Spellz: Attacking robots in cyberspace - Blue Team/Red Team Approach

James Gosling*, Nico Stolz, Lukas Sanz, Matthias Rottensteiner
Höhere Technische Bundes Lehr- und Versuchsanstalt Wiener Neustadt
(Federal Technical Secondary College)
Department of Computer Science
2700 Wiener Neustadt, Austria
*Corresponding author's email: 20210200@htlwrn.ac.at

***Abstract*—This paper examines security vulnerabilities in WombatOS using a red team/blue team approach. We identified key weaknesses, including flaws in password generation, an unsecured web interface, and outdated software components. Our testing revealed practical attack vectors that require minimal resources, along with corresponding defensive strategies. The results indicate that simple configuration changes can significantly enhance security while maintaining functionality.**

I. Introduction

According to the 2025 Botball game review, the infiltration of an opponent's system and the subsequent infliction of damage to the run has been deemed unlawful, representing a significant encroachment on the fundamental ethos of Botball, particularly if perpetrated by the wombat of the opposing team in a double elimination scenario. However, it should be noted that such an attack can still be executed and concealed, leaving no evidence of its occurrence. The configuration of our equipment is as follows: Two wombats have been configured with the latest version (31.0.0) of the WombatOS image (as of January 20, 2025), and several additional wombats are utilized to establish a realistic environment, akin to that observed at the Botball table. One wombat is designated as the "defender" (blue team), while the other is configured as the "attacker" (red team). The objective of the attacker is to infiltrate the defender's system by employing various methods, including WiFi and Bluetooth.

The subsequent presentation will commence with a comprehensive exposition of all attack methods. Subsequently, a detailed exposition on defensive strategies will be presented.

II. Literature Review

In this literature review, the fundamental principles of attacking systems and defending against such attacks will be examined, as these principles are also employed by the technology that has been developed. Initially, the Crowdstrike website offers comprehensive documentation on the principles of blue and red teaming [1]. However, it is imperative to ascertain the specific operating system in use to ensure a comprehensive understanding of the subject matter. However, literature on WombatOS is scarce due to its limited usage. However, given its open-source nature, conducting a code review is a feasible endeavor. The Linux distribution on which the OS is based has been identified: specifically, it is Debian 11, codenamed bullseye, which was found out using the following command:

```
lsb_release -d
```

However, it should be noted that this image is not the current stable Debian version. The rationale behind WombatOS's sustained reliance on an outdated version is likely attributable to the potential ramifications of upgrading, which could compromise the functionality of critical systems, such as the touchscreen drivers. This assertion is further substantiated by the observation that the issue was reported at a time when WombatOS was still at Debian v8.11. This observation suggests the potential for vulnerabilities in subsequent versions, which would necessitate substantial effort from the WombatOS developers to address. A number of vulnerabilities have been identified, the most significant of which is the well-known SSH regression exploit [4] due to openSSH version 8.4p1, which is utilized on the WombatOS. Of greater concern is the fact that access to the terminal is possible via the unsecured web interface, with the only requirement being that the user is within the same network as the desired wombat. This motivates the experiments contained in this paper, as attackers are most likely to use those due to their simplicity. However, it is possible to circumvent this by abusing facets of the password generator, which always generates a password of 8 characters consisting solely of small letters and digits. It is noteworthy that the final two characters of the password will invariably be zeroes, while the initial six characters are derived from the SSID. In order to circumvent the potential for exhaustive experimentation, it is imperative to establish a set of guidelines. These guidelines should encompass temporal constraints, structural complexity limitations (excluding temporal complexity), and the employed methodologies.

- Limitations in time:
  - Fortunately for the purposes of this study, but unfortunately for the competitors, the allotted time frame is six days. This is because if a bot is cracked during this period, a considerable amount of damage can be inflicted on the competitors' later runs. However, it

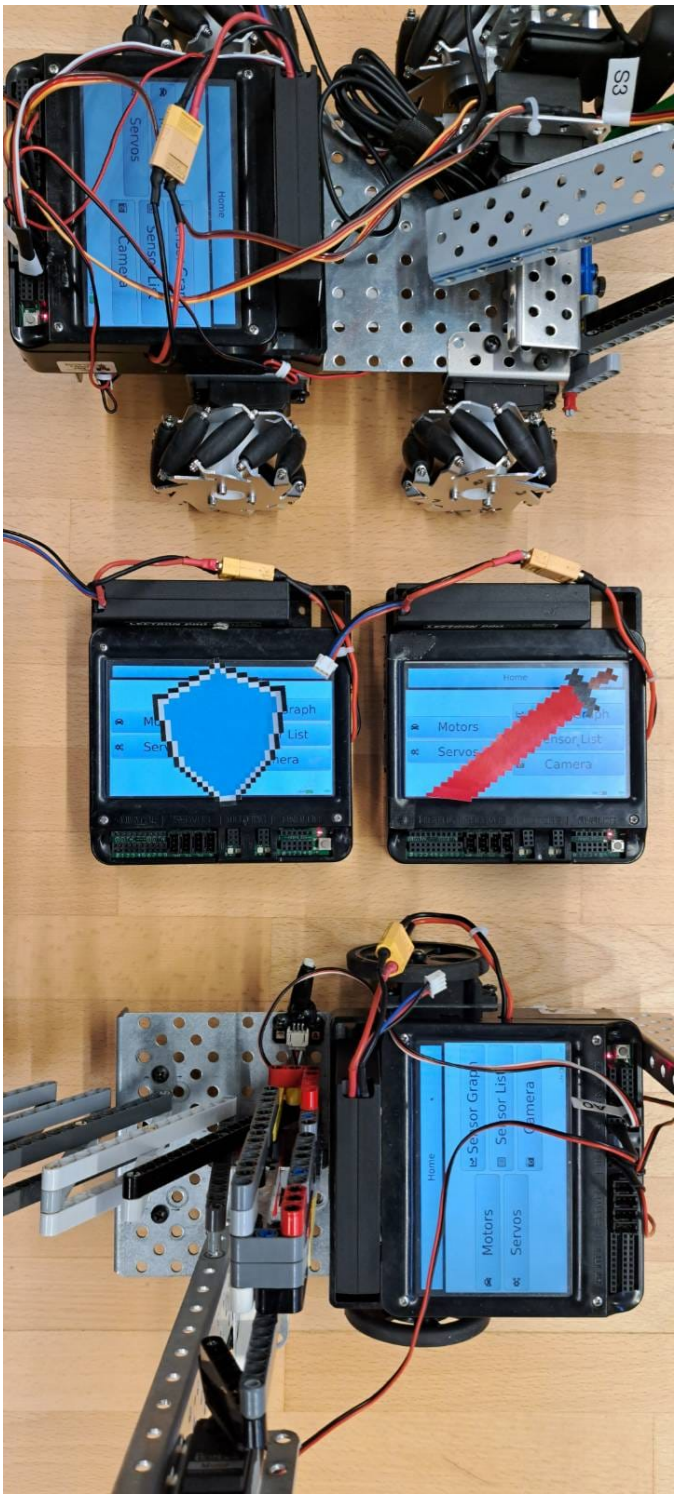

Fig. 1. The red teaming (illustrated with a red sword) and the blue teaming (illustrated with a blue shield) wombats in a realistic environment as could be witnessed at the ECER.

should be noted that the objective is to crack the bot earlier.

- Limits in complexity:
  - If an attacker can execute an undetected action, they will prioritize it over full system breaches. In practice, this implies that if an individual is a participant in the network and is aware of the IP address of the bot, they will utilize the port that has already been opened for them: the web interface. Given its sufficient capabilities to grant access to the terminal, there is no necessity to crack the SSH password.
- Methods used
  - Methods one could use include: brute-forcing, hash-tables or dictionary attacks.

III. Lab Setup

The wombats have been outfitted with the standard wombatOS image, and they are positioned on an even surface, with distances that vary slightly but remain within the confines of the gametable. From this vantage point, the red teaming wombat endeavors to infiltrate the blue teaming wombat. A number of wombats are dispersed in the vicinity to simulate a realistic environment, as would be observed at the ECER. This configuration is illustrated in figure 1.

IV. Red Team - Attack

A. Ethical/legal position

For the purposes of this paper, our red team will not be concerned with any ethical or legal issues. However, as previously mentioned in the introduction, it is both unethical and morally reprehensible to hack into another person's system. The 2025 Botball Game Review has introduced a novel regulation that explicitly prohibits such practices.

1) Botball rulesets: The 2025 Botball Game Review explicitly prohibits interference with opposing teams' systems. Thus, rendering them both unlawful and likely to result in disqualification from the tournament upon discovery. The rules state:

"Any teams found in violation of any communication hacking or tampering with another team's robots or equipment is in violation of the "Spirit of Botball" and may be disqualified from the rest of the tournament."

Beyond the parameters of competition, unauthorized system access may constitute a criminal offense under prevailing legislation. Given that ECER 2025 is to be hosted in Austria, the Austrian Criminal Code (ACC) is applicable, with two key sections being particularly relevant.

Paragraph 118a of the ACC criminalizes the act of gaining unauthorized access to a computer system by circumventing its security measures. This provision is especially pertinent

when such actions are perpetrated with the intent to procure protected personal data or to cause harm, in which case the penalty may include a maximum of two years of imprisonment. [9]

Paragraph 126 of the ACC addresses the disruption of a computer system's functionality through unauthorized data input or transmission. Depending on the severity of the offense, this may result in imprisonment for a maximum of six months or a financial penalty. [10]

It is imperative to underscore that any instance of unauthorized access can potentially result in grave legal ramifications.

2) Ethical/unethical hacking: HACKER noun 1. A person who enjoys learning the details of computer systems and how to stretch their capabilities—as opposed to most users of computers, who prefer to learn only the minimum amount necessary. 2. One who programs enthusiastically or who enjoys programming rather than just theorizing about programming. The term "hacker" is often associated with negative sentiments [11]. The term often conjures up images of individuals attired in black, confined within a small space. However, it should be noted that the majority of hackers do not engage in illegal activities. While some engage in illicit activities to benefit themselves, as discussed in the previous section, there are numerous "pentesters," otherwise referred to as "white hat hackers," who engage in ethical hacking. These individuals engage in the practice of identifying vulnerabilities in systems, often working in collaboration with system maintainers to facilitate the remediation process by the developers. The objective of this paper is consistent with the aforementioned point.

B. Aircrack and Hashcat

Although the attack was successfully automated, it is imperative to emphasize that no script will be provided due to security concerns. It should be noted that remote code execution can be achieved within minutes when executed on a typical machine. We believe that withholding the implementation is the most responsible course of action, as it avoids enabling potential bad actors. The attack, as demonstrated in the aforementioned proof of concept, is carried out in the following steps:

1) The Wombat application can be configured to enter monitor mode by utilizing the airmon-ng tool, which is an integral component of the aircrack package.
2) The airodump-ng software, which is included in the same suite, facilitates the identification of local networks. Wombat networks are systematically designated as such: SSID-wombat.
3) A less aggressive yet more difficult to detect method involves re-using airodump and specifying the BSSID and channel. In the event that a handshake occurs during the execution of the command, a .cap file containing the handshake's data will be generated [7] (see 2). (It should be noted that this method is ineffective unless no other devices are connected.) For a more assertive approach, please refer to the subsequent, more aggressive strategy. This method is referred to as "sniffing." [6]
    - Once sufficient IVs have been captured in the file, the utilization of aircrack-ng is permissible, thereby bypassing the subsequent steps. During the testing phase, this occurrence was infrequent; however, it is plausible that variations may emerge during the tournament.
4) Subsequently, the .cap file must be converted to a .22000 file using hashlib (specifically, the hcxpcapngtool command). It should be noted that the authors of this paper exclusively utilize Python or Bash, and therefore, the functionality of this conversion in other environments cannot be guaranteed.
5) The utilization of Hashcat, configured to crack a six-character password comprising solely lowercase letters and digits, is now required. The password in question must be provided with the suffix 00. The reason why the initial eight-character password always ends with the numbers 00 is not known by the authors of this paper as of the present writing.
6) It is possible for an individual to possess the password and successfully log in to the WiFi network. As a participant of the network, one can utilize a port- or ARP-scan to identify other devices.
7) Finally, the user is instructed to access the web interface using the IP address obtained in the previous step and port number 8888. Upon completion of this step, the user will be granted access to the terminal. This marks the culmination of the present paper, which addresses measures that can be implemented to defend against this practice.

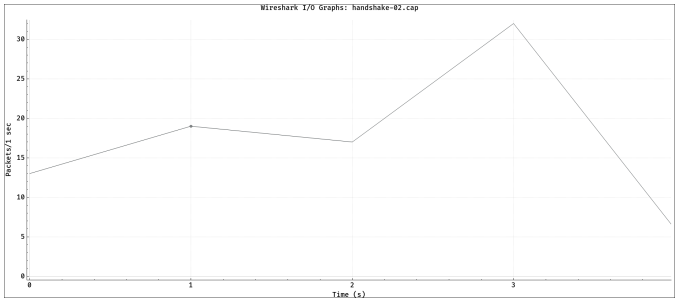

Fig. 2. Depending on what is being done to/with the Wombat, the rate of captured packets changes, thery making it easier/harder to attack (Graph generated by Wireshark)

V. Blue Team - Defend

A. Configuration

It is imperative that each team assign priority to fundamental security configurations to ensure the protection of their systems. The implementation of default passwords and credentials

must be replaced with strong, preferably randomly generated ones to mitigate the risk of unauthorized access. Furthermore, all unused potential attack vectors should be secured. For instance, if a team does not utilize the web interface, it should be disabled.

B. A simple Approach

The most straightforward method of defending against cyberattacks is to deactivate the network adapter on wombats. However, this approach is not without significant drawbacks. It restricts code uploads to wired connections via cables or external USB drives. From a gameplay perspective, this approach is disadvantageous as it hinders communication between wombats, thereby severely restricting the implementation of sophisticated strategies that depend on data sharing between bots.

Furthermore, the efficacy of this measure is questionable, as bots might still be susceptible to cyberattacks despite the implementation of this defense. The red team could potentially install malicious software on the system without the blue team's awareness by connecting an external drive. This type of attack, which involves physical intervention by the red team, will not be further explored in this work, as it is outside the scope of this paper and considered too high-risk due to the increased likelihood of detection compared to a cyberattack. This underscores the necessity for more advanced network security measures.

C. A more advanced Approach

Alternatively, since the primary attack vector relies on intercepting handshakes, it is possible to reconfigure the controllers to prevent them from being sent. This approach limits the resources available to potential attackers while maintaining uninterrupted connectivity, allowing for prompt implementation of necessary changes. However, this method could introduce compatibility issues.

Additionally, enabling MAC address filtering on the Wombat can restrict access to only authorized devices, significantly enhancing network security. This method benefits teams in two ways: first, it serves as a safeguard against hacking attempts; second, it facilitates more efficient task distribution among team members. However, a notable limitation is that team members will not have access to controllers outside their designated domain, though the effectiveness of this restriction is debatable.

Finally, it is important to acknowledge the drawbacks of this approach, including the ongoing need to enable, configure, and maintain the system by adding or removing access privileges as necessary.

D. Suggestions for developers/maintainers

Those responsible for developing the primary version of WombatOS, or those who attempt to create a fork of it, could enhance their security measures to utilize WPA3 rather than WPA/WPA2. This would significantly enhance the system's security. This would result in a substantial enhancement of the system's security. Furthermore, the generation of passphrases with eight characters, as opposed to the prevailing practice of appending two zeroes to a six-character passphrase, could be implemented. This modification would lead to an exponential increase in the time required to crack the passphrase, as illustrated in Table I. Ideally, the number of characters would be increased; however, this would compromise the user interface. Alternatively, the microcontroller could be configured to connect to a different Wi-Fi network automatically, though this would require prior configuration. Additionally, a menu of network choices, similar to those found on mobile devices or personal computers, could be provided, thereby eliminating the need for manual configuration.

with the use of a single computer source: [5]

Type of characters	Allowable number of characters to create a password	3 character password	6 character password	8 character password	12 character password
		Time necessary to decrypt a password	Time necessary to decrypt a password	Time necessary to decrypt a password	Time necessary to decrypt a password
Lower case letters only – the Latin alphabet	26	0,02 s	5 min	58 h	3000 years
Lower case letters only – the Latin alphabet and digits	36	0,04 s	36 min	32 days	150000 years
Upper and lower case letters – the Latin alphabet and digits	62	0,2 s	15 h	7 years	100 million years
Upper and lower case letters – the Latin alphabet, digits, special characters	94	1 s	8 days	193 years	Longer than the Earth exists

TABLE I
EFFICIENCY OF CRACKING PASSWORD WITH VARIOUS LENGTHS AND CHARACTER SETS

VI. Red Team - Strategy No. 2

This approach is more aggressive and thus more readily detectable; however, it proves to be significantly more effective, as it does not depend on the actions of other teams. Undertake the same procedure as delineated in the initial strategy, with the exception of step 3. Although numerous continuation methods exist, we will delineate only one due to the comparative ease of detection among many; the variant presented here is the most expeditious. This method is termed ARP-Replay, and the requisite tools are included within aircrack. The specific tool required is aireplay. For security considerations, scripts or code snippets will not be provided.

VII. Blue Team - Strategy No. 2

The recommendations presented herein largely coincide with those provided in the prior segment pertaining to the blue team. Nevertheless, maintaining a connection remains unfeasible for users, irrespective of the handshake [7] configurations.

VIII. Additional Notes

1) It is imperative to acknowledge that the attack methodologies employed by the red team do not result in any impairment to the wombat system or a permanent alteration to its configuration.

2) It is imperative to acknowledge that the objective of this work is not to encourage participants of the Botball tournament or any other individuals to engage in hacking activities. Furthermore, it is not intended to censure developers for allowing these vulnerabilities to exist. Rather, it is intended to serve as a catalyst for enhancement, offering insights that facilitate the identification of areas requiring refinement by the WombatOS developers and any individuals contemplating the creation of a forked version of the system. The primary objective of this work is to provide insights into defending against cyberattacks within the context of the Botball tournament, provided that the exploits remain unresolved.

IX. Conclusion

This study has explored the security vulnerabilities of WombatOS within the Botball competition environment, highlighting potential attack vectors and effective defense strategies. Through a structured red team/blue team approach, we identified key weaknesses, including outdated software, weak password generation mechanisms, and insecure web interface access. Our findings suggest that competitors utilizing WombatOS must adopt proactive cybersecurity measures, such as disabling unnecessary network interfaces, strengthening authentication methods, and monitoring network activity for anomalies.

While ethical considerations and competition regulations prohibit real-world exploitation of these vulnerabilities, understanding these risks is crucial for fostering a more secure and fair competitive environment. By implementing the defensive strategies outlined in this paper, teams can mitigate the risk of cyber threats while maintaining seamless communication and functionality. Future research should focus on further hardening WombatOS against emerging threats and exploring more sophisticated countermeasures.

Acknowledgements

The authors would like to express their profound gratitude to Michael Stifter, Jakob Eichberger, and the members of robo4you for their invaluable assistance in preparing this paper. They would also like to acknowledge Clemens Koza and Tim Corbly for their support and prompt responses online, as well as for their instrumental role in making the ECER a reality.

References

[1] Explaining the red-vs-blue-team approach
https://www.crowdstrike.com/en-us/cybersecurity-101/advisory-services/red-team-vs-blue-team/
(last accessed on January 29, 2025)

[2] Potential Security Vulnerabilities in Raspberry Pi Devices with Mitigation Strategies
https://digitalcommons.odu.edu/cgi/viewcontent.cgi?article=1077&context=covacci-undergraduateresearch
(last accessed on January 29, 2025)

[3] Get the Wombat to work on a fresh OS without breaking things
https://github.com/kipr/KIPR-Development-Toolkit/issues/5
(last accessed on January 31, 2025)

[4] regreSSHion: Remote Unauthenticated Code Execution Vulnerability in OpenSSH server
https://blog.qualys.com/vulnerabilities-threat-research/2024/07/01/regresshion-remote-unauthenticated-code-execution-vulnerability-in-openssh-server
(last accessed on January 31, 2025)

[5] KIPR Wifi Configurator
https://github.com/kipr/KIPR-Update/blob/master/files/wifi_configurator.py#L81
(last accessed on February 04, 2025)

[6] newbie_guide [Aircrack-ng]
https://www.aircrack-ng.org/doku.php?id=newbie_guide
(last accessed on February 14, 2025)

[7] What happens in a tls handshake
https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/
(last accessed on February 14, 2025)

[8] https://www.cyberly.org/en/how-do-you-prevent-attacks-using-aircrack-ng-on-your-network/index.html https://www.cyberly.org/en/how-do-you-prevent-attacks-using-aircrack-ng-on-your-network/index.html
(last accessed on February 17, 2025)

[9] § 118a StGB https://www.jusline.at/gesetz/stgb/paragraf/118a
(last accessed on February 25, 2025)

[10] § 126b StGB
https://www.jusline.at/gesetz/stgb/paragraf/126b
(last accessed on February 25, 2025)

[11] Ethical hacking
https://ieeexplore.ieee.org/abstract/document/5386933
(last accessed on February 28, 2025)