SWARMFLAWFINDER: Discovering and Exploiting Logic Flaws of Swarm Algorithms

Chijung Jung* , Ali Ahad* , Yuseok Jeon† , and Yonghwi Kwon*
.**Department of Computer Science, University of Virginia*, Charlottesville, VA, USA
†*Department of Computer Science and Engineering, UNIST,* Ulsan, South Korea
Benjamin R. Bement

1. Summary of the paper's findings

1.1. Introduction and context of the study

Swarm robotics is the utilization of specialized algorithms to control a group of machines without a set centralized control. Rather the collective behavior of each individual robot is based on each robot's interactions with each other as well as the surrounding environment. As advancements continue to be made in this field and utilization increases, the importance of having effective testing methods increases. This presented work aims to forward advancements in swarm algorithm testing through the proposal of a new testing metric and new automated testing system.

Swarm robotics aim to revolutionize the robotics industry. With implications in almost every major field including military warfare and commercial business, it becomes important to recognize the need for security advancements within this technology. One area is robotic software fuzz testing responsible for identifying logic flaws. Traditional software testing systems rely on coverage-guided fuzz testing which is ineffective when applied to the field of robotics. Traditional methods rely on predicate conditions (expressions that result in true or false) to determine a test case's effectiveness which is not ideal for robotic systems due to their simple control flow. To account for this, the proposal of the degree of causal contribution (DCC) is made. Utilizing DCC and greybox fuzz testing techniques, SWARMFLAWFINDER was created to address the unique challenges of testing drone swarm algorithms.

1.2. Research questions

This paper aims to propose and test a new systematic approach to find logic flaws in drone swarm. Through the utilization of SWARMFLAWFINDER and the DCC testing metric, various tests are conducted on four swarm algorithms to identify logic flaws as well as test to determine the effectiveness of DCC versus traditional coverage-based methods.

1.3. Main papers contributions

Contributions provided by this work entail the development of improved swarm robotics testing methods aimed at identifying swarm algorithm logic flaws that would otherwise remain hidden. To ensure the ability of SWARMFLAWFINDER to determine a test case's effectiveness, the degree of causal contribution (DCC) is proposed to measure the effects of attack drones on a swarm mission. Utilization of DCC contributes to remedying swarm robotics testing challenges involving the ineffectiveness of traditional coverage-based metrics due to the ability to quickly cover the entirety of code and logic branches without covering all possible behaviors.

With these challenges identified, SWARMFLAW FINDER is presented as a greybox fuzz testing system aimed to expand upon current swarm testing methodologies. Through thorough testing, 42 previously unknown flaws were identified amongst the four test algorithms. This identification eventually led to the remediation of 34 out of the 42 logic flaws. With this system developed and tested, similar results are achievable amongst many swarm algorithms that have been developed and have yet to be developed and in turn provide increased protection against attacks aimed at inducing swarm mission failures.

1.4. Threat model

The proposed threat model provides clarification on the capabilities of the attacker and attack drones tasked with preventing mission success. During each test the adversary is aware of the target swarm's mission and is capable of launching external drone attacks to prevent the mission. Coinciding with SWARMFLAWFINDER's implementation as a greybox fuzz testing system, the adversary is partially aware of the internal code structure of each victim drone but is incapable of compromising the hardware or software of each. In addition to this, the adversary is not capable of executing attacks that sacrifice adversary drones through physical contact. Attacks must rather influence a victim drone's movements to induce a crash among victim drones or separation of a drone from the swarm.

Victim drones follow controlled guidelines as well to simulate real-life drone mission execution. In order to complete a mission successfully, all victim drones must complete the required objectives in no later than twice the normal mission completion time of 400 ticks with zero collisions. Mission execution must be facilitated through autonomous swarm algorithms and the swarm must continuously make progress to mission completion.

1.5. Summary of the methodology

To begin the discussion of the methodology used to answer the presented research questions, we must first discuss the

degree of the causal contribution (DCC). Proposed to provide an effective alternative to coverage-based metrics, DCC works by quantifying each victim drone's reaction to an attack drone or obstacle. This is done through the idea of counterfactual causality which states that a behavior B is causally dependent on an event A and if A did not exist then B would not exist. Computation is conducted through multiple perturbations where objects or drones are removed from a test case resulting in different flight direction vectors. These differences of these vectors from each perturbation are aggregated to calculate a drone's respective DCC value.

Utilizing this DCC value, testing is conducted through a four-step process: the creation of an initial test case, test case evaluation and DCC computation, test mutation, and repeated test execution and mutation until test timeout. The first step in the process involves the creation of an initial test case consisting of two major data components, the drone swarm's mission and the position and attack strategy of the attacking drone. Given the drone swarm's mission to be tested, each attacking drone's position and strategy are given as a set of tuples with position given a 3-dimensional point in space and strategy given as a selection of one of the four major attack strategies. The initial attack drone's position is constricted to all points outside of a predefined distance from the initial starting points of the target drones to prevent immediate collisions.

With the initial test case defined, execution and evaluation of the test case is conducted. After each test, the test case is evaluated to determine if a new behavior from the victim swarm was induced. This is determined through DCC evaluation and comparison to all previous DCC values recorded. Based on this test case evaluation, the level of mutation to this test case is determined. If DCC evaluation determines that the victim swarm exhibited a new behavior, the test case is mutated only slightly to conduct similar test runs. If DCC evaluation determines that a test case's execution resulted in a behavior similar to a previous test, a larger mutation is conducted in an attempt to cause a new behavior to occur in the victim swarm. This four-step process is repeated until a certain timeout value is reached. The result of this process is the compiled test cases responsible for causing a mission failure.

1.6. Summary of the experiments and results

During the execution of this research, four algorithms were tested due to their complexity and compatibility with the proposed testing software. The four selected algorithms were Adaptive Swarm, Socratic Swarm, Sciardo, and Pietro's algorithm. To provide a summary of experiments and main results achieved, each algorithm must be analyzed along with its testing results. Beginning with Adaptive Swarm, it is a Python algorithm created with the purpose of moving a swarm of drones from one point to another while avoiding any presented obstacles. Mission selection for this algorithm was chosen as multi-agent navigation to best align with its purpose. Testing was conducted for 24 hours using 4 drones within the swarm and resulted in 1,554 mission failing executions and the discovery of 10 logic flaws within the algorithm. Analysis of the results identifies 4 failing scenarios for this algorithm which were crashes between victim drones, crashes into external objects, suspended progress, and slow progress. With these test cases identified, fixes were implemented for each respective flaw with one example being ***C1-5*** which identified a flaw within the algorithm's pose management. This logic error would occur when computing the position of all drones and would often neglect drones causing them to fall behind and eventually lead to swarm collapse. With these testing results provided to Adaptive Swarm's developers, important improvements can be made to help better prevent exploits against this algorithm.

The next algorithm tested was Socratic Swarm which is a C# algorithm aimed at conducting coordinated search missions through active interactions within the swarm. Testing was conducted for 24 hours utilizing using a coordinated search mission with 8 drones in the swarm. Following testing, the results identified 755 failing test cases with the discovery of 4 logic flaws. Analyzing these failing test cases, they can be categorized into 3 categories with crashes between victim drones, suspended progress, and slow progress. This experiment was conducted to test a different swarm mission set and is important for the development of Socratic Swarm. Summarizing one logic flaw through the discussion of ***C2-2***, this flaw entailed faulty logic within the algorithm's functionality for assigning tasks for the swarm where crashed drones would still be included in this assignment and would receive tasks that would in turn not be completed.

The Sciardo algorithm was also tested which was created to run multiple swarms over a wide area to search for specific targets. Testing was conducted for 24 hours utilizing using a distributed search mission with 10 drones in the swarm. Following testing, the results identified 287 failing test cases with the discovery of 4 logic flaws. These failing test cases can be placed into 2 categories with crashes into external objects and slow progress. This experiment was conducted to test the functionality of drone swarms to dynamically split into smaller groups and conduct multiple searches while still maintaining a central connection. One logic flaw to highlight is ***C3-1*** which entailed faulty logic within the obstacle avoidance functionality where if a drone selects a new path to avoid an obstacle and another obstacle is in that new path, the drone will crash without avoiding the second obstacle.

Lastly, Pietro's algorithm was tested which was created to achieve cooperative rescue missions. Testing was conducted utilizing a search and rescue mission for 24 hours using 15 drones within the swarm. Results from this test included 2,088 failing test cases and identified 5 logic flaws within the algorithm. Failing scenarios included crashes between victim drones, crashes into external objects, and slow progress. This

test was conducted to evaluate another executable algorithm with a different purpose in swarm robotics. Through these tests, logic fixes can be made to eventually create an improved algorithm ready for testing in the real world. Highlighting one logic flaw found, ***C4-2*** identified a flaw within obstacle avoidance where a drone would avoid an obstacle but would not consider if there was an obstacle in this alternate path.

2. Main takeaways and limitations of the work

2.1. Major takeaways

One major takeaway from this work is the provided evidence of the effectiveness of the DCC system versus traditional coverage-based metrics in providing a more effective spatial distribution of test cases. Traditional coverage-based methods only factor the mission success when determining to perturb a test case to achieve a different result leading to an inefficient selection of test cases. Evidence of this can be seen through various tests conducted in this work, the main one being an evaluation of the spatial distribution of test cases produced by both methods. This test was conducted by running both the traditional coverage approach and DCC approaches through SWARMFLAWFINDER for 24 hours to measure the variety of test cases produced by both. After inspection, it was shown that DCC based methods resulted in the discovery of 25.75% more failures than random testing, showing its effectiveness as a testing metric.

Additional takeaways are the improvements made to the tested algorithms as well as the implications of having an effective testing method on future developments in drone swarm technology. Via SWARMFLAWFINDER, 42 distinct mission failures were identified and credited to 15 unique errors within the four algorithms. Among these issues were distinct logic flaws that went undetected and would've had major implications if they were to remain undetected during real world implementation. One example is ***C1-1*** *Missing Collision Detection* which entailed a logic flaw within the Adaptive Swarm algorithm where the leader drone had no logic for avoiding other drones in the swarm. With this ability to identify swarm algorithm logic flaws effectively, the importance of this development in this advancing field cannot be understated. As stated in the presented work, swarm robotics have been developed to conduct various tasks in numerous areas such as logistic and military operations but have received minimal attention in the area of defensive capabilities. With this development, further steps can now be taken to advance security and provide aid to multiple different areas.

2.2. Major limitations

Through inspection of the presented work, an initial limitation can be found in that it does not test for all possible attack methods. While this is acknowledged and addressed through the consideration that attack strategies can be easily added to SWARMFLAWFINDER, this is an important limitation to present as it opens the door for further research and improvements regarding this testing system. One key area of drone swarm attacks that is limited by the selection in this work are drone attacks through collision. Addressed in this work's threat model, collision attacks were not considered due to economic considerations and subtleness. These are two understandable considerations but as advancements are made in drone technology, these considerations become less valid as drone technology becomes move readily available. With this technology readily available, the cost of mission success versus mission failure begins to outweigh the costs of each required drone making them expendable to a certain degree. For this reason, drone collision attacks should be considered as a viable attack strategy and therefore should be tested in order to prevent these types of attacks.

An additional limitation to this work is the security consideration tied to this field of research and the availability of this testing system. The first security consideration that must be discussed is the disclosure of active logic flaws present in each algorithm, some of which were not confirmed fixed by the release of this work (8 out of 42). While discussion of these logic flaws does provide concrete evidence of the effectiveness of SWARMFLAWFINDER, it must be discussed that specific exploit identifications and discussions are not necessary to prove the effectiveness of this testing system and could possibly have been limited to a small number of case studies. By examining Table 3 within the presented work, an attacker is able to identify the outcome of failing test cases and could possibly tailor future attacks on that algorithm to invoke that behavior. An additional consideration in security that must be discussed is the availability of SWARMFLAWFINDER to the general public. The reason this must be discussed is the nature of this testing system in taking in a desired drone swarm algorithm and outputting potential mission failing test cases. With security in mind, one can see how potential attackers could utilize the readily available software to identify potential flaws in algorithms that are available to the public and have not received proper exploit testing. This raises the question of allowing this testing system to be available on GitHub for possible free download. Further discussions should be had to limit distribution to only those who are drone swarm algorithm developers looking to test their own systems.

3. Fundamental previous work

3.1. Threats to the Swarm: Security Considerations for Swarm Robotics

F. Higgins, A. Tomlinson, and K. M. Martin, *International Journal on Advances in Security*, 2009.

3.2. Research problem

This paper aims to explore security concepts regarding swarm robotics due to the trend of overlooking security issues of emerging technology until later stages in development. With swarm robotics being an emerging technology, discussions are made to identify areas where existing security concepts and technologies are either not translatable or relevant to swarm robotics. This is done through the scope definition of swarm robotic networks to distinguish them from related technologies such as multi-robot systems, mobile sensor networks, MANETs, and multi agent systems [1]. Comparisons made identify clear distinguishing factors that prevent current security techniques from being translated from related technologies to swarm robotics and present a need from new techniques to be developed.

With the need for new security developments identified, this paper also provides areas that could soon make use of this technology and how unmitigated security issues could have potential effects. Discussions in this area include military applications, monitoring, disaster relief, healthcare, and commercial applications. Lastly, conclusions are drawn that highlight the importance of addressing these risks now.

3.3. Main contributions of the work

Contributions of this paper highlight the potential security threats and provide clear distinguishing factors that separate swarm robotics from similar technologies. Comparisons of explicit characteristics (autonomy, large quantity, simple) and implicit characteristics (self-organizing, mobile, co-operate to accomplish tasks) provide distinct areas where swarm robotics are unique in nature and therefore are not covered by security discussions of similar technologies [1]. By identifying these distinctions, this paper then effectively provides possible attacks in each potential service area that have not been mitigated and provides security challenges that arise in developing proper defenses to mitigate these attacks. Through this discussion, this paper presents an overview of challenges that open the door for future related works in the area of drone swarm security.

3.4. Correlation with the presented work

Initial discussions of drone swarm security presented in this paper provide the inspiration for the development of the testing methodology used in SWARMFLAWFINDER as well as the development of a new fuzz testing system metric due to the need of new techniques to cater to the unique nature of swarm robotics. Discussed in the presented work is the inspiration for SWARMFLAWFINDER's threat model through this paper's proposal of a potential security threat to swarm robotics through the use of foreign drones which could influence a swarm's behavior. Presented as a security challenge in swarm robotics, the discussion of the intrusion of rogue robots into a swarm in this paper provides inspiration to develop and effective testing system that identifies logic flaws in an attempt to mitigate this security issue. With this development, SWARMFLAWFINDER identifies concrete logic flaws building upon the conceptual malicious swarm models presented in this paper. With the ability to now identify these concrete logic flaws, SWARMFLAWFINDER expands upon and answers this paper's call to now address these security issues and seek solutions.

An additional inspiration can be found in this paper's discussion of the unique nature of swarm robotic technology. Discussed in the presented work is the ineffectiveness of traditional coverage-based metrics which coincides with the discussion of traditional software techniques being not translatable to swarm robotics. One key inspiration can be found in this paper's identification of swarm robotics as unique in its utilization of autonomous, simple, and self-organizing behavior [1]. When further diving into these concepts, the utilization of a metric such as the degree of causal contribution can be seen as an effective choice to leverage the autonomous nature of each drone when selecting test cases that effect the entirety of the drone swarm.

2.5. On the effects of minimally invasive collision avoidance on an emergent behavior

C. Taylor, A. Siebold, and C. Nowzari, in *International Conference on Swarm Intelligence.* Springer, 2020.

2.6. Research problem

This paper aims to explore and identify issues involving the use of two specific simplifying assumptions when testing prototype swarm algorithms using simulations. In many cases using physical drone hardware to test newly developed swarm algorithms is difficult and not cost-effective. Due to this, researchers often rely on software to model the use of swarm algorithms in a digital space. With this comes the need for simplifying assumptions about the testing environment as not all factors in the real world can be accounted for in the digital space. This paper highlights two often used simplifying assumptions which are often overlooked and can cause issues when beginning testing outside of the digital world. These assumptions are that modeled drones occupy no space in the digital world and the existence of collision avoidance algorithms that can be easily added to existing swarm algorithms [2].

2.7. Main contributions of the work

Contributions of this paper begin with the identification of previous works that overlook the physical space that drones occupy when modeling swarm algorithms. More specifically, this paper identifies one work which defines collisions as two drones being in the exact same position and three works that assume drones can pass right through each other. By identifying these studies, this paper raises important considerations for future swarm algorithm testing.

2.8. Correlation with the presented work

Both the discussions regarding the assumptions of physical space as well the assumption of adding collision avoidance algorithms provide inspirations shown in the presented

work's testing system and algorithm selection. With regard to physical space, restrictions are implemented in SWARMFLAWFINDER to prevent an attack drone's initial pose from being within a certain distance from any victim drone's initial spawn to prevent a crash before the victim drone is able to react. This is a direct outcome of considering the physical space occupied by a drone at any given moment when developing the constraints of the modeling software. More importantly, this consideration is necessary to restrict SWARMFLAWFINDER's threat model to the proposed constraint of no attacks through physical contact.

With regard to collision avoidance algorithms, this work provides clear evidence of the importance of developing swarm algorithms with collision avoidance in mind and in turn provides direct inspiration for the algorithm selection made in the presented work. This can be seen in the presented work's background discussion which identifies that the 4 selected algorithms are designed with collision avoidance in mind.

4. Proposed defenses

4.1. Defense 1 description

As discussed in the presented work, the use of collision avoidance algorithms are the primary defense utilized in drone swarm security to prevent attack drones from causing undesired behavior. One proposed defense that could provide additional capabilities is the use of escort drones. More specifically, the use of additional drones outside of the mission requirements that work to shield the victim swarm from oncoming attacks. Traditional algorithms discussed in the work aim to only avert pursuing drones but have no capabilities to prevent continued attack attempts from these drones. Through the use of escorts, the victim swarm is capable of counteracting attack attempts through the blocking and shielding of attacks.

Capabilities of these escort drones would include the ability to actively detect and track incoming attacks. With these threats detected, the escort drone works by flying directly into the incoming threat's path as to block it from reaching a set distance from the victim swarm. Through inspection of the presented work's threat model, it can be seen that attack drones do not utilize collisions as a form of attack. With this in mind it becomes apparent that this reluctance to collision can be used as a form of defense for the victim swarm against attack drones. By utilizing the "pushing back" strategy used by attack drones, an escort drone could potentially minimize the effects of an attacking drone and greatly increase the chances for mission success.

4.2. Advantages of the defense if implemented.

Advantages of this proposed defense include increased defensive capabilities and easy deployability and integration into the victim swarm. While discussions can be made about the potential cost effectiveness of utilizing additional drones that do not serve mission critical roles, in many cases mission success versus failure greatly outweighs the additional cost of these extra drones. With these increased defensive capabilities, the chances of mission success are greatly improved and therefore can be considered cost effective. With regard to deployability and integration, escort drones could be easily developed through the reverse engineering of already existing collision avoidance algorithms and the use available attack drone algorithms. With the integration of these modified algorithms into a victim drone's software, an effective escort drone can be developed that can travel with a victim swarm and provide defensive maneuvers when an attack drone is detected.

4.3. Defense 2 description

An additional defense in the area of swarm robotics that could be deployed is the capability of swarms to allow easy integration of additional drones in the event of a successful attack or complication with an obstacle. By providing a secure method for additional drones to join the swarm network, various missions could be salvageable when normally deemed as unsuccessful. This is most applicable when multiple swarm missions are being carried out simultaneously, allowing for different swarms to aid each other with additional drones. In the event that a mission critical drone were to be destroyed, a neighboring swarm could donate a drone which would then be integrated into the new swarm and receive it's updated mission.

With this proposal comes the need of a secure method to properly authenticate various drones when conducting integration. To provide this authentication, the integration of a blockchain based security system to manage the identity of each drone could be utilized. This system would leverage existing blockchain mechanics utilized in cryptocurrency such as traceable blocks linked by hash values. This existing security functionality could provide a tamper-proof identity management system allowing for secure authentication into the swarm.

4.4. Advantages of the defense if implemented.

The advantages of this proposed defense include improved disaster recovery and possible security advancements in drone authentication. Similar to the previously proposed defense, arguments could be made that this proposed defense also utilizes additional drones which come with increased costs if there are no neighboring swarms to rely on. With that being said, the cost effectiveness is again justifiable in that it greatly improves the chances of mission success without major costs. In the scenario of a military mission where lives are often at stake, the ability to provide reinforcements if a drone swarm comes under attack can be the difference between life and death. With regard to the blockchain authentication method, adaption of an existing and effective security technique could prove to be effective in remedying

security issues even outside its intended use. Issues of drone authentication have been raised in multiple works and this implementation could prove to be effective and reduce the time needed to develop a new authentication system.

5. Proposed follow-up research idea

5.1. Follow-up research idea description

Deriving inspiration from the attacks tested in SWARMFLAWFINDER, this presented research idea discusses the development of an attack method that focuses on the elimination of a specific drone of the attacker's choice in the swarm. Inspired by the use of leader drones in the presented work which are in charge of directing the swarm, this attack aims to disrupt the swarm using a more effective method of targeting this leader drone if desired. Additional use cases include targeting drones responsible for carrying specific cargo. Key development points of this attack method include the ability of an attacking drone to analyze a victim swarm to identify its intended target. Once this is accomplished it must be able to conduct an attack on that specific drone. Utilization of previously discussed attack methods and algorithms will be leveraged in the development of this attack method. More specifically, the utilization of attack methods such as pushing back, chasing, dividing, and herding which are discussed in the presented work would be leveraged once proper identification of the target drone is carried out.

Testing would be conducted utilizing a modified version of SWARMFLAWFINDER aimed at testing the effectiveness of a drone attack algorithm. Modifications of this testing methodology would include the ability to easily insert various drone attack algorithms for testing. With this completed, testing would be conducted to determine the effectiveness of the targeting attack algorithm at causing mission failures versus previously developed attack strategies. The resulting mission failures produced by SWARMFLAWFINDER could be analyzed to determine if the proposed attack idea proves to be more effective than traditional attack methods.

5.2. Research questions

As advancements are made in the field of drone swarm technology, the need for defensive as well as offensive capabilities for drone swarms becomes a forefront concern in the world of military technology. The ability to effectively disable an enemy's attacking drone swarm soon becomes a matter of life and death for soldiers and civilians in harm's way. In order to counteract this, this research idea proposes a new form of attack aimed at targeting specific drones in a swarm to disrupt the overall behavior and render it ineffective. Through the use of SWARMFLAWFINDER's ability to conduct attack scenarios, this targeting algorithm is tested against previously developed attack methods to determine its overall effectiveness as a new form of attack.

REFERENCES

[1] F. Higgins, A. Tomlinson, and K. M. Martin, "Threats to the swarm: Security considerations for swarm robotics," *International Journal on Advances in Security*, 2009.

[2] C. Taylor, A. Siebold, and C. Nowzari, "On the effects of minimally invasive collision avoidance on an emergent behavior," in *International Conference on Swarm Intelligence.* Springer, 2020.