Evaluating and Identifying Effective Detection Methods for Purple Team Engagements Against RDP Honeypots

Christopher Di-Nozzi
School of Design and Informatics
Abertay University
DUNDEE, DD1 1HG, UK

ABSTRACT

Context: Security experts partaking in offensive engagements need to be able to detect whether the system they are attacking is legitimate or a honeypot. Various methods need to be identified and evaluated to determine which of these is the most suitable.

Aim: To evaluate and identify the most effective detection methods for purple team engagements against contemporary RDP honeypots.

Method: Due to a lack of research within this field, a methodology will be adapted from related research. Each detection technique learned from research, or identified on the individual honeypot, will be tried against legitimate services and their honeypot counterparts. Using a range of both legitimate services and honeypots will provide the most extensive data. The research will also investigate how any vulnerabilities found could be fixed.

Results: The data gained from these experiments will be analysed for its reliability, speed, and deception, and these factors will be used to determine which technique is most effective. These factors were chosen as they are the most important factors during an offensive engagement like a purple teaming exercise. Improvements that could be made to the honeypots to avoid detection will also be included.

Conclusion: If successful, this project will demonstrate how RDP honeypots can be identified from their legitimate counterparts in the most effective way for an offensive engagement.

Keywords: Remote Desktop Protocol, RDP, honeypots, detection, fingerprinting, purple team, offensive engagement, network analysis, open-source.

1. INTRODUCTION

The Remote Desktop Protocol (RDP) is a fundamental aspect to remote administration of computer systems around the globe. RDP allows IT professionals to access and administrate systems from anywhere with an internet connection. It is, therefore, of interest to cyber criminals due to the level of access it can provide. Recently, there has been an increased interest in RDP software due to the COVID-19 pandemic forcing many to work from home. TrustRadius, a software review platform, saw an increase of 1,587% in average traffic to their remote desktop category pages, as seen in Figure 1, (Sullivan-Hasson, 2020).

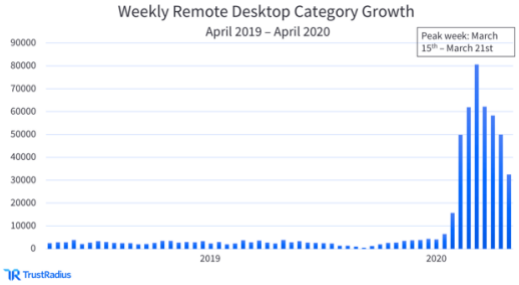

Figure 1: Weekly average traffic to the Remote Desktop category on Trust Radius, a software review service.

However, RDP has seen a reasonable amount of security vulnerabilities, including the infamous remote code execution vulnerability, BlueKeep (CVE-2019-0708), that was used to spread malware across the globe, (Greenberg, 2019). RDP is also a regular victim of password brute force attacks. A report by Sophos, in which 10 RDP honeypots were deployed, found that an internet-facing RDP service would receive an average of 600 brute force attempts per hour, compared to 2 per hour from a similar study conducted in 2012 (Boddy, et al., 2019). With the rise of RDP as a target for cybercriminals there is an inevitable rise in the need for research and monitoring. Honeypots fill this role well as they can be used both to learn about current attack trends and techniques, as well as alert an organisation of potential attacks. Honeypots are applications that mimic an application or network protocol, in this case an RDP service.

While some research has been carried out on honeypot fingerprinting, there has been no research into fingerprinting RDP honeypots specifically or evaluating the most effective methodology for finger printing honeypots in a long-term, offensive engagement. This information is essential since a honeypot is only fit for purpose when it cannot easily be distinguished from the genuine service it is mimicking. Knowing how to effectively fingerprint honeypots can improve the application itself and the work of those participating in ethical offensive engagements.

This project aims to evaluate and identify the most effective detection methods for purple team engagements against contemporary RDP honeypots. This aim can be broken down into three different research questions:

1. What are the existing honeypot detection methods, and how do they compare?

2. Which detection technique is most effective in a purple team engagement?

3. How can these detection methods, honeypots, and the broader network be refined to improve effectiveness?

This proposal will continue as follows: A Background section that will critically examine relevant material to set the context of this project. This will be built on into a Methodology that will outline an approach to probe different RDP implementations. Finally, a Summary will conclude the proposal by exploring the consequences of this research and the value it could pose.

2. BACKGROUND

2.1 Honeypots

Honeypots are devices placed onto networks to lure cybercriminals into attacking them. The actions of the attacker can then be monitored and researched. They exist for many different technologies but are most seen impersonating SSH, HTTP and TELNET protocols, all with varying levels of interaction. They can be classed into three different levels, (Livshitz, 2019).

1. Low-Interaction: The most basic form, usually only emulating the service fingerprint and a login ability. They are the easiest to deploy and configure but offer the least amount of information.
2. Medium-Interaction: Responds to specific set criteria but do not fully emulate the service, e.g., a vulnerable file path on a web server being served to bait a particular exploit, (Spitzner, 2002).
3. High-Interaction: Emulates the service and an entire system to back it. This could include a filesystem, desktop environment or an entire network of honeypots. They are the most time and resource consuming to deploy but offer the most amount of data.

Research shows that honeypots are most commonly low interaction, due to the ease of development, deployment, and maintenance (Nawrocki, et al., 2016). However, due to the advanced capabilities of RDP, medium- and high-interaction honeypots are significantly more helpful due to the extensive behavioural data they can provide. Since medium- and high-interaction honeypots are the only appropriate type for RDP, few honeypot solutions exist.

2.2 Remote Desktop Protocol (RDP)

The Remote Desktop Protocol is a network protocol developed by Microsoft to allow users to connect to their Windows systems over a network using a graphical interface. By default, an RDP server runs on port 3389, (Liang, et al., 2021). RDP offers the whole desktop experience, which makes it a valuable technology to system administrators and remote workers. However, this poses a challenge for creating high-interaction honeypots to replicate the service. The typical approach to this issue is using virtual machines that run legitimate copies of Windows software with RDP enabled. These machines are then rebooted once attacked, or a different machine from a pool of machines is given out to different attackers.

2.3 Honeypot detection methods

There has been little research done into honeypot detection methods and no research into RDP honeypots particularly. This is most likely due to how much detection techniques can vary in the service they mimic and their level of interaction, making it hard to compile general methods. However, information can be gathered from some related research. A paper by Michail Tsikerdekis examined various improvements that could be made, both practical and theoretical, to honeypots behaviour to decrease their chance of being detected, (Tsikerdekis, et al., 2018). By reverse-engineering this list, a handful of methods for detecting honeypots can be attained. This includes automatic redeployment, delays, hardware, and dynamic intelligence.

Automatic redeployment is a technique used by honeypots to reconfigure their system to a default state after being attacked, allowing for another attacker to be trapped by it. However, this can also be an obvious tell. For example, if an attacker connects to a honeypot, creates a file, and then disconnects, they expect to find the file on the system if they ever reconnect. If the honeypot rebuilds itself immediately after the attacker disconnects, the attacker could use the removal of their file as proof the machine has been tampered with by reconnecting within a short time frame.

The delays method involves examining how long it takes on average to connect to the service versus the time taken to connect to the honeypot. Honeypots often take slightly longer to finalise a connection due to extra information being logged by the server that the service would not ordinarily log. This method is general enough that it could be applied to honeypots of any type, including RDP.

The third technique, hardware, would examine if the service is running on bare metal or a virtual machine (VM). This information is helpful, however, it cannot be used on its own to determine if a machine is a honeypot or not. While it is common for honeypots to be run inside VMs, due to the flexibility and ease they offer, it is also common for legitimate infrastructure to be run in VMs for the same reason. This is an even more popular option with the rise of cloud computing.

Dynamic intelligence is a method honeypots use to attempt to be more helpful. It involves using AI to adapt to the user and lure them in by doing things such as changing file names to different languages to determine ethnic background. While this technique is intended to make honeypots harder to detect, an attacker with a keen eye could notice the change in file names. RDP would be a sensible service to implement dynamic intelligence due to its high level of interaction, therefore, this detection technique is also likely applicable to either existing or future RDP honeypots. Not all these techniques will be applicable to RDP services, however, they set a standard of detection techniques that can be used within the methodology.

More novel detection techniques have also been proposed. Alexander Vetteral presented a fingerprinting technique that involved analysing the implementation of network architecture within off-the-shelf honeypots, (Vetterl & Clayton, 2018). This technique involved examining how honeypots implemented network protocols and comparing them to the services they mimic. Although the differences were slight, they were enough to allow distinctions to be made between the honeypots and authentic services. The technique involved generating many packets and a moderate amount of low-level research. For these reasons, it is not an excellent "on-the-fly" method, however, with enough time and the right conditions, it could be implemented into a bespoke tool to be used in offensive engagements.

3. METHODOLOGY

3.1 Research

The first chapter of the project will contain a literature review that will critically analyse the small pool of existing research into honeypot detection methods.

Since this current area of research is lacking, each honeypot will have its behaviour examined to find flaws or inconsistencies within its design. These would then be used to

develop more detection techniques or could be used as markers to denote likelihood.

3.2 Development

The information gained from the literature review will be used as a starting point for detection techniques. These techniques will be used against a variety of legitimate RDP servers and RDP honeypots. For example, a well-known existing technique is examining the time it takes to connect to a legitimate service verses a honeypot. To research this, a number of connections would be made to all the services, and the time taken to connect would be calculated using a packet inspection tool such as Wireshark. The times would then be compared to identify any substantial differences.

Using a range of different software's will yield more conclusive results as to the more effective overall technique, rather than a technique that only works to identify one legitimate service from one honeypot. A table of software to be used can be found in Table 1.

Table 1: The legitimate and honeypot RDP server that will be used for research.

Legitimate Service	Honeypot
Windows RDP Server	ad-honeypot-autodeploy
XRDP	rdppot
	rdpy

All the RDP servers will be run on VMs. This both limits time and costs, while still reflecting a realistic scenario. The default RDP server offered by Microsoft will be run on a copy of Windows Server 2022, while XRDP will be run on an appropriate Linux system. All the honeypots will be deployed on separate VMs running Linux. Every system will be left in its default state and will be run on the standard port of 3389. A final Linux VM will be used to simulate the attacker. Amongst other tools, this machine will use FreeRDP, an open-source RDP client, to interact with the servers. A network diagram of the virtual network that will be used within testing can be seen below.

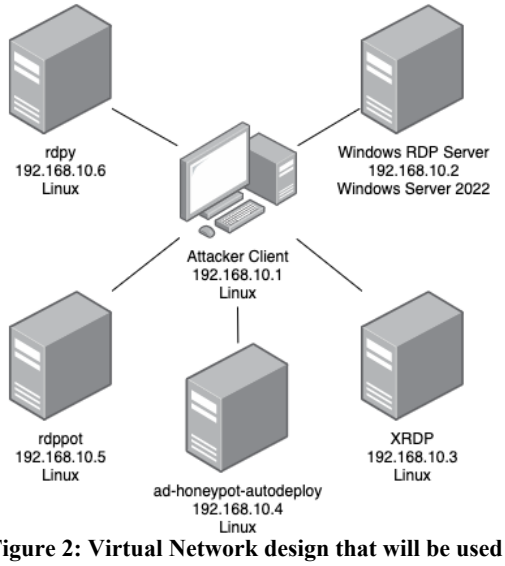

Figure 2: Virtual Network design that will be used during research

3.3 Evaluation

The results found during testing will be used to evaluate the most effective technique for identifying RDP honeypots. This will be based on the speed of determination, the amount of noise created by the technique, the accuracy of detection and the amount of overhead required. These factors were chosen as they are the most important factors during an offensive engagement like a purple teaming exercise.

4. SUMMARY

By completing this project and answering the research questions, this paper will have determined the most effective RDP honeypot detection method for use within an offensive engagement and how it can be conducted. Along with the most effective method, other methods will be outlined, including their short comings. Recommendations around how these techniques could be patched by honeypot developers will also be included.

If successfully completed, this paper will contain information that is the first of its kind. While this is a niche area of research, the popularity and continuing rise in relevance that RDP possesses grants a level of research to be completed against it. If flaws are found, this research could then be taken further to produce tools that automatically exploit the shortcomings of honeypots to quickly identify them from legitimate RDP services.

5. REFERENCES

Boddy, M., Jones, B. & Stockley, M., 2019. *RDP Exposed - The Threat That's Already at Your Door,* s.l.: Sophos.

Greenberg, A., 2019. *BlueKeep Attacks Arrive, Bearing Cryptomining Malware.* [Online]
Available at: https://www.bankinfosecurity.com/new-bluekeep-attacks-drop-cryptominer-malware-a-13341
[Accessed 10 10 2021].

Livshitz, I., 2019. *What's the Difference Between a High Interaction Honeypot and a Low Interaction Honeypot?.* [Online]
Available at: https://www.guardicore.com/blog/high-interaction-honeypot-versus-low-interaction-honeypot-comparison/
[Accessed 4 October 2021].

Nawrocki, M. et al., 2016. A Survey on Honeypot Software and Data Analysis. pp. 1-38.

Spitzner, L., 2002. *Medium-Interaction Honeypots.* [Online]
Available at:
https://www.oreilly.com/library/view/honeypots-tracking-hackers/0321108957/0321108957_ch05lev1sec3.html
[Accessed 11 October 2021].

Sullivan-Hasson, E., 2020. *Remote Desktop Software Statistics and Trends.* [Online]
Available at: https://www.trustradius.com/vendor-blog/remote-desktop-buyer-statistics-and-trends
[Accessed 30 September 2021].

Tsikerdekis, M., Zeadally, S., Schlesener, A. & Sklavos, N., 2018. Approaches for Preventing Honeypot Detection and Compromise. *Global Information Infrastructure and Networking Symposium,* pp. 1-6.

Vetterl, A. & Clayton, R., 2018. Bitter Harvest: Systematically Fingerprinting Low- and Medium-interaction Honeypots at Internet Scale.