CS5434 – Homework 4

Due Wednesday November 26th, 2014.

Problem 1

Consider the description of Blackhole v2 at http://blog.spiderlabs.com/2012/09/blackhole-exploit-kit-v2.html with a particular focus on the initial obfuscated javascript described there. Develop one or more snort signatures to detect this javascript.

Do you think this is a good strategy to detect Blackhole exploit attempts. Why or why not?

Problem 2

Take the heapspray code from http://www.thegreycorner.com/2010/01/heap-spray-exploit-tutorial-internet.html and get it to work in your browser (eg by loading an HTML file from disk). Note, not the exploit itself (which is very unlikely to work in a current browser) just the heapspray with enough initialization code to get it to run. Now vary the number of iterations in the main for() loop, and measure the size of the browser as a function of this loop count (eg using ps on a unix/Linux system) over a wide range of values. Can you determine the increments in which the browser gets memory from the OS? How much overhead is there between successive copies of the nop-sled/payload combination?

Problem 3

Suppose X is one of nine messages, A through H, where

$P(A) = P(B) = P(C) = P(D) = 1/16$

$P(E) = P(F) = P(G) = P(H) = 1/8$

$P(I) = 1/4.$

What is the entropy of the distribution?

Problem 4

Decrypt the following word, which was encrypted with a Caesar cipher of shift 3.

Dqwhgloxyldq

Problem 5

Suppose we have a strong cryptographic hash – a function which encrypts an input to produce a hash value of fixed size of *n* bits in way in which makes it extremely difficult to deliberately produce collisions. You may assume that the output has perfect entropy.

a) Suppose we have two completely different plaintexts. What is the probability that they will have the same hash, as a function of *n*?
b) Suppose that we have *m* completely different plaintexts. What is the probability that any two of them will have the same hash, as a function of *n* and *m*?

Problem 6

The following cipher-text was encrypted with a single-character substitution cipher of some kind. Decrypt it to determine the English plain-text.

Hint 1) Line breaks were not encrypted, but spaces and punctuation were.
Hint 2) The plain-text is famous.

```
dxwg,kwxbwvxdwdxwg,ttdqfdwrcwdq,wae,cdrxv*
 q,dq,bw;drcwvxgu,bwrvwdq,worvnwdxwce..,b
dq,wcurvpcwfvnwfbbx cwx.wxedbfp,xecw.xbdev,
xbwdxwdf-,wfbocwfpfrvcdwfwc,fwx.wdbxegu,c
fvnwgiwxyyxcrvpw,vnwdq,olwdxwnr,kwdxwcu,,ytt
vxwoxb,ttfvnwgiwfwcu,,ywdxwcfiw ,w,vn
dq,wq,fbdfmq,kwfvnwdq,wdqxecfvnwvfdebfuwcqxm-c
dqfdw.u,cqwrcwq,rbwdxlw;drcwfwmxvceoofdrxv
n,zxeduiwdxwg,w rcq,nlwdxwnr,kwdxwcu,,ytt
dxwcu,,ytty,bmqfvm,wdxwnb,fo*wfikwdq,b,;cwdq,wbegk
.xbwrvwdqfdwcu,,ywx.wn,fdqw qfdwnb,focwofiwmxo,
 q,vw ,wqfz,wcqe..u,nwx..wdqrcwoxbdfuwmxruk
oecdwprz,wecwyfec,lwdq,b,;cwdq,wb,cy,md
dqfdwof-,cwmfufordiwx.wcxwuxvpwur.,l
.xbw qxw xeunwg,fbwdq,w qrycwfvnwcmxbvcwx.wdro,k
dq;wxyyb,ccxb;cw bxvpkwdq,wybxenwofv;cwmxvdeo,ui
dq,wyfvpcwx.wn,cyrc,nwuxz,kwdq,wuf ;cwn,ufik
dq,wrvcxu,vm,wx.wx..rm,kwfvnwdq,wcyebvc
dqfdwyfdr,vdwo,brdwx.wdq;wev xbdqiwdf-,ck
```

 q,vwq,wqroc,u.worpqdwqrcwaer,decwof-,
 rdqwfwgfb,wgxn-rv*w qxw xeunw.fbn,ucwg,fbk
dxwpbevdwfvnwc ,fdwevn,bwfw ,fbiwur.,k
gedwdqfdwdq,wnb,fnwx.wcxo,dqrvpwf.d,bwn,fdqk
dq,wevnrcmxz,b,nwmxevdbikw.bxow qxc,wgxebv
vxwdbfz,uu,bwb,debvckwyejju,cwdq,w ruuk
fvnwof-,cwecwbfdq,bwg,fbwdqxc,wruucw ,wqfz,
dqfvw.uiwdxwxdq,bcwdqfdw ,w-vx wvxdwx.*
dqecwmxvcmr,vm,wnx,cwof-,wmx fbncwx.wecwfuuk
fvnwdqecwdq,wvfdrz,wqe,wx.wb,cxuedrxv
rcwcrm-ur,nwx;,bw rdqwdq,wyfu,wmfcdwx.wdqxepqdk
fvnw,vd,bybrc,wx.wpb,fdwyrdmqwfvnwoxo,vd
 rdqwdqrcwb,pfbnwdq,rbwmebb,vdcwdebvwf bi
fvnwuxc,wdq,wvfo,wx.wfmdrxvlwttwcx.dwixewvx k
dq,w.frbwxyq,urf*wttwvioyqkwrvwdqiwxbrcxvc
g,wfuuwoiwcrvcwb,o,og,b,nl