Developing a capability to classify technical skill levels within a Cyber Range

William Aubrey Labuschagne, CSIR, Pretoria, South Africa [1]
Prof Marthie Grobler, CSIR, Pretoria, South Africa [2]

[1] wlabuschagne@csir.co.za
[2] mgrobler1@csir.co.za

Abstract: With the increase in technology adoption, quality assurance in terms of the technical skill level of cybersecurity experts working on a task is crucial. Educating employees and ensuring that they have the necessary tools and skills required to resolve cyber attacks against assets are essential. If a novice is assigned to resolve an attack on a critical asset, the attack may not be resolved as successfully and as timeously as when an expert is assigned to resolving the attack. Unfortunately, the classification of technical skill levels are often difficult to quantify and subject to personal opinion and experience. By developing a capability that allows for the assessing of technical skill level based on index similarity, it becomes feasible to more accurately classify the level of technical skills that an individual has. In testing the application of this assessment capability, an experimental test was designed where two test groups of participants skilled in cybersecurity took part in a challenge to resolve a simulated cyber attack against a specified asset. An analysis is done on the ways in which the various participants resolved the attack, considering amongst other metrics the time to resolution, the number of commands entered, and the similarity index to the optimal solution. Such a capability will contribute to correctly inventorying technical skills within an organisation. The benefit of knowing exactly what technical skills the cybersecurity experts have will result in the more timeous resolution of any cyber incidents within an organisation's domain. The focus of this paper will be the design and implementation of the experiment, and the analysis of the experimental results. The contribution of the paper will be a recommendation on the viability of such a classification capability pertaining to skillsets.

Keywords: Capability, Classification, Cyber Range, Experimental, Technical skills assessment

1. Introduction

Within the cyber domain there are a number of variables that can determine the expertise level of a cybersecurity expert. An individual may be regarded as an expert on a specific technical subdomain, whilst being regarded as a novice on another technical subdomain. These varying levels of expertise may present a skewed view of the person's overall technical abilities. Especially in a critical environment where fast and accurate reaction on a cyber incident is of utmost importance, organisations need to be sure that they have an accurately classified cybersecurity capable technical expert working on the problem at hand.

To ensure an adequate level of quality in terms of the technical skill level of cybersecurity experts working on a task is crucial. Educating employees and ensuring that they have the necessary tools and skills required to resolve cyber attacks against assets are essential. If a novice is assigned to resolve an attack on a critical asset, the attack may not be resolved as successfully and as timeously as when an expert is assigned to resolving the attack.

As such, a need has been identified for the capability to independently assess the technical skill level of an individual based on index similarity to an actual skilled sample answer sheet. This ensures that technical prowess is measured based on practical performance and not on pure theoretical knowledge. By developing a capability that allows for the assessing of technical skill level based on index similarity, it becomes possible to more accurately classify the level of technical skills that an individual possess. This paper presents the development of such a capability that will allow organisations to know exactly what technical skills the cybersecurity experts have; this will result in the more timeous resolution of any cyber incidents within an organisation's domain. The contribution of the paper will be a recommendation on the viability of such a classification capability pertaining to skillsets. Such a capability will contribute to correctly inventorying technical skills within an organisation.

2. Background

A cyber range is a virtual environment that is used for cyber training and technology development. It provides tools that help to strengthen the stability, security and performance of cyber infrastructures and systems used (Techopedia 2017). Ferguson and Tall describe the cyber range concept as a realistic environment to conduct and perform cyber security testing, training and rehearsal exercises (Ferguson, Tall & Olsen 2014). In other words, the use of such a platform would allow users to validate that learning has occurred and knowledge could be applied within an operational environment. For example, an incident response team could be trained how to respond to a cyber incident and verify that an incident response plan is effective.

In essence, a cyber range is a practice environment where cyber experts can be trained in terms of skills and capacity development, as well as specific system knowledge within a safe and controlled environment (Winter 2012). It provides an environment to practice various technical skills, including penetration testing, defending networks, hardening critical infrastructure and responding to attacks. It can further act as testbed for any network operations that needs to be performed, without the security concerns of performing this on an operational network, and serve as a research and development arena (Davis, Magrath 2013).

Although cyber ranges are generally used within the military context, it can also be implemented in government, academic and commercial domains. Some implementations of cyber ranges include the Simulator Training Exercise Network (SIMTEX) that revolves around the Cyber Flag concept of operations, the Cyber and Joint Effects Demonstration (CAAJED) that integrates a commercial war game simulator with a cyber inference model for cyber operations training, and the Virtual Cyber-Security Testing Capability (VCSTC), an automated testing capability to assess the security impact of a new device before deployment (Davis, Magrath 2013).

This paper addresses one implementation of a cyber range, was configured for incident response training. Currently, the classification of technical skill levels are often difficult to quantify and subject to personal opinion and experience. This is also often influenced largely by the technical expert's view or perception of his/her own technical prowess. This paper aims to address this problem by developing a capability for assessing technical skill level based on indexed similarity.

In testing the application of this assessment capability, an experimental test was designed where more than one cybersecurity expert participated in a challenge to resolve a simulated cyber attack against a specified asset. All their actions were recorded online, and analysed after completion of the experiment. The analysis pertains to the ways in which the various participants resolved the attack, considering amongst other metrics the time to resolution, the number of commands entered, and the similarity index to the optimal solution. The focus of this paper will be the design and implementation of the experiment, and the analysis of the experimental results. This capability is developed for inclusion into a larger cyber range. The end result of this experiment would provide a level of confidence in the solutions tested within a cyber range.

3. Design

The current design of the experiment provides the capability to configure the platform for various scenarios which include, but are not limited to performance testing, cyber exercises, risk assessments, user training, configuration validation and testing of incident response plans. These scenarios would allow for the integration of wireless and physical devices. As such, various hypothetical scenarios can be designed for assessing the skill level, based on the specific organisation's requirements. The high level architecture of the cyber range is depicted in Figure 1.

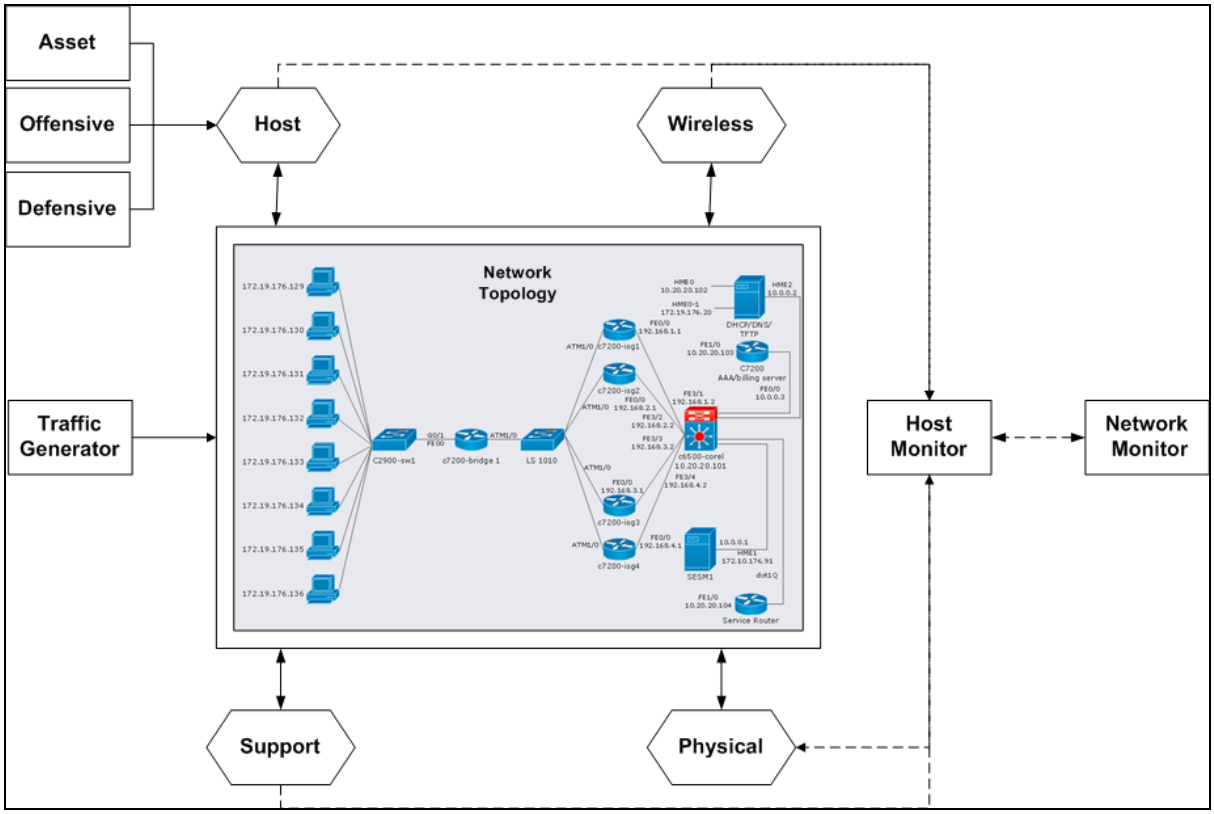


Figure 1: High Level Architecture of Cyber Range

The current design of the cyber range provides the following capabilities, as shown in Figure 1:

- Traffic generator - This capability is used to inject network traffic into the defined network. The network traffic within the cyber range needs to simulate an operational environment.
- Network topology - This capability provide a mechanism to create a network topology where the different machines would be allowed.
- Host - The host machines can be configured to have offensive and defensive capabilities or represent an asset. The offensive machines will be configured with tools to exploit other hosts. These machines will either be configured to be automated or used by a user, i.e. scripts will either be automatically executed or executed by the user. The incident response team or users required to participate in a scenario will have a base machine with no additional tools installed; users would need to install tools that are deemed necessary for the scenario. Machines that are the focus of the scenario testing are classified as assets. For example, a database server that needs to be load tested, a web server that needs to be hardened or the effects of a custom developed software package on a operating system. In a cyber exercise, the assets would be deemed valuable and the attackers (offensive hosts) would attempt to exploit the machines while the incident responders (defensive hosts) would protect them.
- Wireless - This capability interfaces the cyber range with wireless capabilities which include, but are not limited to mobile devices, Internet of Things (IoT) and access points.
- Support - Specific services would be configured and deployed as a support capability, for example log servers, time servers, repository servers to be able to update the operating systems, email servers, host intrusion detection machines, honeypots and file servers.
- Physical - Physical devices can be included in the scenarios; this implies having physical machines configured as a network to become part of the virtualized network.
- Host monitor - This capability collects metrics from the host machine, for example processes created, application installed, changes to the registry (if Microsoft Windows), files created, etc.
- Network monitor - This capability collects metrics from the host machine specific to networking on the host machine, for example data packets send to another host or network connections made. The

collection of metrics from the host monitoring and the network monitoring would be used to identify events and correlate effects.

4. Experiment

For this specific experiment, the cyber range was configured to enable a cyber exercise whereby an incident response team (defensive host) has to identify which asset within the network is under attack. The intention of the experiment was to enable the assessment of the participants' technical skills and initiate the development of a prediction system to identify intent. During the experiment, the response team machines will be targeted by attackers. Subsequently the individual responders also need to identify the type of attack and conduct attribution. This scenario required the enabling of the *host* and *support* capabilities of the cyber range. Offensive, defensive and asset machines (Host capability) were configured and deployed.

Kali Linux was used as the attacker (Offensive machine) (Linux 2015). Kali Linux is widely used within the cyber security community to perform various actions, including vulnerability identification, penetration testing and exploiting nodes on wireless and wired networks (Pritchett, De Smet 2013). A brute force attack was conducted using Hydra (Van Hauser 2017), and SlowHTTPTest (Shekyan 2016) was used to perform an Application Layer Denial of Service (DoS) attack. The DoS attack was conducted against the defined assets which were represented by Apache Web Servers, while the brute force attack was targeting the incident responders machines.

A Network File System (NFS) server was configured to store a file containing the authentication details of the targeted web servers. A folder was created and shared on the file server to provide the response team with access to the authentication information. The responders' machines were preconfigured base machines with the capability to capture commands entered by the end user (to facilitate analysis of their skill levels). Figure 2 presents the overall architecture of the experiment.

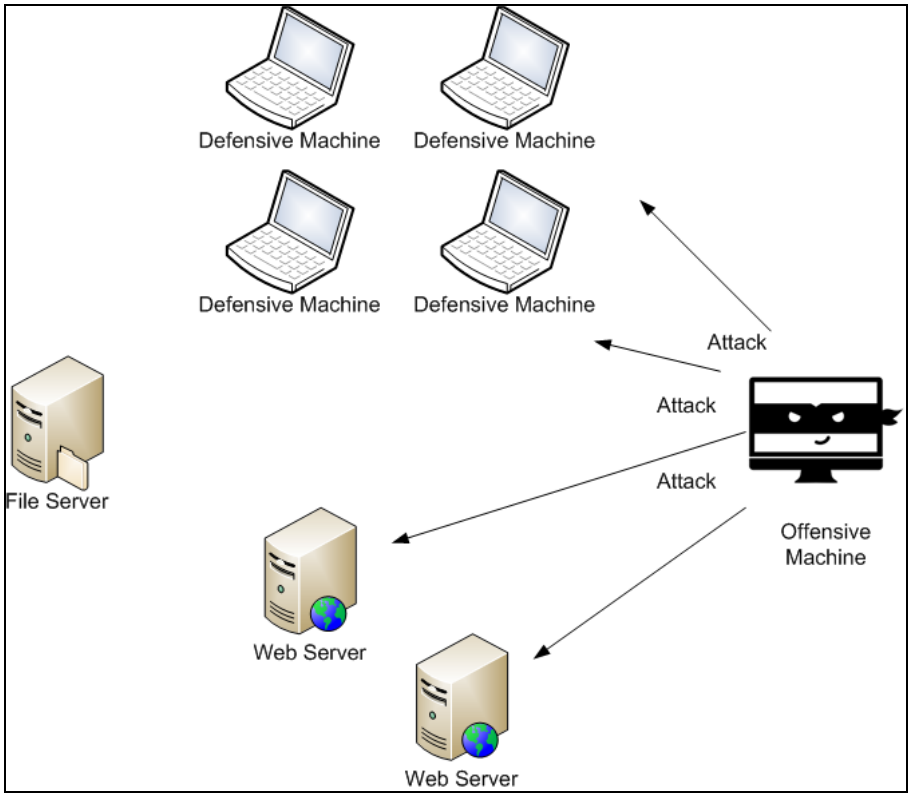


Figure 2: Experimental Design

The experiment makes use of quantitative and qualitative data. Two test groups were selected to comprise of four individuals each. The group that participated on 19 January 2017 consisted of four individuals with a

perceived higher level of technical expertise, whilst the group that participated on 20 January 2017 consisted of four individuals with a perceived lower level of technical expertise. Each session lasted two hours.

An important lesson learned through this experimental design is that all devices should have the same time configuration to ensure an optimal solution validation platform. This is essential in identifying cause and effect using temporal data. Metrics are extracted from each machine and stored at a centralised data storage, the timestamp and IP address forms part of the metrics. In the event whereby the machines have different timestamps it would skew results and invalidate observations.

5. Optimal Solution

The indexed solution process flow for this experiment is indicated in Figure 3. Assessment of the individual skill levels will be based on the similarity of their responses to this indexed process flow in solving the scenario.

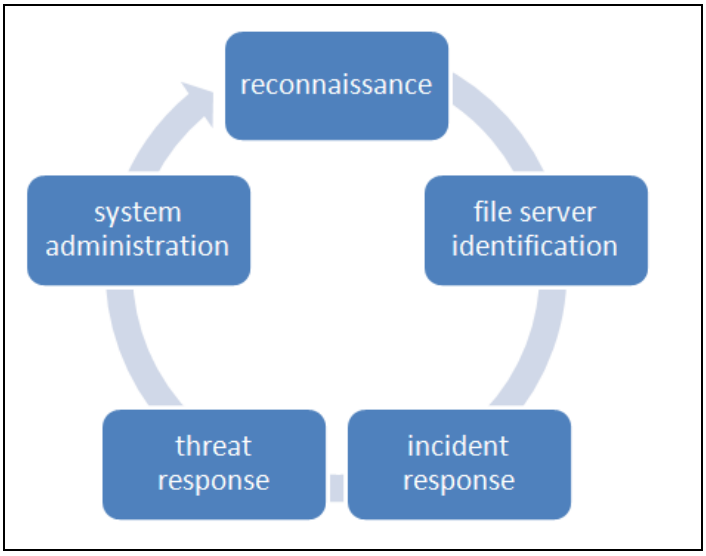


Figure 3: Objective Classification and Flow

First, the participants should conduct reconnaissance on the scenario. The solution to the scenario required the participants to perform a network scan to identify the relevant web servers and file server. Typically port 80 is assigned to web servers and port 2049 is assigned to NFS servers for network communication (Cotton et al. 2011). Ideally, participants should have performed five commands in order, in the reconnaissance portion of the experiment:

- ifconfig
- nmap (optional step)
- sudo apt-get install nmap
- sudo nmap -sT 10.0.5.2-30 > data.txt
- vi data.txt

After reconnaissance, the participants should identify the affected (targeted) servers on the network. The participants have to install a network mapping tool, such as nmap, and search for nodes with the characteristics of a web server or a NFS server. The results should highlight that Secure Shell (SSH) is open on the other nodes. The participants can use this to access the servers remotely. The participants will need authentication details in the form of a username and a password to access the node; this information is available on the file server. The participants further need to mount the share available once the file server is identified. After mounting the share, the participants will have access to the web server's credentials. The participants then need to find the web servers who would authenticate against the discovered credentials. Ideally, participants should have performed 11 commands in the file server identification portion of the experiment:

- showmount
- sudo apt-get install nfs-common
- showmount -e 10.0.5.18

- pwd
- mkdir mynfs
- sudo mount 10.0.5.18:/home/downloads mynfs/
- cd mynfs/
- ls
- vi detail.txt
- cd ..
- sudo umount mynfs/

Once the servers are identified, the participants have to respond appropriately to the incident. The participants will have to access the two web servers remotely and start conducting a search to identify which of the web servers are under attack. The number of web server processes initiated and analyses of the log files will be valuable in resolving this task. The participants could also utilise tools providing information about the web server to identify and confirm additional information to support the notion that a particular web server is under attack. Once the web server under attack has been identified, the participants can open the website on another machine's web browser to confirm that a denial of service attack is being conducted (the affected web site would have a very slow response to a request). Ideally, participants should have performed six commands in the incident response portion of the experiment:

- vi data.txt
- sudo ssh roque@10.0.5.22
- netstat -ntlp | grep LISTEN
- sudo ssh roque@10.0.5.25
- netstat -ntlp | grep LISTEN
- ps faux | grep apache2 | wc -l

Participants also have to address and respond to the active threat by the attacker. The specific attack aims to determine the username and password combination to access the secure shell (SSH) on the target machine (the participant's machine). The responders have to be vigilant and analyse their own system for possible compromise or potential attack. This is done through the analysis of the log files and network connections created on the machine. The network connection analysis would have highlighted connections repeatedly been made from one machine on the network. Ideally, participants should have performed two commands in the threat response portion of the experiment:

- ps faux | grep sshd | wc -l
- sudo tail -f /var/log/auth.log

Finally, the participants have to conduct basic system administration on the machines that were assigned to them. This consists of verifying, configuring, uninstalling and installing specific services. Ideally, participants should have performed five commands in the system administration portion of the experiment:

- sudo dpkg -s nis
- sudo apt-get purge nis
- sudo sysctl -w net.ipv4.ip_forward=1
- sudo dpkg -s auditd
- sudo apt-get install auditd

6. Results

To facilitate the comparison of the two sessions, four usernames were assigned to the four participants in each group. Thus, there were two participants with the same username, one in each test group. The participants were instructed to obtain various objectives. These objectives were classified together as reconnaissance, file server identification, incident response, threat response and the system administration, based on the optimal solution process flow. The users' interaction generated data associated with each of these objectives. All commands entered by the respective participants were logged. To facilitate the analysis of the commands logged, a number of metrics will be applied.

6.1 Experiment Metrics

In comparing the test groups and participants, and developing a capability for assessing technical skill level based on indexed similarity, the participants were ranked based on the commands used to achieve the specified objectives. The textual commands were entered and executed by the participant resulting in data points which could be analysed. The list of metrics used for analysis is captured in Table 1.

Table 1: Metric Description

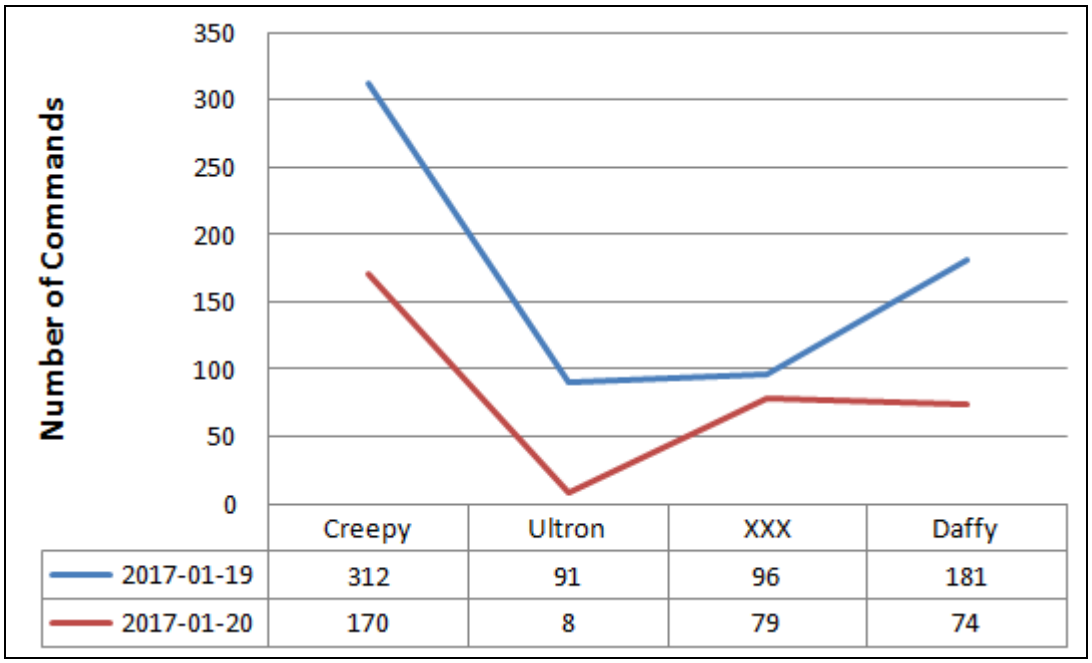
Abbreviation	Metric	Description
TCT	Total Completion Time	Entire time in seconds taken by the participant to complete the exercise.
TKS	Total Commands Entered	Total commands entered by the participant.
NM	Time Before Nmap	Determine the time before nmap was used.
RC	Reconnaissance Commands	Total number of commands executed to conduct reconnaissance of the network.
RCS	Reconnaissance Similarity	Average of similarity index on most accurate command executed.
FC	File Commands	Total number of commands executed to identify and connect to file server.
FCS	File Server Identification Similarity	Average of similarity index on most accurate command executed.
IC	Incident Commands	Total number of commands executed to identify asset under attack and connect.
ICS	Incident Similarity	Average of similarity index on most accurate command executed.
TC	Threat Commands	Total number of commands executed to identify attack on own system.
TCS	Threat Similarity	Average of similarity index on most accurate command executed.
AC	System Administration Commands	Total number of commands executed to conduct system administration as per requirement.
ACS	System Administration Similarity	Average of similarity index on most accurate command executed.
TT	Total	Calculation of total based on similarity indexes (RCS + FCS + ICS + TCS + ACS).

6.2 Metrics Discussion

The total number of commands (TKS) metric determined the total number of commands entered by the participant during the experiment. According to the optimal solution presented, all five steps in the lifecycle could have been completed in 29 steps. Keeping in mind that this is an ideal solution and that the experiment administrator has full knowledge of what the experiment is about, i.e. what the attack type is, where to locate the file server and login details, etc. The hypothesis is that an advanced user would have a lower count of

commands entered (equal to or higher than 29) due to preexisting knowledge to achieve the objective. This metric measures only the commands entered in the command shell. Participants were allowed to make use of the Internet via a search engine to look for guidance on how to approach and address the scenario. These activities are not included into the total number of commands entered by the participants.

Figure 4 presents a visual display of the total number of commands entered by each participant. In the technically more advanced test group, the number of commands entered vary between 91 (*ultron* username) and 312 (*creepy* username). In the technically less advanced test group, the number of commands entered vary between 8 (*ultron* username) and 170 (*creepy* username).



	Creepy	Ultron	XXX	Daffy
2017-01-19	312	91	96	181
2017-01-20	170	8	79	74

Figure 4: Number of Commands Entered by Participants

The total time from the first to the last command was initially calculated (TCT). The maximum time for the experiment was 2 hours (7200 seconds). The first time a participant executed the command to conduct reconnaissance was measured by the Time Before Nmap (NM) metric.

The reconnaissance metric (RCS) was calculated by determining which command was the most similar to the recommended command. This metric analysed whether the participant determined the device network configuration and installed a network mapping tool in order to conduct reconnaissance. All commands that were relevant to the expected action were collected and individually tested, and the most accurate commands were selected. The average of the similarity test for both expected actions was used as the final allocation for the reconnaissance metric. The similarity test was achieved by using cosine similarity testing whereby an index value closer to 1 resembles a match (Mihalcea, Courtney & Strapparava 2006).

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 5: Cosine Similarity Testing (Source: (Wikipedia 2017))

For example, the participant entered three commands: “nmap”, “apt-get update nmap” and “install nmap”. The most recommended command is “sudo apt-get update nmap”. The second command “apt-get update nmap” is selected because it has the highest similarity index (See Table 2).

Table 2: Example of Similarity Testing

Number	Command Entered	Recommended Command	Similarity Index
1	nmap	sudo apt-get update nmap	0.29
2	apt-get update nmap	sudo apt-get update nmap	0.88
3	install nmap	sudo apt-get update nmap	0.44

The identification of the file server (FCS) metric was calculated by determining if the participant installed the required tools and then mounted the share of the file server. The incident metric (ICS) determined if the participant accessed the web servers with credentials obtained from the file server. The threat response metric (TCS) determined if the participant identified the number of processes opened to specific ports before reviewing the log files. Lastly, the system administration metric (ACS) required the installation of auditd, enabling Internet Protocol (IP) forwarding and uninstalling Network Information Services (NIS). The incident, threat response, administration metrics were all averaged and similarity testing conducted on each respectively, similar to the reconnaissance metric (RCS) similarity test. All the metrics were then summed to form the objective metrics.

The final assessment metric (TT) for each participant was calculated by summing the objective metrics (consisting of the RCS, FCS, ICS, TCS and ACS). This was added to the result of the division of the number of most optimal commands (29) by the total commands entered (TKS) per participant, and then multiplied with the objectives metric. The use of the total commands as part of the equation is important as more commands entered do not necessarily indicate a higher skillset. The equation used is depicted in Figure 6, followed by Figure 7 presenting the results of applying the equation.

$$AssessmentMetric(TT) = ObjectiveMetrics + \left(\frac{OptimalNumberofCommands}{NumberofCommands(TKS)}\right) * ObjectiveMetrics$$

Figure 6: Equation for Assessment Metric

Name	TCT	TKS	NM	RC	RCS	FC	FCS	IC	ICS	TC	TCS	AC	ACS	TT
creepy(2017-01-19)	5776	312	258	73	0.96	66	0.86	8	0.87	0	0.0	2	0.54	1.78
xxx(2017-01-19)	5306	96	0	33	0.96	40	0.92	5	0.77	7	0.44	6	0.82	2.56
daffy(2017-01-19)	5813	181	789	67	0.92	24	0.92	6	0.92	8	0.41	7	0.67	2.21
ultron(2017-01-19)	3523	91	155	20	0.96	30	0.91	10	0.82	2	0.41	6	0.47	2.36
creepy(2017-01-20)	5246	170	1841	48	0.89	27	0.8	0	0.0	0	0.0	8	0.55	1.35
xxx(2017-01-20)	5193	79	1703	13	0.49	15	0.75	0	0.0	1	0.42	0	0.0	1.47
daffy(2017-01-20)	6308	74	1934	8	0.92	45	0.93	4	0.71	4	0.31	5	0.29	2.18
ultron(2017-01-20)	5221	8	2298	4	0.49	0	0.0	0	0.0	0	0.0	0	0.0	0.49

Figure 7: Result Set for Incident Response Exercise

7. Data Analysis

The final result was plotted using a heatmap (Bojko 2009) and ordered by the final assessment metric (TT) in descending order (Figure 8). In other words, the participant that completed the majority of objectives with the most accurate commands issued would be placed on top. The heatmap consists of cells with a gradient of colour to represent the value. The darker the colour the higher the value; in this case the participant “xxx” from the group of 19 January 2017. The participant with the lowest assessment metric is placed at the bottom of the heatmap; in this case the participant “ultron” from the group of 20 January 2017.

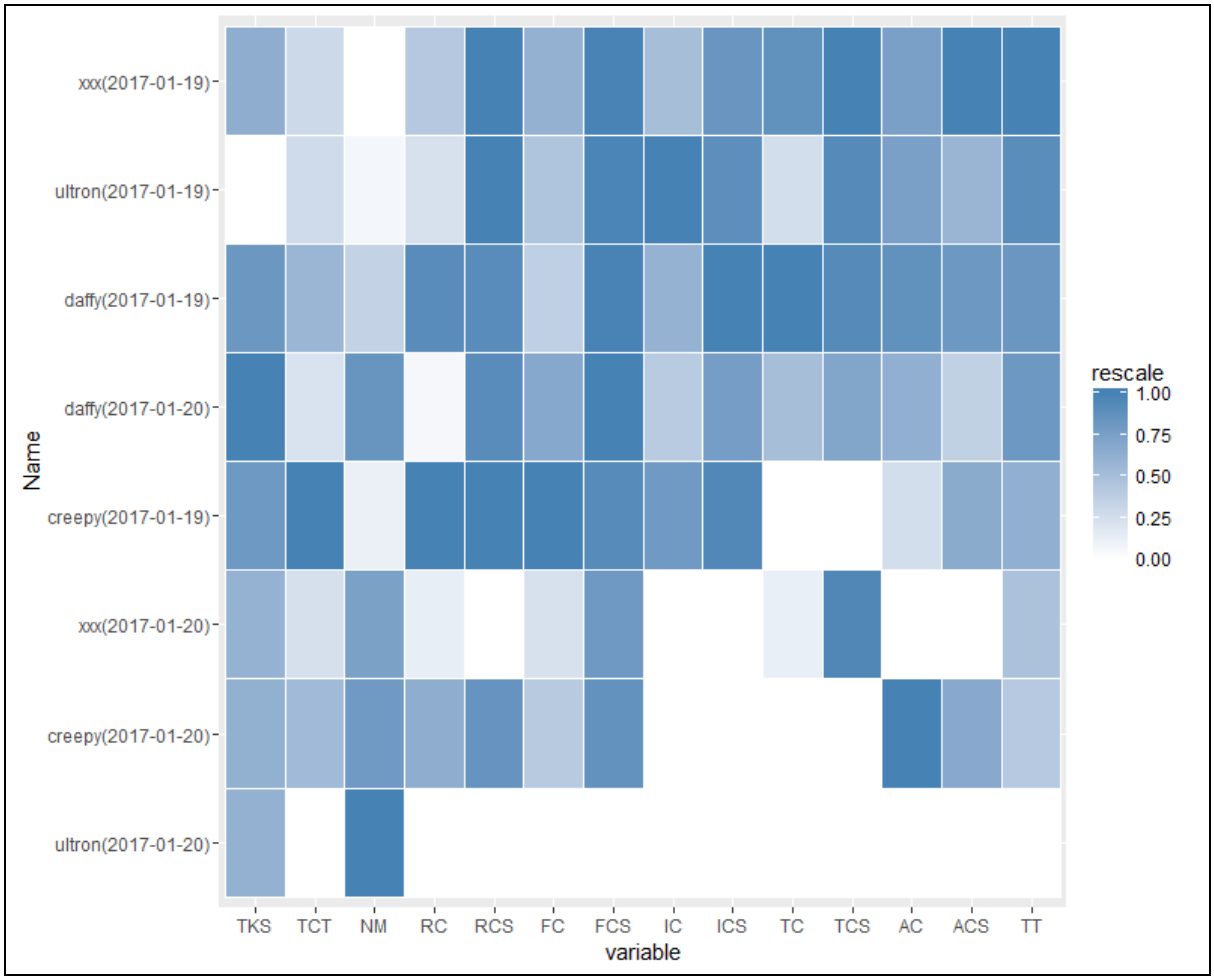


Figure 8: Heatmap of Skills Assessment

The level of each participant's practical knowledge could be inferred from these results. The hypothesis was that the technical stronger participants would receive higher assessments. The participants were grouped based on perceived skillsets. The stronger group, except for one novice participant, conducted the incident response exercise on 19 January 2017, while the more novice users were grouped together on 20 January 2017. One participant in the 20 January 2017 group had a higher technical background.

The objectives that the participants were experiencing difficulty with are clearly highlighted by the number of attempts made to achieve the objective. This is typical with a "trial and error" approach whereby commands are entered until the desired result is achieved. The results also indicate that more advanced participants would use less commands to achieve an objective. The number of commands therefore is an important metric to consider when assessing a technical skillset (for example, script kiddies can launch an attack by executing a single command). In most cases, the default command will be used, although advanced users would understand the tool's capabilities better and thus may provide additional parameters when executing commands. The complexity of the command will also need to be considered together with subsequently commands.

Another important result is the possibility to classify actions performed by the participants. This is an important capability to develop as actions could be automatically deducted using the clustering of commands. For example, the combination of "*ifconfig*","*sudo apt-get install nmap*" and "*sudo nmap -sT targetIP*" together could be classified as reconnaissance. By using a predictive capability deployed within a host machine used by an end user, it would be possible to determine what the intention of the user is.

From the heatmap, the time to start the process to conduct reconnaissance was also determined as a metric to identify which participant took the longest. This could be used to assess if there is a correlation between the time to start reconnaissance and the final assessment value. The first observation was that the participant who obtained the highest total assessment value also took the shortest time to initiate the reconnaissance. The participant who scored the lowest overall took the longest to initiate the reconnaissance.

As such, a need has been identified for the capability to independently assess the technical skill level of an individual based on index similarity to an actual skilled sample answer sheet. This ensures that technical prowess is measured based on practical performance and not on pure theoretical knowledge. By developing a capability that allows for the assessing of technical skill level based on index similarity, it becomes possible to more accurately classify the level of technical skills that an individual possess. This paper presents the development of such a capability that will allow organisations to know exactly what technical skills the cybersecurity experts have; this will result in the more timeous resolution of any cyber incidents within an organisation's domain. The contribution of the paper will be a recommendation on the viability of such a classification capability pertaining to skillsets. Such a capability will contribute to correctly inventorying technical skills within an organisation.

8. Conclusion

The use of cyber ranges is essential in validating solutions before deployment within an operational environment. This could also be used to perform training of solutions and knowledge assessment. The design and development of a configurable cyber range is essential to not only ensure that future technologies could be incorporated, but also be technology independent.

This paper discussed the use of a cyber range for skills assessment by having participants respond to a hypothetical cyber incident. Data metrics were collected as the participants interacted with their devices to achieve the set objectives. Specific to this environment, only textual commands were collected. As such, the hypothesis of using text to identify intent and application of knowledge was tested with this incident exercise.

The analysed data clearly showed a positive result for assessing skills within a practical environment. Furthermore, the basis for developing an intent capability whereby the intent of the user can be predicted is shown to be possible. However, within a command based environment many different approaches could be followed by the user.

References

Bojko, A.A. 2009, "Informative or misleading? Heatmaps deconstructed", International Conference on Human-Computer Interaction, Springer, pp. 30.

Cotton, M., Eggert, L., Touch, J., Westerlund, M. & Cheshire, S. 2011, Internet assigned numbers authority (IANA) procedures for the management of the service name and transport protocol port number registry.

Davis, J. & Magrath, S. 2013, A survey of cyber ranges and testbeds, Cyber Electronic Warfare Division, Edinburgh, Australia.

Ferguson, B., Tall, A. & Olsen, D. 2014, "National Cyber Range Overview", 2014 IEEE Military Communications Conference, Oct, pp. 123.

Linux, K. 2015, Kali Linux| penetration testing and ethical hacking linux distribution.

Mihalcea, R., Courtney, C. & Strapparava, C. 2006, "Corpus-based and knowledge-based measures of text semantic similarity", Proceedings of the 21st national conference on Artificial Intelligence Association for Computing Machinery, New York, United States of America, July 16 - 20, 2006, pp. 775.

Pritchett, W.L. & De Smet, D. 2013, Kali Linux Cookbook, Packt Publishing Ltd.

Shekyan, S. 2016, SlowHTTPTest, 1.7th edn, github.com, California, United States of America.

Techopedia, 2017, *Cyber Range* [Homepage of Techopedia], [Online]. Available: https://www.techopedia.com/definition/28613/cyber-range [2017, 02/02].

Van Hauser, R. 2017, THC-Hydra, 8.4th edn, github.com, California, United States of America.

Wikipedia 2017, 2 February-last update, Cosine similarity [Homepage of Wikipedia], [Online]. Available: https://en.wikipedia.org/wiki/Cosine_similarity [2017, 20/10].

Winter, H. 2012, "System security assessment using a cyber range", 7th IET International Conference on System Safety, incorporating the Cyber Security Conference 2012, Oct, pp. 1.