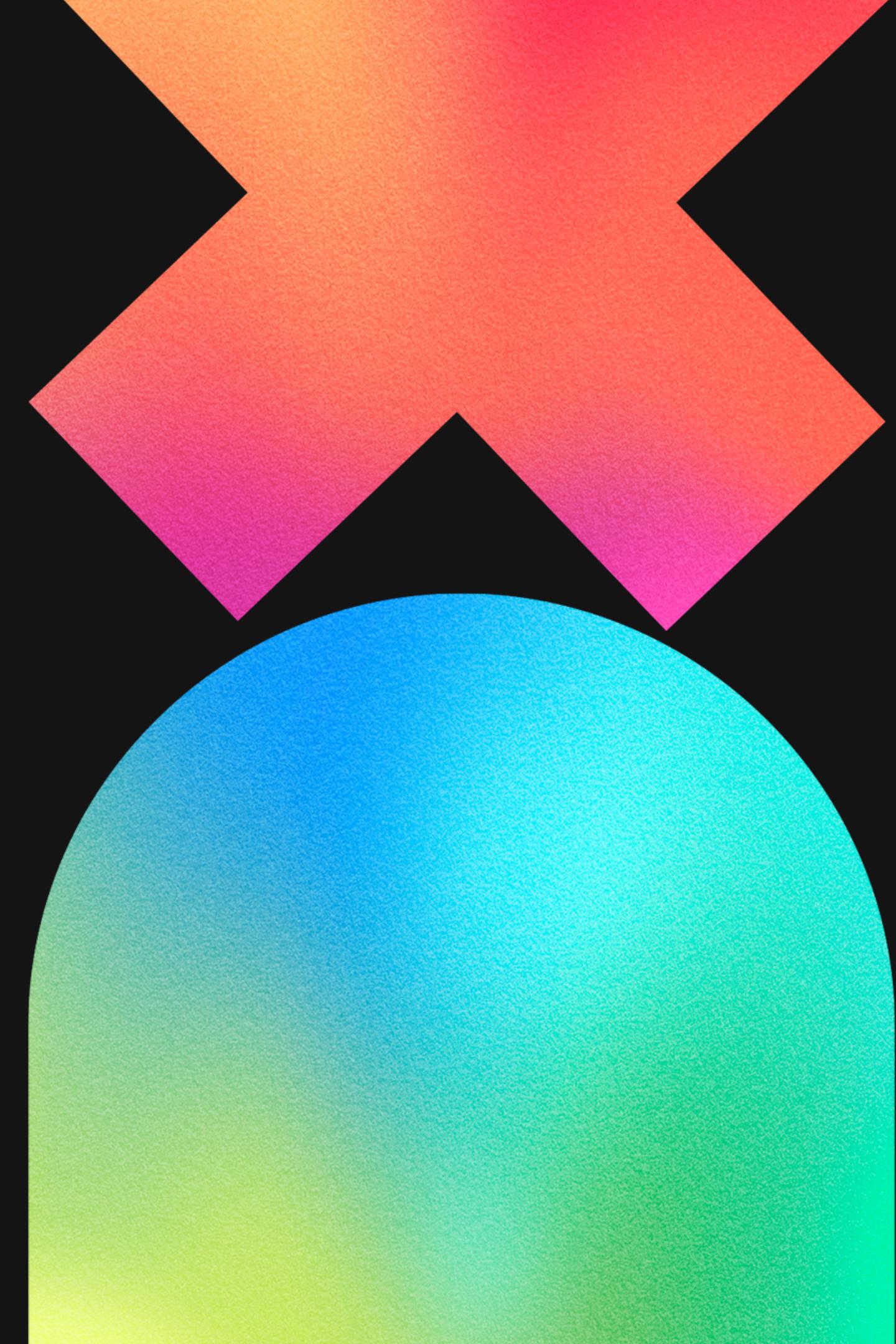


NET SHELL

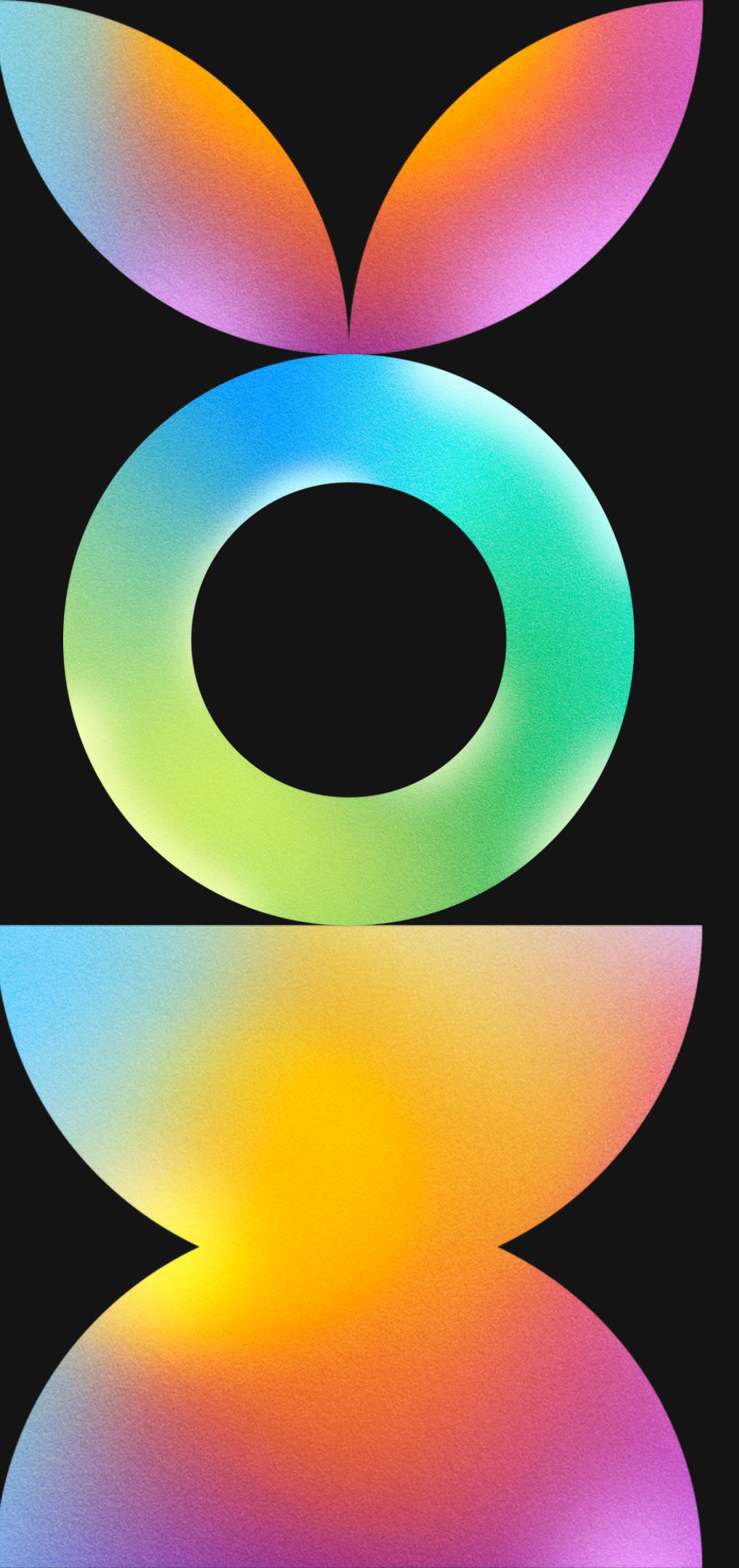
Gateway to remote system
administration

SACHIN PANDEY
SANSKRITI YADAV
SANYAM JAIN



OVERVIEW

The Reverse Shell project is a tool for remote access and administration of a target system. It allows a remote user to access the target system's shell, run commands, and transfer files between systems. The project is implemented in Python and supports multiple protocols, including TCP and HTTP.



OBJECTIVES



- To develop a reverse shell script that can establish a connection between an attacker computer and a victim computer.
- To enable remote execution of commands on the victim computer.
- To evaluate the performance of the script in terms of connection speed, bandwidth usage, stability, reliability, and security.
- To test the script on different operating systems and computers.

DATA RESOURCES

- Operating System: Linux (attacker computer), Windows (victim computer)
- Programming Language: Python
- Tools and Libraries: Netcat
- Test Computers: Dell Inspiron 15 (Windows 10), Lenovo IdeaPad (Ubuntu 18.04)
- Test Network: WAN
- Data: Various commands and messages exchanged between the attacker and victim computers during testing and evaluation.



METHODOLOGY

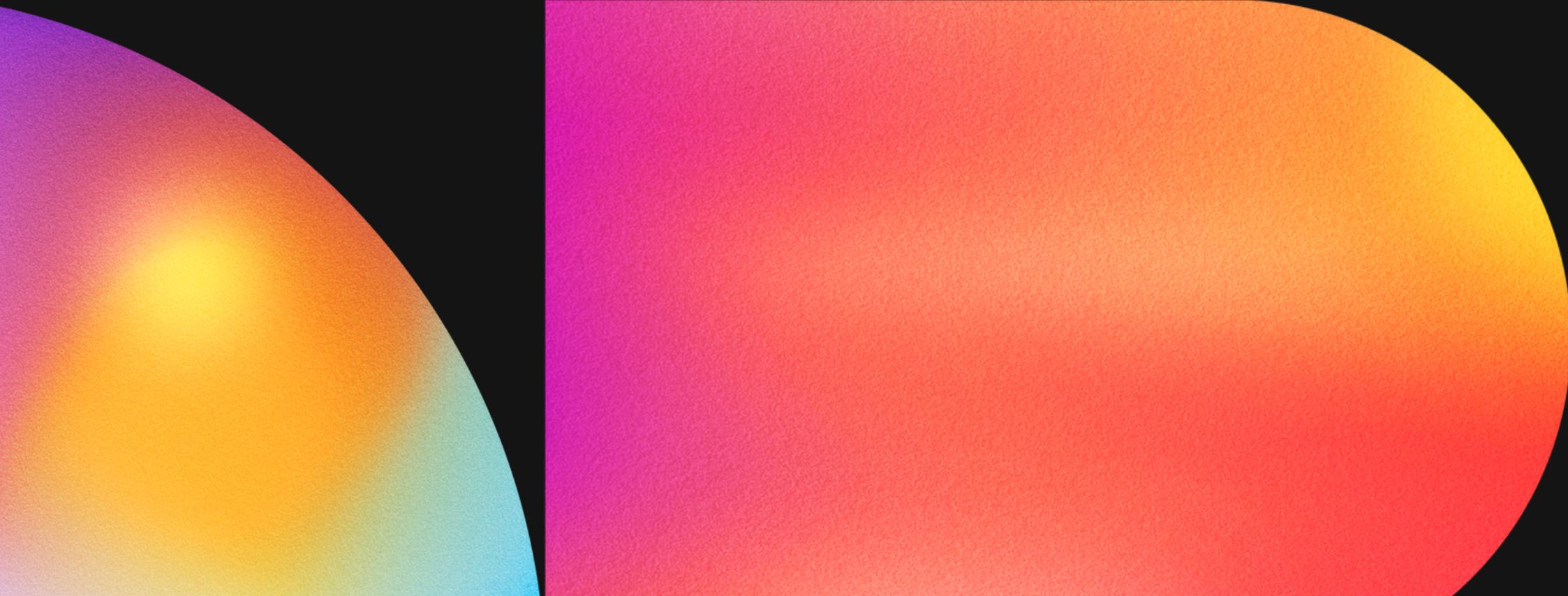
- Briefly elaborate on what you want to discuss.
- Briefly elaborate on what you want to discuss.
- Briefly elaborate on what you want to discuss.

RESULTS ACHIEVED

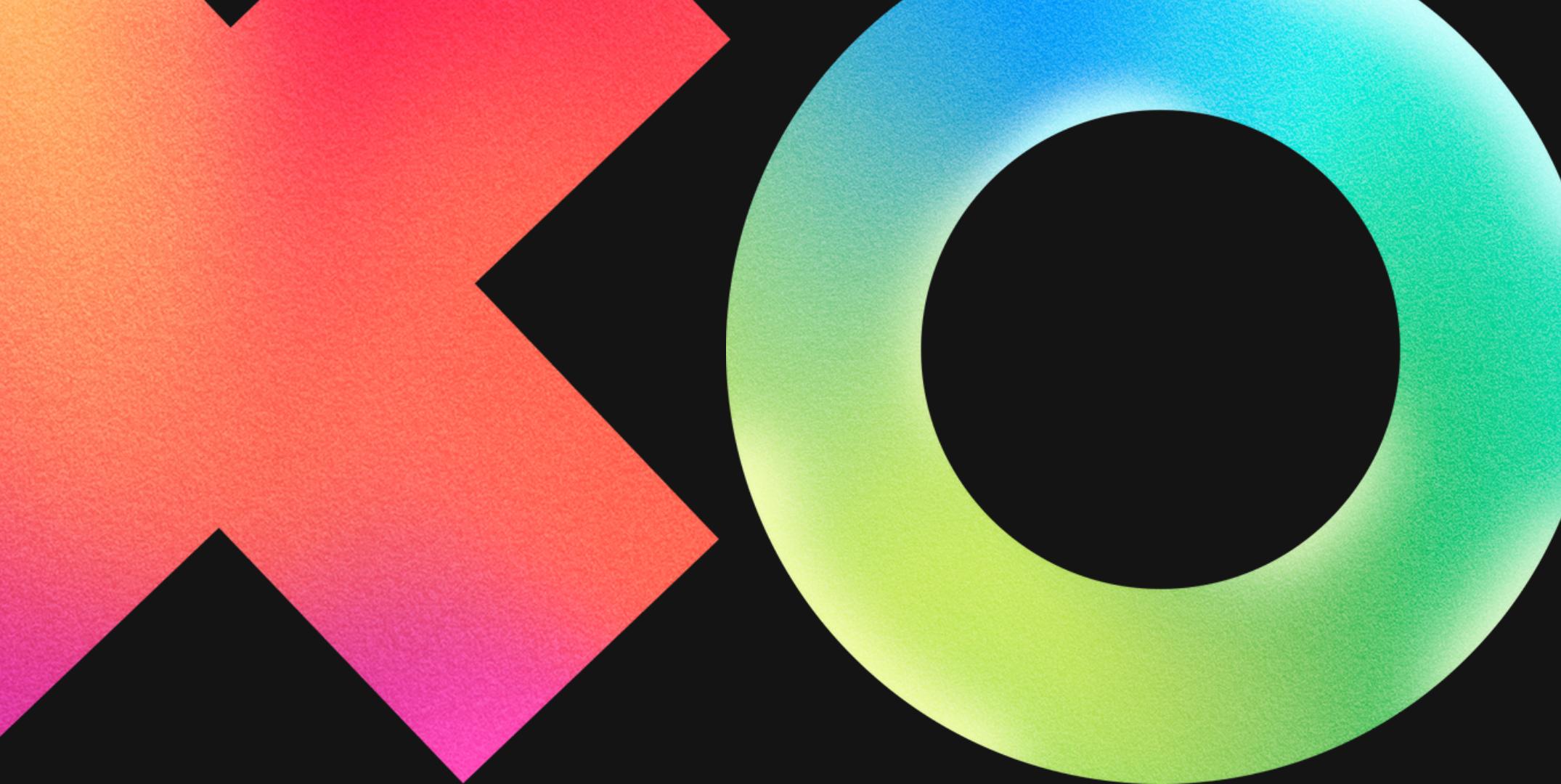
- Reverse shell script implemented in Python and tested on Linux and Windows.
- Successful remote connection and command execution achieved.
- Low bandwidth usage and reliable connections demonstrated.
- Performance evaluated and presented through visual aids.
- Conclusion: script achieved intended purpose.
- Future work: improve functionality and security, add more features, and test on larger scale.

PERFORMANCE EVALUATION

- Connection speed, bandwidth usage, stability, reliability, and security evaluated.
- Low bandwidth usage and stable connections demonstrated.
- Script found to be reliable and secure.
- Overall, script performed efficiently and reliably.



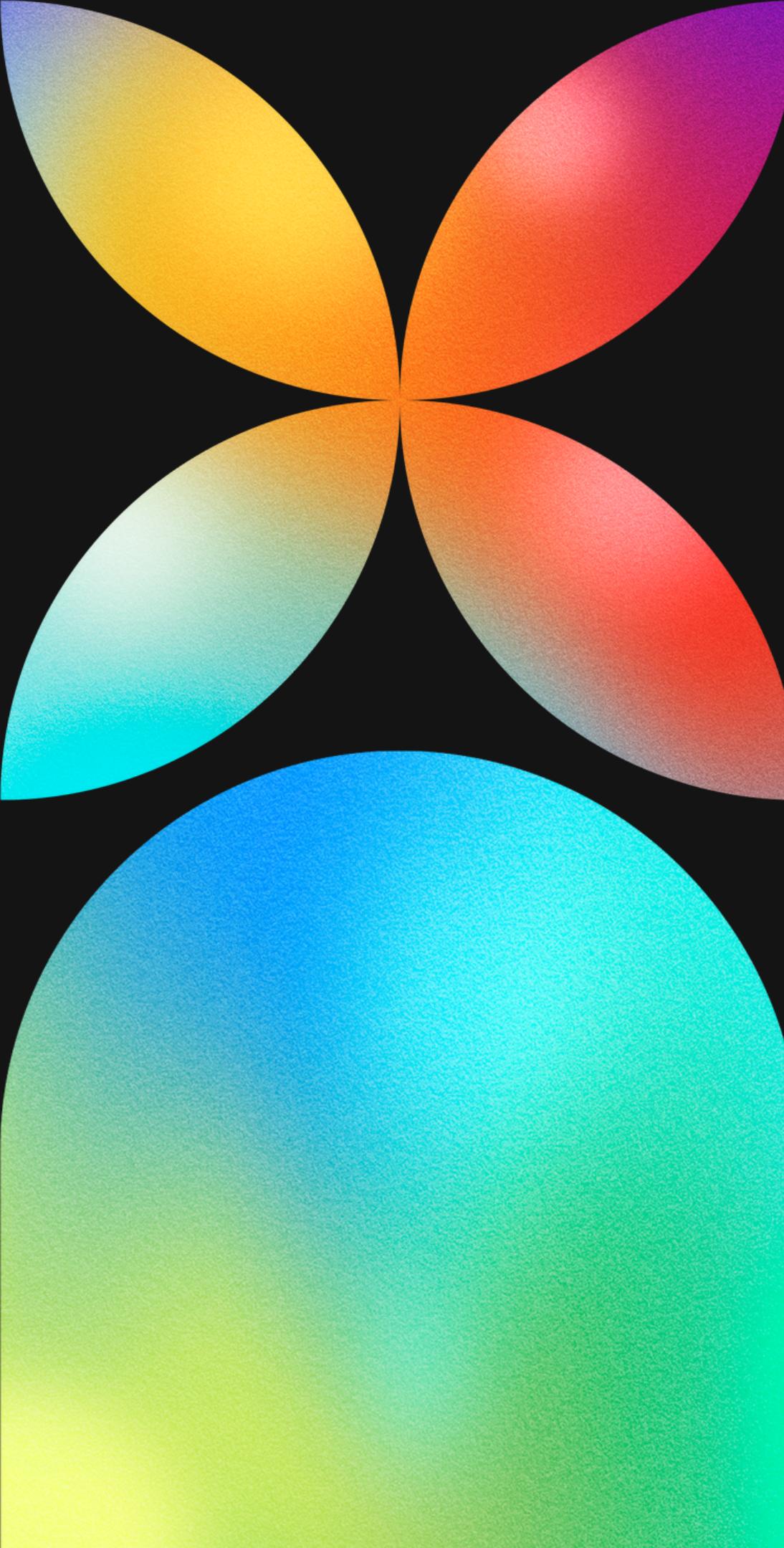
CONCLUSION



In summary, the reverse shell script developed using Python successfully established a remote connection with a victim computer and executed commands. The script's performance was efficient and reliable, with low bandwidth usage and secure connections. With future improvements, the script has the potential to become even more effective and secure. This project showcases the power of reverse shell scripting as a tool for remote administration and control.

REFERENCES

- ·<https://github.com/swisskyrepo/PayloadsAllTheThings>



THANK YOU

Any Questions?