Automatically quantifying information leaks in software

CREST January 2012

Pasquale Malacaria
Queen Mary University of London

The problem

An attacker has some a priori knowledge of the secret which is improved by observing the system

measure this improvement: how much did the attacker gain from the observations?

Example: an attacker steal your cash card; he has no idea about your pin (apriori probability to guess it 0.0001)

to randomly try a pin number at a cash machine will generate two possible observations:

1. the pin is accepted (with probability 0.0001),

2. the pin is rejected (with probability 0.9999)

Quantitative analysis of confidentiality according to a measure F:
difference of the measure F on the secret h before and after observing the system P

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

1. $F(h)$ = measure of the secret h before observations

2. $F(h|P)$ measure of the secret h given observations P

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

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

$F(h) = H(h)$=entropy of secret h before observations= a priory information about h

$F(h|P) = H(h|P)$=entropy of secret h given observations= information about h given observations

Δ_H (Cash machine,h)=0.00147 (bits of information)

(B) : *Probability of guessing in one try:* (introduced by Smith and noted ME)
$F(h) = -\log(\max_{x \in h} \mu(h = x))$ = a priory probability of guessing h
$F(h|P) = -\log(\sum_{y \in P} \mu(y)(\max_{x \in h} \mu(h = x|P = y)))$ = probability of guessing h given observations

Δ_{ME} (Cash machine,h)=1 (= log(2): chances have doubled)

(C) *Expected number of guesses:* (GE)

$F(h) = \sum_{x_i \in h, i \geq 1} i \ \mu(h = x_i) =$ a priory average number of guesses for h

$F(h|P) = \sum_{y \in P} \mu(y)(\sum_{x_i \in h, i \geq 1} i\mu(h = x_i|P = y)) =$ av. n. of guesses for h given observations

(assume $i < j$ implies $\mu(h = x_i) \geq \mu(h = x_j)$)

Δ_{GE} (Cash machine,h)$=$ 0.9999

From now on assume:

System=deterministic program (e.g. C code),

Observations=outputs, return values ... time

Two questions:

1. how these measures F classify threats?

2. what do they have in common?

How do they classify threats?

Define a ”more F secure” ordering between programs P, P' by

”the measure F on P is always less than the measure F on P' ”:

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

Does this ”source code secure” ordering depend on the choice of F?

remember F can be

1. entropy,

2. probability of guessing,

3. average number of guesses

In general there is no relation between entropy, probability of guessing or average number of guesses (Massey) but...

All measures give the same ordering:

Teo: $\leq_H = \leq_{ME} = \leq_{GE}$

This answer "what do they have in common?"

They agree on the classification of source code threats

So what is this common order to 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)

associate to a program P the partition $L(P)$ whose blocks are h undistinguishable by the observations:
formally $L(P) = ([|P|])^{-1}$

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

What do they have in common?...

the channel capacity coincide

i.e. the maximum measure according to entropy and probability of guessing coincide:

$$\max_h \Delta_{ME}(P,h) = \max_h \Delta_H(P,h) = \log_2(|L(P)|)$$

$|L(P)|$ (number of blocks)

Applying these concepts to real code:

"is the channel capacity of this C function $> k$"?

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 $\geq 2^k$ classes (active attacker model e.g. underflow leak CVE-2007-2875)

Linux Kernel analysis verification practicalities:
$h =$ kernel memory. size $\simeq$ 4 Gigabits
low $=$ C structures. size $\simeq$ arbitrary

e.g. for a small 5 integer structure and bound $k = 16$ the question is: exists a relation among 2^{160} equivalence relations over a space of 2^{64} atoms with more than 2^{16} equivalence classes?

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

Current directions

Bit pattern analysis. Meng and Smith 2011

Bit pattern analysis of Linux kernel. Sang and Malacaria 2012

Also work on side channels Kopf et Alt. (Timing + ongoing on Cache leaks)

"Black box" approaches(Chotia work, Sidebuster)

Conclusions:

Scientific: different measures of confidentiality are not so different

Engineering: impossible verification tasks are sometimes possible

Testing: David?

Description	CVE 20-	LOC	$k^\star$	Patch	bound	Time
AppleTalk	09-3002	237	64	Y	$>$6 bit	83s
IRDA	09-3002	167	64	Y	$>$6 bit	30s
tcf_fill_node	09-3612	146	64	Y	$>$6 bit	3m
sigaltstack	09-2847	199	128	Y	$>$7 bit	49m
cpuset†	07-2875	63	64	$\times$	$>$6 bit	1m
eql	10-3297	179	64	Y	$>$6 bit	16s
SRP getpass	–	93	8	Y	$\leq$1 bit	0.1s
login_unix	–	128	8	–	$\leq$2 bit	8s

table 1: Experimental Results. $\star$ Number of unwindings $\dagger$