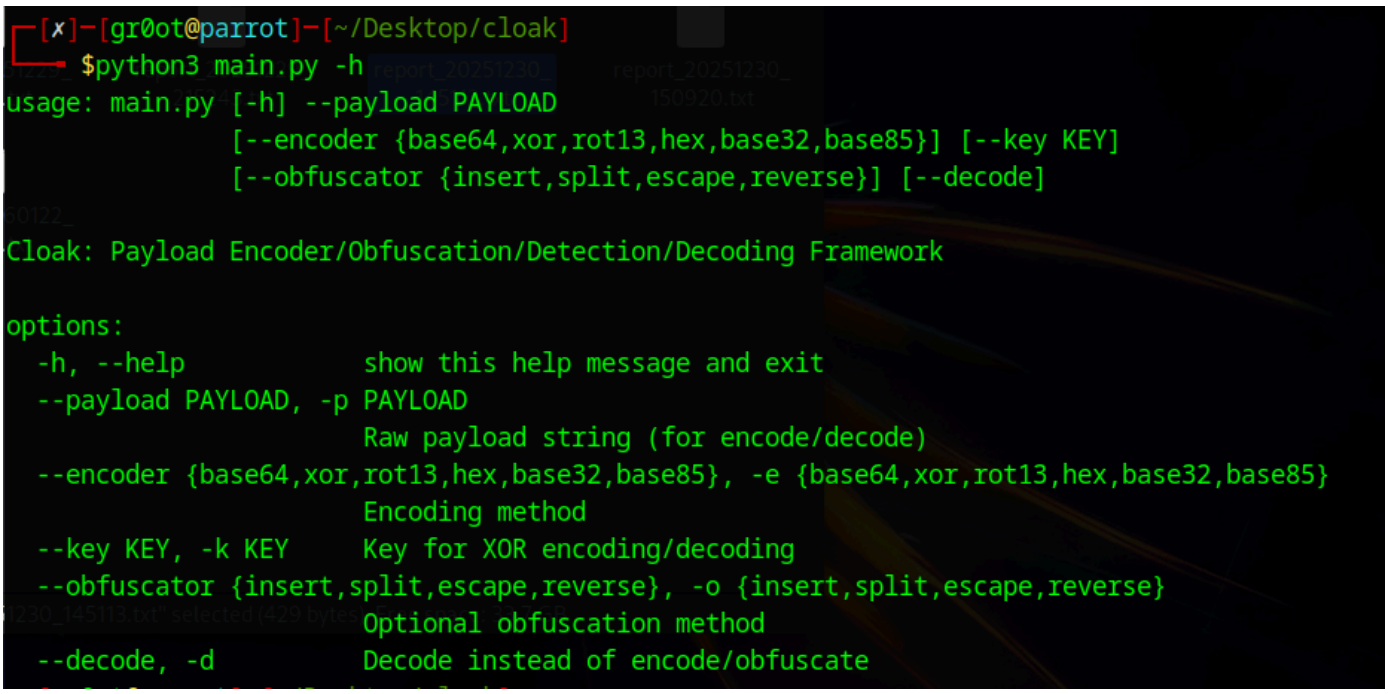
cloak_ss_file

Cloak: Payload Encoder, Obfuscation, Detection, Reporting, and Decoding Framework

1. **Help Options :**

   This screenshot shows the available command-line arguments for Cloak. It demonstrates how users can select payloads, encoders, obfuscators, and the decode flag.



---

2. **Normal String Encoding & Decoding with Obfuscation** :

   Here, a simple string is encoded using Base64 with an obfuscation method applied. The same string is then decoded back, showing that reversible transformations work correctly.

```
┌─[gr0ot@parrot]─[~/Desktop/cloak]
└──╼ $python3 main.py --p "hello" -e base64 -o reverse
=== Cloak Execution ===
[ENCODER] BASE64
[OBFUSCATOR] reverse
[OUTPUT] =8GbsVGa
[DETECTION] Original: BYPASSED
[DETECTION] Transformed: BYPASSED
[EFFECTIVENESS] NEUTRAL — Both original and transformed bypassed.

--- Report saved to results/report_20260204_143730.txt ---
┌─[gr0ot@parrot]─[~/Desktop/cloak]
└──╼ $python3 main.py --p "=8GbsVGa" -e base64 -o reverse -d
=== Cloak Decoding ===
[DECODER] BASE64
[DE-OBFUSCATOR] reverse
[OUTPUT] hello
```

---

3. **Detectable String Encoding & Decoding with Obfuscation** :

   This example uses a payload that would normally be flagged by detection. After encoding and obfuscation, the transformed payload bypasses detection. The next run shows successful decoding back to the original string.

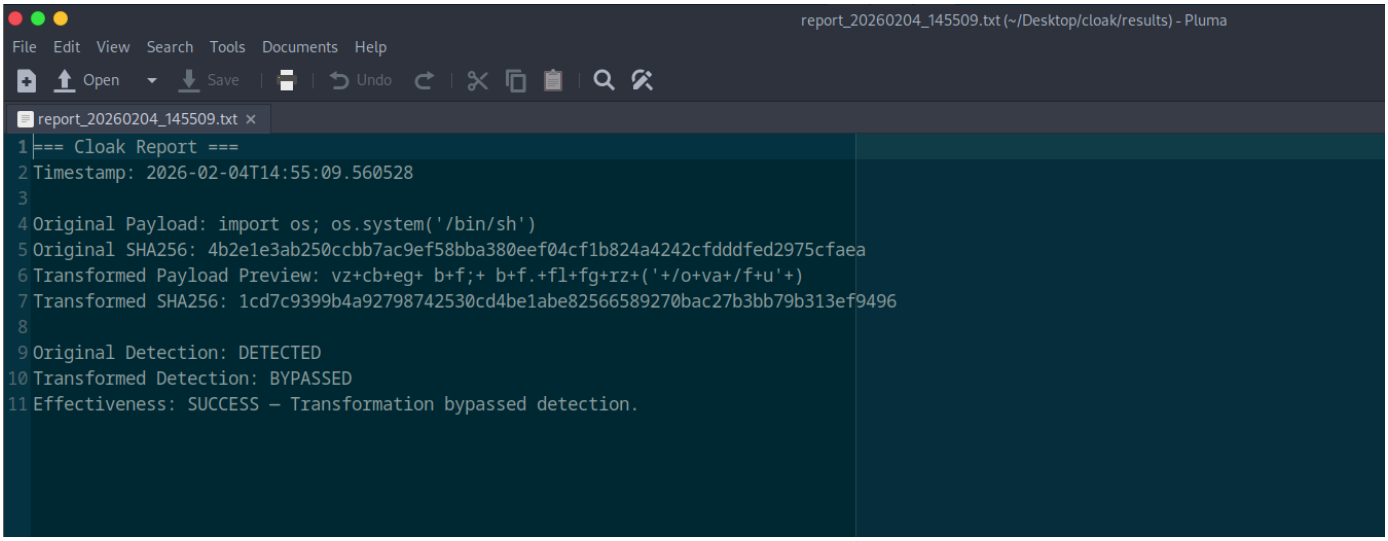
```
┌─[gr0ot@parrot]─[~/Desktop/cloak]
└──╼ $python3 main.py -p "import os; os.system('/bin/sh')" -e rot13 -o split
=== Cloak Execution ===
[ENCODER] ROT13
[OBFUSCATOR] split
[OUTPUT] vz+cb+eg+ b+f;+ b+f.+fl+fg+rz+('+/o+va+/f+u'+)
[DETECTION] Original: DETECTED
[DETECTION] Transformed: BYPASSED
[EFFECTIVENESS] SUCCESS — Transformation bypassed detection.

--- Report saved to results/report_20260204_145509.txt ---
┌─[gr0ot@parrot]─[~/Desktop/cloak]
└──╼ $python3 main.py -p "vz+cb+eg+ b+f;+ b+f.+fl+fg+rz+('+/o+va+/f+u'+)" -e rot13 -o split -d
=== Cloak Decoding ===
[DECODER] ROT13
[DE-OBFUSCATOR] split
[OUTPUT] import os; os.system('/bin/sh')
```

---

4. **Saved Report Output** :

   This screenshot highlights Cloak's reporting feature. Each run generates a detailed report file with detection results, verdicts, and hashes, saved automatically in the results/ directory.



5. **Results Directory Overview** :

   This view shows the results/ folder where all generated reports are stored. It confirms that Cloak maintains organized output files for later review.