

Date: 11/28/2023

Tester: Daryl Mokolo

CWE-327 HIGH: Use of a Broken or Risky Cryptographic Algorithm

Threat overview

After analyzing the system, I found that symmetric encryption file ("symmetric.enc") exhibits a vulnerability associated with the use of a weak encryption algorithm. The specific weakness identified is the utilization of an insecure encryption method, as revealed through frequency analysis.

Affected:

Users data and sensitive information

Steps to recreate the vulnerability:

Step 1:

Conduct the a frequency analysis using the command: `./ciphertools.py analyze -f symmetric.enc` that use a python frequency analysis tool to determine potential keys.

```
OpenSSH SSH client
permitted by applicable law.
Last login: Tue Nov 28 01:46:15 2023 from 72.106.55.223
Daryl.Mokolo@sec450 ~ % cd test
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py analyze -f symmetric.enc
Begin Frequency Analysis:
SYMBOL  HEX VALUE  COUNT  FREQ(%)
g        0x67      655    14.58%
~        0xac      380     8.46%
"        0xa8      302     6.72%
"        0xb0      293     6.52%
"        0xbb      282     6.28%
"        0xb6      254     5.66%
"        0xba      246     5.48%
µ        0xb5      243     5.41%
!        0xb9      199     4.43%
!        0xb3      159     3.54%
"        0xab      136     3.03%
#        0xaa      136     3.03%
-        0xaf      111     2.47%
k        0xbc      103     2.29%
-        0xb7      85      1.89%
-        0xb4      77      1.71%
-        0xad      73      1.63%
e        0xae      71      1.58%
A        0xc0      66      1.47%
0        0xa9      59      1.31%
Q        0x51      49      1.09%
s        0x73      44      0.98%
%        0x9a      41      0.91%
%        0xbd      39      0.87%
u        0x75      38      0.85%
k        0xbe      32      0.71%
9b       11      0.69%
        0x8c      23      0.51%
        0x96      21      0.47%
        0x88      21      0.47%
        0xbd      12      0.27%
        0x8a      12      0.27%
        0x94      11      0.24%
        0x99      11      0.24%
z        0xb2      11      0.24%
        0x93      9       0.20%
A        0xc1      9       0.20%
        0x97      8       0.18%
        0x8b      8       0.18%
        0x9c      8       0.18%
        0x8e      8       0.18%
t        0x74      8       0.18%
        0x89      6       0.13%
```

Step 2:

After Analyzing frequency which revealed weaknesses, allowing me to determine that 'g' corresponds to a space character and '~' corresponds to the letter 'e.' Then calculating the hex value which will

determine the main key by subtracting the value of g which is 103 and space = 32 from the ASCII table and got the value 71 which is 0x47 in hex value. Hence using this key in the command:

```
./ciphertools.py cipher --cipher symmetric --mode decrypt --file symmetric.enc --key 0x47 --modulo 0xff
```

The symmetric.enc file could be decrypted since the correct key was entered.

```
OpenSSH SSH client
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py cipher --cipher symmetric --mode decrypt --file symmetric.enc --key 0x47 --modulo 0xff
Simple Sabotage Field Manual
Office of Strategic Services

OSS REPRODUCTION BRANCH
SIMPLE SABOTAGE FIELD MANUAL
Strategic Services
(Provisional)
STRATEGIC SERVICES FIELD MANUAL No. 3

Office of Strategic Services
Washington, D. C.
17 January 1944

This Simple Sabotage Field Manual Strategic Services (Provisional) is published for the information and guidance of all concerned and will be used as the basic doctrine for Strategic Services training for this subject.

The contents of this Manual should be carefully controlled and should not be allowed to come into unauthorized hands.

The instructions may be placed in separate pamphlets or leaflets according to categories of operations but should be distributed with care and not broadly. They should be used as a basis of radio broadcasts only for local and special cases and as directed by the theater commander.

AR 380-5, pertaining to handling of secret documents, will be complied with in the handling of this Manual.
[Illustration]

William J. Donovan
CONTENTS
1. INTRODUCTION
2. POSSIBLE EFFECTS
3. MOTIVATING THE SABOTEUR
4. TOOLS, TARGETS, AND TIMING
5. SPECIFIC SUGGESTIONS FOR SIMPLE SABOTAGE

1. INTRODUCTION

The purpose of this paper is to characterize simple sabotage, to outline its possible effects, and to present suggestions for inciting and executing it.

Sabotage varies from highly technical coup de main acts that require detailed planning and the use of specially-trained operatives, to innumerable simple acts which the ordinary individual citizen-saboteur can perform. This paper is primarily concerned with the latter type. Simple sabotage does not require specially prepared tools or equipment; it is executed by an ordinary citizen who may or may not act individually and without the necessity for active connection with an organized group; and it is carried out in such a way as to involve a minimum danger of injury, detection, and reprisal.

Where destruction is involved, the weapons of the citizen-saboteur are salt, nails, candles, pebbles, thread, or any other materials he might normally be expected to possess as a householder or as a worker in his particular occupation. His arsenal is the kitchen shelf, the trash pile, his own usual kit of tools and supplies. The targets of his sabotage are usually objects to which he has normal and inconspicuous access in everyday life.
```

```
OpenSSH SSH client

This Simple Sabotage Field Manual Strategic Services (Provisional) is published for the information and guidance of all concerned and will be used as the basic doctrine for Strategic Services training for this subject.

The contents of this Manual should be carefully controlled and should not be allowed to come into unauthorized hands.

The instructions may be placed in separate pamphlets or leaflets according to categories of operations but should be distributed with care and not broadly. They should be used as a basis of radio broadcasts only for local and special cases and as directed by the theater commander.

AR 380-5, pertaining to handling of secret documents, will be complied with in the handling of this Manual.
[Illustration]

William J. Donovan
CONTENTS
1. INTRODUCTION
2. POSSIBLE EFFECTS
3. MOTIVATING THE SABOTEUR
4. TOOLS, TARGETS, AND TIMING
5. SPECIFIC SUGGESTIONS FOR SIMPLE SABOTAGE

1. INTRODUCTION

The purpose of this paper is to characterize simple sabotage, to outline its possible effects, and to present suggestions for inciting and executing it.

Sabotage varies from highly technical coup de main acts that require detailed planning and the use of specially-trained operatives, to innumerable simple acts which the ordinary individual citizen-saboteur can perform. This paper is primarily concerned with the latter type. Simple sabotage does not require specially prepared tools or equipment; it is executed by an ordinary citizen who may or may not act individually and without the necessity for active connection with an organized group; and it is carried out in such a way as to involve a minimum danger of injury, detection, and reprisal.

Where destruction is involved, the weapons of the citizen-saboteur are salt, nails, candles, pebbles, thread, or any other materials he might normally be expected to possess as a householder or as a worker in his particular occupation. His arsenal is the kitchen shelf, the trash pile, his own usual kit of tools and supplies. The targets of his sabotage are usually objects to which he has normal and inconspicuous access in everyday life.

A second type of simple sabotage requires no destructive tools whatsoever and produces physical damage, if any, by highly indirect means. It is based on universal opportunities to make faulty decisions, to adopt a noncooperative attitude, and to induce others to follow suit. Making a faulty decision may be simply a matter of placing tools in one spot instead of another. A non-cooperative attitude may involve nothing more than creating an unpleasant situation among one's fellow workers, engaging in bickerings, or displaying surliness and stupidity.

This type of activity, sometimes referred to as the "human element," is frequently responsible for accidents, delays, and general obstruction even under normal conditions. The potential saboteur should discover what types of faulty decisions and the operations are normally found in this kind of work and should then devise his sabotage so as to enlarge that "margin for error."

2. POSSIBLE EFFECTS

Acts of simple sabotage are occurring throughout Europe. An effort should be made to add to their efficiency, lessen their detectability, and increase their number. Acts of simple sabotage, multiplied by thousands of citizen-saboteurs, can be an effective weapon against the enemy. Slashing tires, draining fuel tanks, starting fires, starting arguments, acting stupidly, short-circuiting electric systems, abrading machine parts will waste materials, manpower, and time. Occurring on a wide scale, simple sabotage will be a constant and tangible drag on the war effort of the enemy.

Simple sabotage may also have secondary results of more or less value. Widespread practice of simple sabotage will harass and demoralize enemy administrators and police. Further, success may embolden the citizen-saboteur eventually to find colleagues who can assist him in sabotage of greater dimensions. Finally, the very practice of simple sabotage by natives in enemy or occupied territory may make these individuals identify themselves actively with the United Nations war effort, and encourage them to assist openly in periods of Allied invasion and occupation.

Daryl.Mokolo@sec450 ~/test %
```

Remediation:

Upgrading the encryption algorithm to a more secure symmetric encryption algorithm, such as AES-256, to mitigate the identified vulnerabilities will greatly improve the file security. In addition, implementing a robust key management system by including regular key rotation and secure storage will strengthen overall cryptographic controls to prevent future vulnerabilities.

CWE-326 HIGH: Inadequate Encryption Strength**Threat overview**

The asymmetric encryption file ("asymmetric.enc") reveals a vulnerability linked to weak key management and a potential flaw in the chosen asymmetric encryption algorithm.

Affected:

Users data and sensitive information

Steps to recreate the vulnerability:**Step 1:**

Similarly to the symmetrical encryption that was decrypted earlier, the "asymmetric.enc" file exhibited certain patterns after the cipher frequency analysis was done. Hence, specific identification of characters and their probable plaintext equivalents was found. The character 's' in the ciphertext consistently mapped to a plaintext space character, revealing a potential weakness. By entering the commands:

```
./ciphertools.py analyze -f asymmetric.enc
```

```

OpenSSH SSH client
the citizen-saboteur eventually to find colleagues who can assist him in sabotage of greater dimensions. Finally, the very practice of simple sabotage by natives in enemy
ake these individuals identify themselves actively with the United Nations war effort, and encourage them to assist openly in periods of Allied invasion and occupation.
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py analyze -f asymmetric.enc
Traceback (most recent call last):
  File "/home/Daryl.Mokolo/test/./ciphertools.py", line 325, in <module>
    do_args(mode, parsed)
  File "/home/Daryl.Mokolo/test/./ciphertools.py", line 311, in do_args
    message = get_message(infile)
              ^^^^^^^^^^^^^^^^^
  File "/home/Daryl.Mokolo/test/./ciphertools.py", line 94, in get_message
    with open(path, "rb") as f:
         ^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'asymmetric.enc'
Daryl.Mokolo@sec450 ~/test % ls
asymmetric.enc  ciphertools.py  group2.zip  symmetric.enc
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py analyze -f asymmetric.enc
Begin Frequency Analysis:
SYMBOL  HEX_VALUE  COUNT  FREQ(%)
s       0x73      432   17.31%

  0x85      273   10.94%
  0xb2      171    6.85%
  0xce      159    6.37%
  0x3c      145    5.81%
  0xa0      140    5.61%
  0x4e      136    5.45%
  0x05      134    5.37%
  0x57      131    5.25%
  0x8e      100    4.01%
  0xd7       85    3.41%
  0x45       68    2.72%
  0x2a       62    2.48%
  0xf2       51    2.04%
  0x33       51    2.04%
  0x17       49    1.96%
  0x60       48    1.92%
  0xe0       36    1.44%
  0x7c       36    1.44%
  0x0e       33    1.32%
  0xfb       30    1.20%
  0xb0       29    1.16%
  0xd9       26    1.04%
  0x35       13    0.52%
  0x86       12    0.48%
  0x97        9    0.36%
  0xe9        4    0.16%
  0xfc        4    0.16%
  0xc4        3    0.12%
Daryl.Mokolo@sec450 ~/test %

```

./ciphertools.py crack --cipher-byte 0xa3 --plaintext-guess 0x20 --modulo 0xff

Hence, The subsequent application of a brute-force attack using the command : **./ciphertools.py crack** identified a limited set of potential keys that eventually led lead to the plain text decrypted message.

```

OpenSSH SSH client
Daryl.Mokolo@sec450 ~/test %
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0xa3 --plaintext-guess 0x20 --modulo 0xff
Examined 30 potential keys
Given cipher byte 0xa3 and plaintext guess 0x20 the encryption key would be 0x2c
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x2c --plaintext-guess 0x20 --modulo 0xff
Examined 98 potential keys
Given cipher byte 0x2c and plaintext guess 0x20 the encryption key would be 0xa3
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0xa3 --plaintext-guess 0x2c --modulo 0xff
Examined 138 potential keys
Given cipher byte 0xa3 and plaintext guess 0x2c the encryption key would be 0xbc
Daryl.Mokolo@sec450 ~/test %
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0xa3 --plaintext-guess 0x33 --modulo 0xff
Examined 255 potential keys 0.04%
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x33 --plaintext-guess 0x2c --modulo 0xff
Examined 255 potential keys 9466866392
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x2c --plaintext-guess 0xa3 --modulo 0xff
Examined 189 potential keys
Given cipher byte 0x2c and plaintext guess 0xa3 the encryption key would be 0x89
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x2c --plaintext-guess 0x89 --modulo 0xff
Examined 113 potential keys
Given cipher byte 0x2c and plaintext guess 0x89 the encryption key would be 0x94
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x2c --plaintext-guess 0x94 --modulo 0xff
Examined 84 potential keys
Given cipher byte 0x2c and plaintext guess 0x94 the encryption key would be 0xd4
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x2c --plaintext-guess 0xd4 --modulo 0xff
Examined 83 potential keys
Given cipher byte 0x2c and plaintext guess 0xd4 the encryption key would be 0xc
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x2c --plaintext-guess 0xc --modulo 0xff
Examined 39 potential keys
Given cipher byte 0x2c and plaintext guess 0xc the encryption key would be 0x2f
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x2f --plaintext-guess 0xc --modulo 0xff
Examined 30 potential keys
Given cipher byte 0x2f and plaintext guess 0xc the encryption key would be 0x2c
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x2c --plaintext-guess 0x2c --modulo 0xff
Examined 2 potential keys
Given cipher byte 0x2c and plaintext guess 0x2c the encryption key would be 0x1
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x2c --plaintext-guess 0x1 --modulo 0xff
Examined 45 potential keys
Given cipher byte 0x2c and plaintext guess 0x1 the encryption key would be 0x1d
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x2c --plaintext-guess 0x1d --modulo 0xff
Examined 152 potential keys
Given cipher byte 0x2c and plaintext guess 0x1d the encryption key would be 0x4c
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x2c --plaintext-guess 0x4c --modulo 0xff
Examined 15 potential keys
Given cipher byte 0x2c and plaintext guess 0x4c the encryption key would be 0xa4
Daryl.Mokolo@sec450 ~/test % ./ciphertools.py crack --cipher-byte 0x2c --plaintext-guess 0xa4 --modulo 0xff
Examined 107 potential keys

```

Remediation:

The best remediation to improve these files security is to transition to a stronger asymmetric encryption algorithm, such as RSA with a larger key size, to enhance encryption strength. In addition, Ensuring the alignment with industry best practices and cryptographic standards.

Reference:

[CWE - CWE-347: Improper Verification of Cryptographic Signature \(4.13\) \(mitre.org\)](#)

[CWE - CWE-327: Use of a Broken or Risky Cryptographic Algorithm \(4.13\) \(mitre.org\)](#)