Analysis of A Malicious Deutsch-Jozsa Circuit

Jaden Hawley
Electrical Engineering and Computer Science
University of Missouri - Columbia
Columbia, MO, USA
jrh22r@umsystem.edu

Chi-Ren Shyu
Electrical Engineering and Computer Science
Institute for Data Science and Informatics
University of Missouri - Columbia
Columbia, MO, USA
shyuc@missouri.edu

2024 IEEE International Conference on Quantum Computing and Engineering (QCE) | 979-8-3315-4137-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/QCE60285.2024.10425

***Abstract*—Active research into multi-tenancy means multi-tenant quantum computers could be soon. However, the growth of quantum computing introduces new security risks. One such risk is crosstalk error in multi-tenant superconducting quantum computers, which can be used to inject faults into other users' circuits. Concurrent works in the field have designed malware that exploits this vulnerability. This work demonstrates a malicious Deutsch-Jozsa malware that resists mitigation. This is created by designing a Deutsch-Jozsa circuit that contains a malicious payload. To analyze this malware, we assess both covertness and mitigation resilience. For covertness, we evaluated the impact on the integrity of the Deutsch-Jozsa circuit it hides in. Our findings show that the malware closely mimics the performance of an innocent Deutsch-Jozsa circuit at all optimization levels, which shows success in covertness. To assess mitigation resilience, we use Qiskit's circuit optimization. We found that our malware remains mostly unmitigated by optimization and almost doubles the victim error rate at the highest level of optimization offered by IBM. This shows that hiding a well-designed payload within a wrapping circuit does not reduce the malware's effectiveness, and it even performs better than the non-deceptive variant.**

Keywords—Quantum Computer, Security, Cybersecurity

I. Introduction

Multi-tenant quantum computing is nearing reality [1][2], but this development brings new concerns. Crosstalk errors could be exploited to sabotage other user's circuits in a multi-tenant environment [3]. Crosstalk occurs on connected qubits when a gate influences adjacent qubits the gate does not act on [4]. If the affected qubit belongs to a circuit from a different user, this interference compromises the integrity of the other user's circuit [3]. Attackers may exploit this by creating malicious circuits that generate significant crosstalk known as quantum malware [5][6]. Modern quantum compilers have a circuit optimizer, which can coincidentally mitigate potential quantum malware [5][7]. We introduce and analyze deceptive malware that hides as a Deutsch-Jozsa (DJ) circuit to avoid mitigation.

II. Related Work

The malware presented in this paper is inspired by the malware presented in the works of Deshpande et al. [5][6], which is designed to attack superconducting quantum computers using crosstalk errors. Mitigations for this malware include compiler optimization [5], antivirus [6], and circuit separation [8], which were all used to mitigate a plain attack. A similar vulnerability exists on ion-trap quantum computers [9]. Malware that can exploit this has been showcased [9] and mitigations have been developed [9][10].

This research is supported by the National Science Foundation DEG-1946619.

III. Methods

We use real quantum hardware: IBM Osaka (127 qubits). The malware and victim are placed adjacent to each other and run at the same time. This captures a scenario that could happen if today's quantum computers were multi-tenant. This is done through the usage of publicly available hardware and running our victim and malware in parallel. We use three 10-qubit clusters for our tests to capture the behavior at multiple locations. This size was chosen to reduce the size and depth of the tested circuits. This reduces the effects of other noise and decoherence on the results. The cluster is partitioned as follows. Cluster 1 uses qubits 0, 1, 2, 3, 4 for the victim and qubits 5, 6, 7, 8, 9 for malware. Cluster 2 uses qubits 122, 123, 124, 125, 126 for the victim and qubits 117, 118, 119, 120, 121 for malware. Cluster 3 uses qubits 62, 63, 64, 65, 66 for the victim and qubits 45, 46, 47, 48, 54 for malware. When testing malware or victims alone, only victim qubits are used. We chose these clusters to capture the behavior of different sections and layouts on the quantum computer to ensure robust results.

A. The Victim Circuit

The victim circuit used for our benchmarking is the Deutsch-Jozsa (DJ) circuit. This was chosen because it is deterministic, therefore, error could be measured precisely. The effectiveness of the malware is measured by comparing how much it affects the error rate of the victim. The error rate is the total incorrect outputs divided by total shots.

B. The Malware Circuit

The malicious DJ Circuit we design and explore is made of two components: the payload and the wrapper.

1) The Payload: The payload is a circuit that causes crosstalk. It is designed to be placed in another circuit. The payload can also be run alone as simple plain malware. The DJ payload shown in Fig. 1 is the standalone payload that we will use to test plain malware in our experiments.

2) The Wrapper: The deceptive malware is developed by placing the payload inside a wrapper circuit, so it runs while the wrapper is running. We use the DJ circuit as our wrapper. The DJ circuit can have unused qubits, which are perfect for our payload placement. Fig. 2 shows the complete malicious DJ circuit. The wrapper circuit should still function similarly to the original innocent circuit. Retaining its functionality can be used to trick unsuspecting victims into running the malware as a

979-8-3315-4137-8/24/$31.00 ©2024 IEEE
DOI 10.1109/QCE60285.2024.10425

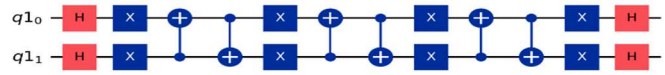

Fig. 1. A 2-qubit payload that can be inserted into the DJ wrapper.

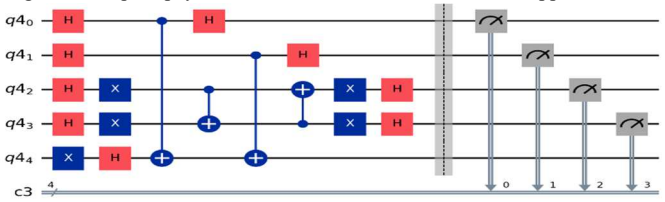

Fig. 2. Malicious DJ circuit with a small example payload on qubits 2 & 3.

library. The functionality was assessed by running the malware with no victim and comparing the accuracy of its wrapping circuit to the accuracy of the same wrapper without a payload.

C. Optimization

Qiskit provides four levels of optimization when compiling, ranging from 0 (no optimization) to 3 (heaviest optimization). To ensure the circuits are still adjacent, the circuits will be forced onto specific qubits.

IV. Results and Disscusion

A. Attack Performance

Table I showcases the effects of the malware on the victim circuit. The percentages represent the average error of the victim across all three clusters. The table presents results for each level of optimization tested. The similar behavior of DJ malware and plain malware at low optimization levels demonstrates that DJ malware is as much of a threat as a plain fault injection. However, the deceptive malware remains effective at higher optimizations as well.

B. Wrapper Functionality

Another key aspect of this malware is that it does not impact its own integrity too severely. Table II shows the error rate of the DJ malware's wrapper circuit with and without the payload. The circuit remains comparably functional at all optimizations.

TABLE I. Malware vs Optimization

Test	Victim Error Rate			
	Opt. 0	***Opt. 1***	***Opt. 2***	***Op. 3***
Victim Alone	37.7%	20.8 %	20.8%	23.1%
Plain Malware	49.7%	31.4%	19.7%	23.7%
DJ Malware	50.9%	35.2%	38.7%	45.3%

TABLE II. Deutsch-Jozsa Malware

Test	Wrapper Error Rate			
	Opt. 0	***Opt. 1***	***Opt. 2***	***Opt. 3***
No Payload	37.7%	20.8%	20.8%	23.1%
With Payload	35.4%	22.9%	24.5%	22.7%

V. Conclusion

We showcase quantum malware that hides in an innocent circuit. At higher optimizations, the effects of the DJ malware persisted and showed resilience to optimization as a mitigation method. The DJ malware remained functional with the payload. This demonstrates that the DJ malware remains hidden by not significantly affecting the size or integrity of its host circuit. We conclude that circuits flagged as suspicious by antivirus software [6] should be quarantined to prevent them from running with other circuits or should undergo alternative mitigation methods, rather than simply applying higher optimization to remove the payload. The effectiveness of Qiskit's circuit optimizer demonstrates that optimization is a valuable layer of defense in a quantum computer's security design. However, cybersecurity must be multi-layered, as the DJ malware's ability to penetrate circuit optimization highlights the need for additional layers of defense.

A. Future Works

There is a need for more mitigation methods to combat this threat. We plan to explore more mitigation methods and devise a multi-layered defense that can successfully mitigate the deceptive malware shown in this paper. We also plan to expand the number of wrapping circuits, victim circuits, and payloads. As well as exploring more multi-tenant scenarios.

Acknowledgment

Jaden Hawley is currently supported by the National Science Foundation under the grant number DEG-1946619.

References

[1] P. Das, S. S. Tannu, P. J. Nair, and M. Qureshi, "*A case for multi-programming quantum computers*", International Symposium on Microarchitecture, p.291–303. 2019

[2] X. Dou and L. Liu, "*A new qubits mapping mechanism for multi-programming quantum computing*", International Conference on Parallel Architectures and Compilation Techniques, p. 349–350, 2020

[3] A. Ash-Saki, M. Alam, S. Ghosh, "*Analysis of Crosstalk in NISQ Devices and Security Implications in Multi-programming Regime*", ACM ISLPED, pp. 25–30, 2020

[4] K. M. Rudinger, M. Sarovar, D. Langharst, T. J. Proctor, K. Young, E. Nielsen, and R. J. Blume-Kohout, "*Classifying and diagnosing crosstalk in quantum information processors*", 2018, Available: https://www.osti.gov/biblio/1513744

[5] S. Deshpande, C. Xu, T. Trochatos, Y. Ding, J. Szefer, "*Towards an Antivirus for Quantum Computers*" IEEE International Symposium on Hardware Oriented Security and Trust, pp. 37–40, 2022.

[6] S. Deshpande, C. Xu, T. Trochatos, H. Wang, F. Erata, S. Han, Y. Ding, J. Szefer, "*Design of Quantum Computer Antivirus*" IEEE International Symposium on Hardware Oriented Security and Trust, pp. 260-270, 2023

[7] IBM "*Compilation Routines*", IBM Quantum Documentation, Accessed 2024, Available: https://docs.quantum.ibm.com/api/qiskit/compiler

[8] P. Murali, D. C. McKay, M. Martonosi, A. Javadi-Abhari, "*Software Mitigation of Crosstalk on Noisy Intermediate-Scale Quantum Computers*", arXiv, 2020

[9] A. A. Saki, R. O. Topaloglu, S. Ghosh, "*Shuttle-Exploiting Attacks and Their Defenses in Trapped-Ion Quantum Computers*", IEEE Access, Vol. 10, pp. 2686–2699, 2022

[10] A. A. Saki, R. O. Topaloglu, S. Ghosh, "*Muzzle the Shuttle: Efficient Compilation for Multi-Trap Trapped-Ion Quantum Computers*", Design, Automation and Test in Europe Conference, pp. 322–327, 2022