Pablo Gagliardi

Evaluación continua criptografía (auxiliar).

Consigna 1:
Qué significan los primeros 2 valores que están a continuación del signo $?

yj9t$y8fiVkfkbfem7sKXGHh01$9vU9IRpnJkkD6mCE1KD2nFBRxpgqdRrSnS6RWfhisq1

1LJTtsJB7$bIL9zq515waTDo88oDrc40

idsalt$hashed/encripted

`$<id>[$<param>=<value>(,<param>=<value>)*][$<salt>[$<hash>]]`

donde:

- `id`: En el primer campo, aparece el tipo de hash que ha sido utilizado, en este caso $1 para MD5 (unix) y $y para yescript
- `Nombre y valor del parámetro`: Parámetros de complejidad de hash.

Consigna 2: Descifrar los mensaje cifrado en cifrado cesar.

iwxsiwyqiniptoshigmjvehspsqseojefixsxmtsgiwev
Descifrado con una rotación de 3 letras.
“Esto es un ejelmplo de cifrado mono alfabeto tipo cesar”

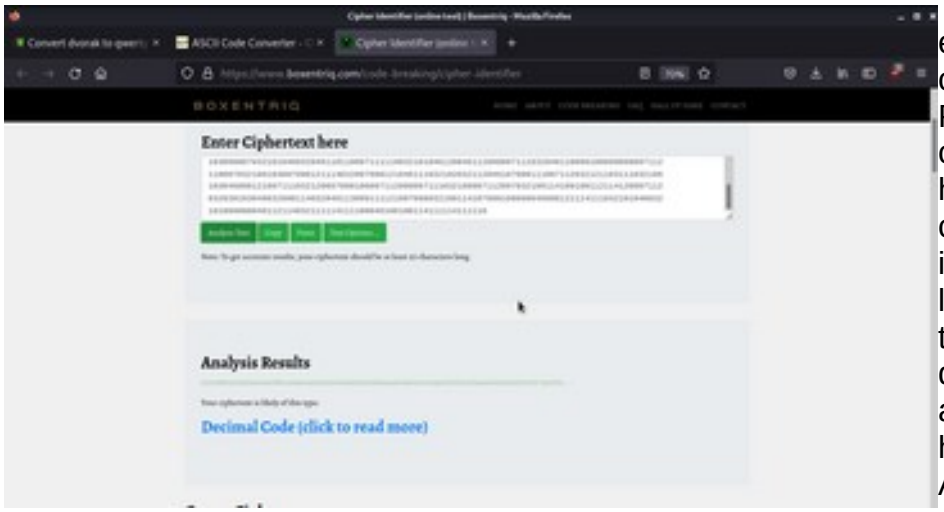
fx afxez azdsesgz od colvsklc vl oglvflnszx nzxesxfl
Descifrado con una rotación de 11 letras.
“Un punto positivo es realizar la evaluacion continua”

Consigna 3:
Se intercepta un mensaje que inició un emisor para un receptor, el mensaje contiene informacion para usuarios bancarios que utilizan apps en sus celulares

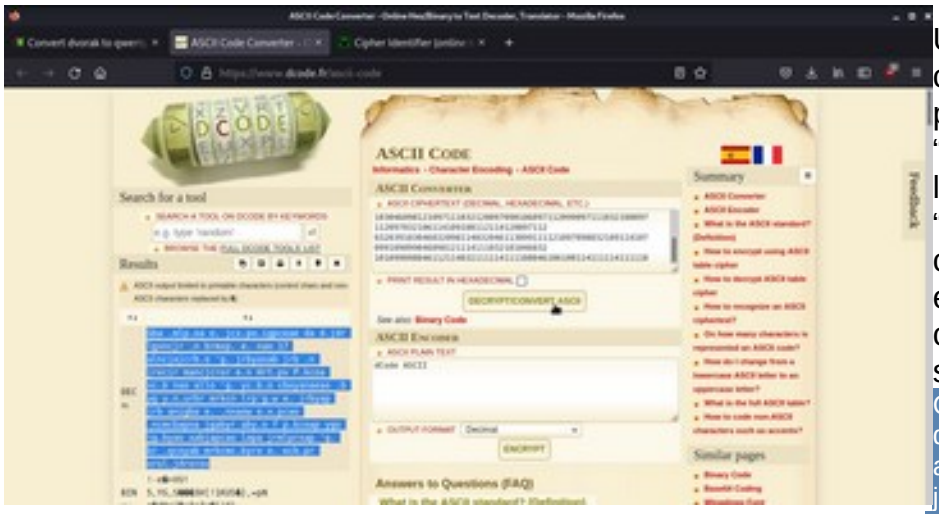
Pista: Mensaje escrito con un teclado querty, con un SO X cuyo teclado está configurado en dvorak

071098097032046109108112046111097032101046032106099120046112111046105103112099101097101032100
097032100046106100114032108103120110099106114032046110032098114109120112046032101046032110097
111032049055032097108110099106097106099114098046111032039103046032106114098120971209709803 2
106114098032046110032106114101099105114032109097110099106099114111114032101046110032072114116
046112118032080046107099111097032120099046098032110097111032097108108111032039103046032121099
046098046111032099098111121097110097101097111032046098032121103032121046100461171140981140 32
109114107099110032108114112039103046119032101046032106114098121097112032106114098032097110105
103098097032101046032046110110097111119032101046120046112099097111032046100991090990981120 97112
110097032106103097098121114032097098121046111032102032112046107099111097112032121103111032106
103046098121097110321200970981060971120990971110321080971120970321061141091081121141200 97112
032039103046032098114032046113099111121097098032109114107099109099046098121114111032101046032
101099098046112114032111114111108046106100114111114111118

Resolución:



Tras algunos tropiezos, encontré el resultado con un camino de 3 simples pasos:
Primero analicé el tipo de codigo de cifrado con https://www.boxentriq.com/code-breaking/cipher-identifier
lo cual me indicó que se trataba de un código decimal, y me sugirió analizarlo con las herramientas de A1Z26 y de ASCII



Una vez realizado el testeo, cuyo resultado fue que probablemente fuese un "código decimal", me dirigí a la página "https://www.dcode.fr/ascii-code"
en la cual pude descifrar el código Ascii, el cual resultó ser el siguiente:
Gba .mlp.oa e. jcx.po.igpceae da d.jdr lgxncjr .n brmxp. e. nao 17 alncjajcrb.o 'g. jrbyaxab jrb .n jrecir mancjcror e.n Hrt.pv P.kcoa xc.b nao allo 'g. yc.b.o cboyanaeao .b yg y.n.urbr mrkcn lrp'g.w e. jrbyap jrb anigba e. .nnaow e.x.pcao .ncmcbapna jgabyr aby.o f p.kcoap ygo jg.byao xabjapcao lapa jrmlprxap 'g. br .qcoyab mrkcmc.byro e. ecb.pr orol.jdrorov

Por último, con la herramienta de teclados disponible en la página https://awsm-tools.com/text/keyboard-layout
pude "pasar" el texto escrito en dvorak a querty, dando como resultado el texto a continuación expuesto.

Una empresa de ciberseguridad ha hecho publico el nombre de las 17 aplicaciones que contaban con el codigo malicioso del Joker. Revisa bien las apps que tienes instaladas en tu telefono movil porque, de contar con alguna de ellas, deberias eliminarla cuanto antes y revisar tus cuentas bancarias para comprobar que no existan movimientos de dinero sospechosos.