adjustment is made their score would still be 100%. The final adjusted Packet Tracer results are given in table 1 column 5. The average difference from applying the adjustment was 1.4%, but in one case the score increased by 3.5% and in another case it decreased by 3.8% depending how much work they had done on the servers.

Table 1. Comparative Results

Student Number	Packet Tracer score %	PT corrected score %	Server score /15	PT score - server adjusted	Network Assessor score %	NA corrected score %	NA correction difference	Difference between PT & NA
1	3.5	3.5	4	1.6	22	17.9	4.1	16.3
2	1.5	1.5	0	1.6	8.8	4.1	4.7	2.4
3	4.5	7.5	1	7.6	31.3	27.7	3.6	20.2
4	3	30.5	9	28.1	40.7	37.6	3.1	9.5
5	25	33.3	12	29.5	45.6	42.8	2.8	13.3
6	34	34	12	30.3	44	41.1	2.9	10.8
7	15	38.5	11	35.7	47.8	45.1	2.7	9.4
8	41	41	9	39.5	53.3	50.9	2.4	11.4
9	39	39	0	42.2	53.8	51.4	2.4	9.2
10	16.5	49	11	47.0	58.8	56.7	2.1	9.6
11	10	49.5	1	53.0	58.2	56.0	2.2	3.1
12	59	59	15	55.7	56	53.7	2.3	-2.0
13	58	58	12	56.2	62.1	60.1	2.0	3.9
14	61	61	14	58.4	70.9	69.4	1.5	11.0
15	17.5	60.5	11	59.5	70.9	69.4	1.5	9.9
16	43	63	13	61.1	63.2	61.3	1.9	0.2
17	67.5	67.5	14	65.4	76.4	75.2	1.2	9.8
18	17	68	14	65.9	76.9	75.7	1.2	9.8
19	69	69	12	68.1	78.6	77.5	1.1	9.4
20	17	70	11	69.7	80.2	79.2	1.0	9.4
21	73.5	73.5	15	71.4	79.7	78.6	1.1	7.3
22	74	74	13	73.0	83.5	82.6	0.9	9.7
23	78	78	15	76.2	84.6	83.8	0.8	7.6
24	83	83	15	81.6	86.8	86.1	0.7	4.5
25	29.4	84.3	15	83.0	94	93.7	0.3	10.7
						Avg diff:	2.0	8.7

Although the same items were assessed in both systems the new application utilizes a template based on the configuration files from each network device (eg. the startup-config files) once a solution has been configured within the network topology. Items that were in the original configuration should be removed from the template files to leave only lines containing changes made as a result of applying the solution. Some lines will be treated as comments or may be commented out by the assessor and wild cards should be added as necessary. In the new application each uncommented line is treated as an assessed item. A problem was also noticed when running the new application in the exercise. Some lines that existed in the original configuration files were still present in the template files giving rise to false positives, meaning that all students were being awarded marks for something that they hadn't done. These related to commands that were present by default on active ethernet interfaces and the "login" command also present by default on console and telnet ports. This only involved 9 lines from a total of 182 in the template files and since it affected all students a spreadsheet formula could be used to apply a correction as follows:

$$\text{score } \% = \left(\frac{\left(\left(\frac{m}{100}\right)*182\right)-9}{173}\right) * 100 \quad (2)$$

where m is the original student percentage mark from the new application, 182 are the number of original items to be assessed, 9 items are to be removed, and 173 the new items from the corrected template. Alternatively the application could be run again against the updated template. The uncorrected scores are shown in table 1 column 6 and the corrected ones in column 7. The average difference between the original score and the corrected one (see table 1 column 8) was 2%, although in one case it was 4.7%, making most difference on the lowest scores. Whilst the line count was reduced it doesn't necessarily mean that the new application was assessing less items than Packet Tracer because they both use different assessment mechanisms. The template was nevertheless inspected to ensure that wherever possible it still assessed the same items as Packet Tracer.

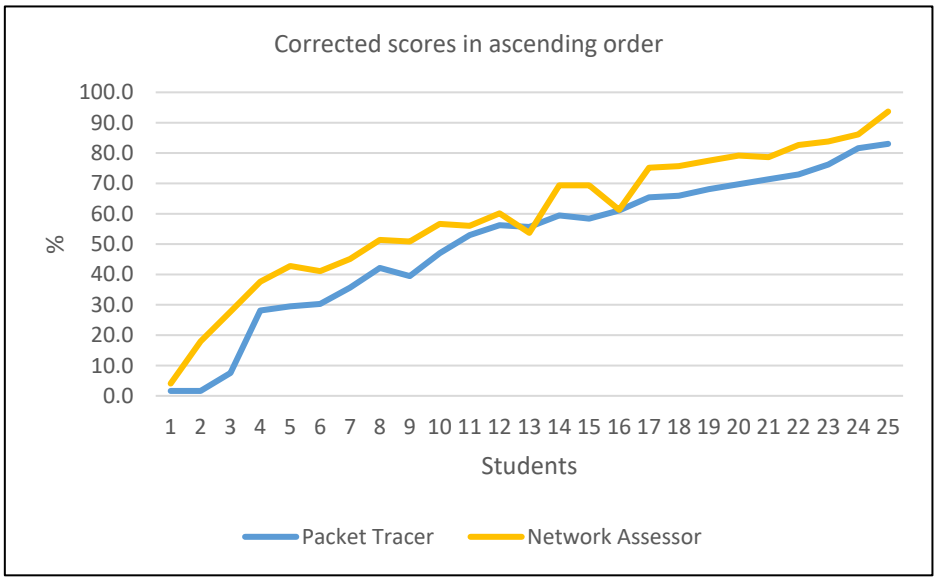

Figure 4. Comparison between scores for PT vs Network Assessor

After corrections were applied to both the new application and Packet tracer results it was possible to conduct a like for like comparison between the two. In most cases the new application gave a higher score compared to Packet Tracer even after the corrections as shown in figure 4 (see also table 1 column 9), but as suggested it wasn't thought that this was due to the reduced line count.

Only in one case did the new application score lower than Packet Tracer. When investigated it was found that the reason for this was because the new application had failed to mark the line relating to passive interfaces for the routing protocol on the routers. This can either be applied to individual interfaces, which is the way that the original solution was configured, or initially to the device as a whole, with Packet Tracer recognizing both methods using its command-based parser, which the new application did not. The biggest difference was a positive uplift of over 20% from the new application over Packet Tracer. It seems that the student had used some partial commands that the Packet Tracer parser had considered incomplete and invalid, however the new application does not use a command-based parser and found some items that could be awarded credit in isolation. Perhaps the student knew some commands, but not necessarily how to use them in a valid command sequence. A question might arise as to whether they should be awarded marks for applying partial commands that wouldn't allow device services to function, although the student in question didn't actually achieve a passing grade with either application. The average increase in mark awarded by

the new application was 8.7%. In some cases Packet Tracer had failed to award marks for something that could be considered at least partially correct, for example in assessing the application of sub-interface commands on a router as shown in the Packet Tracer output on the left hand side of figure 5, whereas the new application has awarded some marks for this (right hand side of figure 5).

On investigation the student had used a sub-interface of 0/0.21 instead of 0/1.21 as instructed, however this would still be valid and make no difference to the desired result. This phenomenon in which marks aren't awarded for an item that is correct can be considered a false negative. Both applications could have awarded marks for this, but only the new application has awarded marks for the two associated sub-commands, though Packet Tracer hasn't awarded any. It would appear that the student has some understanding of how to apply the commands and perhaps even shown evidence of independent thinking. If this is the case then the new application may have provided a fairer assessment. A similar marking issue also occurred because students had used different variable names to the ones given in the instructions. Packet Tracer only awarded marks for an exact match to statements, whereas with wild cards the new application allowed students to use variables that had simply been misspelled and even allowed them to use their own variable names. Examples of this included Access Control List (ACL) names and numbers that were specified in the instructions. This can be mitigated to some extent by the use of Packet Tracer's variable manager that can be used to seed a range of acceptable values, whereas the new application can easily accommodate instructions that might be less specific in demanding a particular spelling of a variable or ACL number. Instead of specifying "configure an access list named ACL-MGT" one could just instruct students to "configure an access list". At undergraduate level it might be considered good practice for assessment practices to encourage application of principles to practice rather than in specifying step by step instructions.

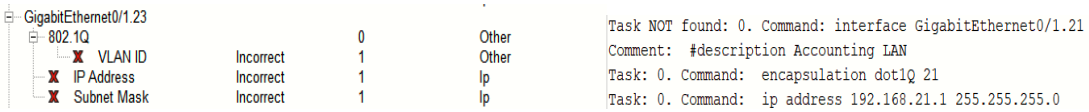

Figure 5. Packet Tracer output (left), Network Assessor output (right)

Packet Tracer is able to provide immediate feedback to students during the course of an exercise to indicate whether a task has been completed successfully. It will present the current overall progress, unless students have changed the device names to ones that it doesn't recognise, which as noted was an issue for a number of students who were initially presented with results well below expectations. On the other hand feedback from the new application was only available after the student files had been transferred to the server on which it was installed and the marking process run. The success of this necessitated ensuring that all students had exported their device configurations with the right filenames into the correct location. If they had failed to do this then the new application could not remotely retrieve their work. This involved pro-active monitoring and intervention if necessary. Nevertheless at the end of the exercise students could see the results from the new application. Assuming all student files were correctly named and located the process of transferring and assessing them took just a few seconds.

5. Conclusions and Recommendations

Testing has shown that the application is able to successfully and accurately assess security mechanisms applied to a computer network within a laboratory environment. The results were

validated by comparison to an existing proprietary system where the new application provided a close match to the scores provided by the other system, as seen in figure 4. Assessing all lines of output from every student would take many hours using manual means, but with automation the process can take a few seconds. This represents a huge saving in time, though the application does produce some false negatives and positives. One reason for this is that it is sometimes unaware of the context or inter-dependency of commands in a sequence because it doesn't use a command parser-based assessment mechanism. There may also be more than one way to apply solutions to a network and the application needs to be able to accept alternatives. If a command parser-based system were employed, since the application is intended to be a universal system independent of vendor or operating system, a range of command parsers would need to be used or developed [21]. The simplicity of the current system is that it compares a system under test, in this case the output of a time constrained assignment, with a known solution. The current parser mechanism is interchangeable and more sophisticated mechanisms for assessing work could be investigated, which might eventually lead to an expert system or other machine learning approaches specifically applied to networking and security education [17], [18], [19].

So far the tests have involved simulators rather than real devices and have not assessed operating system services. The application is intended to be vendor and system independent therefore tests will need to investigate a full range of physical devices and operating systems. The configuration management functions available within the application are able to deploy the environment for a practical exercise and extract the outcomes of this work. These features allow commands and scripts to be executed remotely using secure connections to computers and devices in the laboratory environment. Future enhancements might also enable live interrogation and assessment of these systems.

Whilst these tests investigated assessment of laboratory exercises the same principles could be applied to a real-world network where we may be trying to establish whether it is correctly configured or secure according to an expected baseline or security standard [22]. For example the application could be used to assess compliance as part of a testing framework for industrial network security [23].

References

[1] Omar Santos, "End-to-End Network Security: Defense-in-Depth". USA: Cisco Press, 2007.

[2] Lance Cleghorn, "Network Defense Methodology: A Comparison of Defense in Depth and Defense in Breadth", Journal of Information Security, vol. 4, (3), pp. 144-149, 2013. DOI: 10.4236/jis.2013.43017, Available: https://www.scirp.org/pdf/JIS_2013071213311297.pdf.

[3] Lyle D. Feisel and A. J. Rosa, "The Role of the Laboratory in Undergraduate Engineering Education," J Eng Educ, vol. 94, (1), pp. 121-130, 2005, Available: https://doi.org/10.1002/j.2168-9830.2005.tb00833.x.

[4] Simon Ball, Carolyn Bew and Sue Bloxham, et al., "A Marked Improvement - Transforming Assessment in Higher Education". UK: Higher Education Academy, 2012, Available: https://www.heacademy.ac.uk/system/files/A_Marked_Improvement.pdf.

[5] Neville Palmer, Warren Earle, Dr. Jomo Batola, "Automating the configuration management and assessment of practical outcomes in computer networking laboratories", in Advances in Intelligent Systems and Computing, Vol. 857, K. Arai, S. Kapoor, R. Bhatia, Ed. Switzerland: Springer, 2018, pp. 307-318, Available: https://link.springer.com/chapter/10.1007/978-3-030-01177-2_22.

[6] Sue Bloxham and Pete Boyd, "Developing Effective Assessment in Higher Education: A Practical Guide". (1st ed.) UK: Open University Press, 2007.

[7] Omar Santos and John Stuppi, "CCNA Security 210-260 Official Cert Guide". USA: Cisco Press, 2015.

[8] Cisco, "CCNA Security Lab Manual Version 2 (Lab Companion)". USA: Cisco Press, 2015.

[9] Cisco. "Cisco Packet Tracer Skills Assessments", 2010, Available: https://www.cisco.com/c/dam/en_us/training-events/netacad/course_catalog/docs/PT_AAG.pdf

[10] Jason C. Neumann, "The Book of GNS3: Build Virtual Network Labs using Cisco, Juniper, and More". USA: No Starch Press, 2015.

[11] Jeff. Geerling, "Ansible for DevOps". (1st ed.) USA: Midwestern Mac, 2015.

[12] Thomas Uphill, John Arundel and Neependra Khare, et al., "DevOps: Puppet, Docker, and Kubernetes". UK: Packt Publishing, 2017.

[13] Neville Palmer, "Work in progress - automation of a computer networking laboratory", in 2015 IEEE Global Engineering Education Conference (EDUCON), 2015, pp. 348-353, DOI: 10.1109/EDUCON.2015.7095995, Available: https://ieeexplore.ieee.org/document/7095995.

[14] Michael Heap, "Ansible from Beginner to Pro". USA: Apress, 2016.

[15] Neville Palmer, "Portable tool for assessing practical learning outcomes," in 2017 IEEE Global Engineering Education Conference (EDUCON), 2017, pp. 688-692, DOI: 10.1109/EDUCON.2017.7942921, Available: https://ieeexplore.ieee.org/document/7942921.

[16] Zhimin Zahn, "Selenium WebDriver Recipes in C#: Second Edition". (2nd ed.) USA: Apress, 2015.

[17] Ravinder Ahuja, Subhash C. Sharma and Maaruf Ali, "A Diabetic Disease Prediction Model Based on Classification Algorithms", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 44-52, Vol. 3, No. 3, 1st July 2019, DOI: 10.33166/AETiC.2019.03.005, Available: http://aetic.theiaer.org/archive/v3/v3n3/p5.html.

[18] Kevin Murphy and Francis Bach, "Machine Learning: A Probabilistic Perspective". USA: MIT Press, 2012.

[19] Peter Jackson, "Introduction to Expert Systems". (3rd ed.) Addison Wesley, 1999.

[20] Neville Palmer, "Automating the assessment of network security in higher education", in 2019 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Aug 2019, pp. 141-146, Available: https://ieeexplore.ieee.org/document/8941804.

[21] Cisco. "Programmability Configuration Guide, Cisco IOS XE Everest 16.6.x", 2019, Available: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/166/b_166_programmability_cg.html

[22] gov.uk. "Minimum Cyber Security Standard", 2018, Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/719067/25062018_Minimum_Cyber_Security_Standard_gov.uk__3_.pdf.

[23] Daniel DesRuisseaux, "Practical Overview of Implementing IEC 62443 Security Levels in Industrial Control Applications". USA: Schneider Electric, 2018, Available: https://download.schneider-electric.com/files?p_Doc_Ref=998-20186845.