Algebraic Foundations for Quantitative Information Flow

Oxford October 2011

Pasquale Malacaria
Queen Mary University of London

The (little)Expectation Monad

Oxford October 2011

Pasquale Malacaria
Queen Mary University of London

measure change in confidential data due to possible observations of the system.

(a different angle: more oriented to application)

Quantitative analysis according to a measure F = difference of the measure F on the secret before and after running the program

$$\Delta_F(P, h) = F(h) - F(h|P)$$

1. P = r.v. observations about the program

2. h = r.v. secret data

possible choices for $F, F(-|-)$ are:

(A) for *uncertainty about the secret:* F and $F(-|-)$ are Shannon entropy and conditional entropy

(B) for *probability of guessing in one try:* (introduced by Smith and noted ME)

$$F(X) = -\log(\max_{x \in X} \mu(X = x)) \text{ and}$$

$$F(X|Y) = -\log(\sum_{y \in Y} \mu(y)(\max_{x \in X} \mu(X = x|Y = y)))$$

(C) for *the expected number of guesses:* (noted GE)

$$F(X) = \sum_{x_i \in X, i \geq 1} i \ \mu(X = x_i) \text{ and}$$

$$F(X|Y) = \sum_{y \in Y} \mu(y)(\sum_{x_i \in X, i \geq 1} i\mu(X = x_i|Y = y))$$

(here we assume $i < j$ implies $\mu(X = x_i) \geq \mu(X = x_j)$)

Few questions:

1. what these measures look like?

2. how they classify threats?

3. what do they have in common?

what these measures look like?

consider the following programs:

1. $M_1 \equiv$ `if(h == 1) then o = 0; else o = 1;`

2. $M_2 \equiv$ `o = h;`

`h` is a 2 bits secret (all values equally likely)
Initially

probability of guessing secret=0.25
av n of guesses=$1*0.25+2*0.25+3*0.25+4*0.25 = 2.5$
Entropy= $-4*(0.25\log(0.25)) = 2$

$G(h) = 0.25, NG(h) = 2.5, H(h) = 2$

$M_1 \equiv \texttt{if(h} == 1) \texttt{ then o} = 0; \texttt{else o} = 1;$
$\mu(h|o = 0) = 1, \mu(h|o = 1) = 0.33$

$M_2 \equiv \texttt{o} = \texttt{h};$
$\mu(h|o = 0) = \mu(h|o = 1) = \mu(h|o = 2) =$
$\mu(h|o = 3) = 1$

$M_1 \equiv$ `if(h == 1) then o = 0; else o = 1;`

	H	G	NG	ME	GE
M_1	0.8112	0.5	1.75	1	0.75
M_2	2	1	1	2	1.5

table 1: comparing measures

M_1: chances of guessing doubled from 0.25 to 0.5 (=G) so the rate of increase is $2^{ME} = G/G(h) = 2^1$; the average n of questions reduced by 0.75 (=GE) so that it will take $NG = 1.75$ tries. $H = 0.8112$ bits leaked.

$M_2 \equiv$ o = h;

	H	G	NG	ME	GE
M_1	0.8112	0.5	1.75	1	0.75
M_2	2	1	1	2	1.5

table 2: comparing measures

M_2: $H = 2$ everything is leaked: secret guessed in one try ($G = NG = 1$). Chances increased 4 folds ($2^{ME} = \frac{1}{0.25} = 2^2$). Average number of questions reduced by 1.5 ($= GE$) to one ($NG = 1$).

How do they classify threats?

how to define a "more secure" order between programs P, P'?

$$(A): \ P \leq_F P' \iff \forall \mu(h).\, \Delta_F(P; h) \leq \Delta_F(P'; h)$$

Is this "more secure" order depending on the choice of F?

It turns out that order (A) is the same for all measures F.

It is the order in the Lattice of Information (LOI)

LOI= lattice of all partitions (eq. rel.) on a set of atoms. Is a complete lattice with ordering:

$$X \leq_L Y \iff y \simeq_Y y' \Rightarrow y \simeq_X y'$$

assume a distribution on the atoms then we can see LOI as a lattice of random variables....

$$\mu(X = x) = \sum\{\mu(x_i) | x_i \in x\}$$

strictly speaking is the set theoretical kernel of a r.v. (but as we don't need the values of the r.v. that will be fine)

So we can define these "measures" (assume $x_i \geq x_{i+1}$) on LOI:

$H(X) = -\sum_{x \in X} \mu(X = x) \log(\mu(X = x))$

$G_n(X) = \sum_{x \in X} g_n(x),\ g_n(x) = \sum_{i \leq n, x_i \in x} \mu(x_i),$

$NG(X) = \sum_{x \in X} ng(x),\ ng(x) = \sum_{x_i \in x} i\mu(x_i),$

entropy, n-tries guessability, av. n. of guesses

We can define the induced orders on LOI ($F = H, G_n, NG$):

$$X \leq_F Y \iff \forall \mu.\ F(X) \leq F(Y)$$

(μ= distribution on atoms)

Teo: $\leq_H = \leq_{G_n} = (\leq_{NG})^{\mathsf{op}} = \leq_L$

Connecting LOI and leakage analysis of programs:

associate to a program P the partition $L(P) = ([|P|])^{-1}$ (whenever this make sense e.g. deterministic programs)

notice that for a k bit secret $L(-)$ is a surjection on a lattice over 2^k atoms

Connecting Δ_F and LOI:

$ME(P) = \log(G_1(L(P))) - \log(G_1(L(h)))$

$GE(P) = NG(L(h)) - NG(L(P))$

$H(P) = H(L(P)) - H(L(P)|h) (= H(L(P)))$

($L(h)$ is the eq.rel: $\forall h, h'.\ h \simeq h'$)

Teo: $\leq_H = \leq_{G_n} = (\leq_{NG})^{\text{op}} = \leq_L$

Corollary on leakage:

$$\leq_H = \leq_{ME} = \leq_{GE} = \leq_L$$

This answer "what do they have in common?"

They agree on the classification of source code threats

Because of those results if two programs are not ordered (as partitions) then we can always find distributions making any of the two less than the other

$A = \{\{a,b\}\{c,d\}\{e\}\}, B = \{\{a,d\}\{b,c,e\}\}$

$|A| = 3 > 2 = |B|$. $A < B$ using u.d.

For $A > B$ take μ s.t.

a=0.1, b= 0.1,c=0.39,d=0.4,e=0.01 then

$H(A) = H(0.2, 0.79, 0.01) = 0.7994 < 1 = H(B) = H(0.5, 0.5)$

$G(A) = 0.1 + 0.4 + 0.1 = 0.6 < 0.79 = 0.4 + 0.39 = G(B)$

Smith's examples: (h is a secret of size $8k$ bits):

1. $P_1 \equiv$ `if (h%8 == 0) then o = h; else o = 1;`

2. $P_2 \equiv$ `o = h& 0`$^{\texttt{7k-1}}$`1`$^{\texttt{k+1}}$;

Smith's observation: P_1, P_2 similar leakage ($H(P_1) = k + 0.169 < H(P_2) = k + 1$) but they have a very different guessing behaviour:

P_1 prob guess whole secret=1/8, P_2 leaks $k+1$ bits but nothing of the remaining ones

As partitions ($m = 2^{8k-3}, r = k + 1$):

$L(P_1) = \{\{h_1000\}, \ldots, \{h_m000\}, X_1\}$

$L(P_2) = \{Y_1, \ldots, Y_r\}$

this view makes simple to see what distribution make one more secure than the other:
$<$: u.d. on X_1 and ϵ elsewhere
$>$: u.d. on Y_i ending in 000, and ϵ elsewhere

Code leakage $\neq$ environment leakage

Further concepts:

many runs = l.u.b.s in LOI

loops=$L(P) = \sqcup_{n \geq 0} W_{\leq n} \sqcap C$

Channel capacity (maximum leakage) easy: $|L(P)|$ (number of blocks)

Smith: Channel capacity coincide for Shannon and guessability measures (det)

Applying these concepts to real code:

Bounding leakage of C code: "does it leak more than k bits"?

See a C program as a family of equivalence relations (one for each choice of low inputs)

verify whether exists an equivalence relation in this family with $\geq k$ classes (active attacker model e.g. underflow leak CVE-2007-2875)

Linux Kernel analysis

practicalities:
h = kernel memory. size: thousands of bits
low = C structures. size: arbitrary

so exists an eq.rel. among 2^{100} eq. rel. over a set of 2^{10000} atoms with ≥ 16 bits leakage?

not easy..

CBMC can help: symbolic+unwinding assertions

(Heusser-Malacaria 2010) use assume-guarantee reasoning and use CBMC for these questions on bounds

The approach is powerful, e.g. quantifying architecture leaks : CVE-2009-2847 doesn't leak on a 32 bits architecture but leaks on a 64 bits machine.

It is also the first verification of linux kernel vulnerability patches

Demo