

# Noisy Defenses: Subverting Malware's OODA Loop

[Extended Abstract]

Daniel Bilar  
Department of Computer Science  
Wellesley College  
Wellesley, MA 02481  
dbilar@wellesley.edu

## Categories and Subject Descriptors

K.6.5 [Management of Computing and Information Systems]: Security and Protection—*Invasive software, Information flow controls*

## General Terms

Process Query System, Interactive Computation, Malware, Games, Entropy

## 1. SYNOPSIS

The question is how to deal in practice with complex evolved malicious software (malware). In light of recent empirical and theoretical findings, we propose moving beyond techniques premised on Turing Machine models towards iterative games and black-box process modeling within an interactive computational framework.

## 2. INFORMATION-GAIN ADVERSARIAL MALWARE

Modern malware attempts to systematically thwart the hoped-for entropy reduction of the defense. The implementation characteristics of this design philosophy include generation of functionally equivalent phenotypes, sophisticated EPO (entry point obscuring), code integration, substitution, decoy and permutation techniques, within a time-dependent, multi-stage structure increasingly resistant to white-box analysis.

Given constant static analysis time, an analyst must deal with increasing uncertainty about the location, control flow handoff and activation triggers, and even existence of malicious functionality. Also, given constant dynamic analysis time, through dummy loops,  $k$ -ary design, and anti-emulation countermeasures, modern malware systematically lessens the hoped-for information gain of these detection techniques. We stress that the reduction needn't be complete, or even unmitigatable under relaxed resource con-

straints. Our concern are the practical, day-to-day, near-real-time detection and containment demands.

### 2.1 Practical concerns

Polymorphic malware contains decryption routines which decrypt encrypted constant parts of the malware body. The malware can mutate its decryptors and keys in subsequent generations. The decrypted body remains constant. Metamorphic malware generally does not use encryption, but is able to mutate the body in subsequent generations using techniques such as code transposition, equivalent instruction substitution and register reassignments. The net result of these techniques is a shrinking usable "constant base" for strict signature-based detection approaches.

For these types of modern malware, empirical antivirus (AV) detection effectiveness is faltering. In February 2008, for instance, fourteen state-of-the-art, updated AV scanners were checked against thousands of replicants of well-known, *previously submitted*, highly poly- and metamorphic malware samples. The miss rate was 100% to 0%, with an average detection miss rate of roughly 20%.

### 2.2 Theoretical concerns

So-called  $k$ -ary malware, of which at present only laboratory or very trivial examples are known to exist, is able to elude conventional deployed defenses. This is accomplished by partitioning the malware's functionality spatiotemporally into  $k$  distinct parts, with each part (human action can be a part, as well) containing a seemingly innocuous functionality subset. In serial or parallel combination, they subsequently become active.

Even more worrisome is that current AV seems unable to detect  $k$ -ary threats or remove them completely after detection. This is due to fundamental theoretical model limitations: Recent research indicates that neither multi-tape Turing machines, nor the classic Cohen model can thoroughly describe  $k$ -ary codes [2]. Thus, Turing machine models - equating computation as such with mathematical closed-box transformations of input to output - are insufficiently expressive to capture complex malware.

### 2.3 Interactive Computation

The notion of computability rests largely on the Church-Turing thesis. Although the Church-Turing thesis refers ex-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CSIR '08, May 12-14, Oak Ridge, Tennessee, USA  
Copyright 2008 ACM 978-1-60558-098-2/08/05 ...\$5.00.

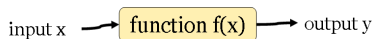


Figure 1: Computation as a function-based, closed transformation from input to output.

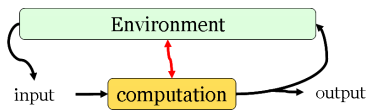


Figure 2: Computation as an open system

Explicitly to the computation of *functions*, this small but important caveat is not emphasized to the extent that it should be; instead it is understood to mean that Turing machines model *all* computation.

Herein lies the flaw: Not only are there some functions that cannot be computed by Turing machines, but more fundamentally, not all computable problems are function-based to begin with [3]. Function-based or algorithmic (with loops) computation requires the input to be specified at the start of the computation; in other words it is a closed transformation from input to output (Fig. 1). In contrast, interactive computational models serve as a theoretical bridge between Turing Machines (functions) and interaction (communication) with the environment. The computation is open, I/O happens during computation, not just before or after (Fig. 2). This model describes everyday computing more accurately than closed transformation, since a GUI, an OS, a control system does not ‘compute’ in the strict closed algorithmic sense of computing.

Within the domain of theoretical computational virology, TM modeling paucity concerns have very recently begun to be addressed [5]. What is missing are practical solutions.

### 3. CONTROLLED CONFUSION

Information-adversarial design strives to reduce the relative information gain of the defender’s strategies. Similarly, defenders may adopt a Bayesian, information-centric viewpoint viz the malware, controlling its assumptions via the entropy of prior distribution, as well as its information gain in the a posteriori distribution.

Some defenses will *actively* engage in an iterative 2-player (possibly n-player), imperfect, non-zero-sum game in order to control the relative information gain of the malware’s reconnaissance and influencing its decision algorithms. Defenses may also be expressed *passively* and target the prior distributions. This passive tack seeks to lessen exploit success by introducing heterogeneity in the digital biotopes.

Hence, informally, through decoys and irregularities, interactions and observations, defenders probabilistically implement strategies that actively and passively lead the malware astray. This active approach bears some similarities to the concept of subverting an enemy’s OODA (Observe, Orient, Decide, and Act) loop, an information warfare strategy pioneered by military fighter pilot Col. John Boyd which seeks to pro-actively influence and change enemy behavior. We shall sketch an active defense framework based on this

information-centric point of view<sup>1</sup>

## 4. ACTIVE DEFENSE FRAMEWORK

We seek to manipulate malware’s actions. This entails modeling its internal hypothesis structure, entering its OODA loop and controlling its decisions. To this end, we need an *observation framework* that can infer said internal hypothesis structure and a *control framework* that dynamically chooses strategies which control adversarial information gain for the benefit of the defender.

### 4.1 Observation: Process Query System

We propose that a PQS serve to dynamically ‘black-box model’ malware. The necessary observation events can be both passively recorded and actively enticed through interactive interactions.

PQSS [1] were initially designed to solve the Discrete Source Separation Problem by setting up a DBMS framework that allows for *process description* queries against internal models, a task for which traditional DBMS are unsuitable, since the queries (e.g. SQL) are typically formulated as Boolean expressions. These models can take the form of Finite State Machines, rule sets, Hidden Markov models, Hidden Petri Nets, among others.

Four sequential components are linked together in a PQS: Incoming observation → multiple hypothesis generation → hypothesis evaluation by models → model selection. The overarching goal is to detect processes by leveraging the correlation between events (such as observations) and the processes’s states.

Fig. 3 illustrates these components. Processes have hidden states which emit observables. The relationship between observables and states is not bijective, meaning a given observation may be emitted by more than one state. The so-called ‘tracks’ are associations of observations to processes. Hypotheses represent consistent tracks that explain the observables. The hypotheses in our domain correspond to the malware’s internal control structure, which is inferred from its behavior through observation.

### 4.2 Control: Games

The goal of the control framework is to create the illusion of win-win (non-zero-sum) viz the malware’s goals by iteratively either weakening useful/accurate and strengthening useless/misleading information gain through defensive strategies. Again, we seek to influence the malware’s actions by manipulating its view of the ‘outside world’ and thereby influencing its OODA loop. Game theory may provide a suitable interactive framework.

## 5. ILLUSTRATION

Our active defense framework is sketched in a toy example in Fig. 4. Suppose malware has been (perhaps probabilistically) identified through interaction and observation. Its

<sup>1</sup>Passive techniques that affect the prior belief distribution have been developed for some time now, even though their deployment is relatively recent. Examples include honeynets, address space layout randomization, and homogenizing the a priori belief distribution of defense mechanisms.

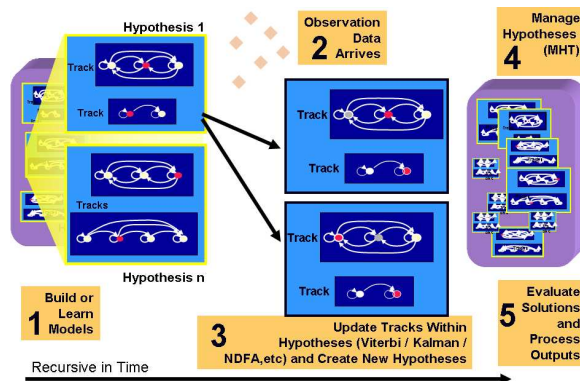


Figure 3: PQS operation sequence (Cybenko, UCLA IPAM Workshop, 2005)

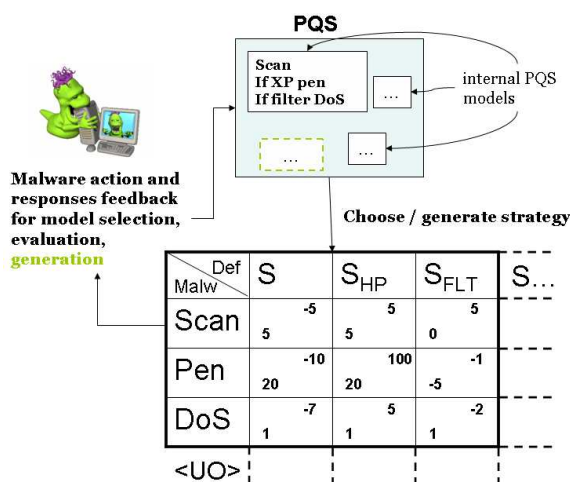


Figure 4: Observation/Modeling through PQS, Control/Response through interactive game strategies

toy internal hypothesis structure (and strategies) are modeled by `Scan;if XP penetrate;if filtered DoS` in a PQS internal model. The defense's strategies are  $S$  (no defense),  $S_{HP}$  (honeypot),  $S_{FLT}$  (filter/block ICMP response). The game matrix shows (hypothetical) payoffs of the defense's and malware's strategy combinations.

The malware starts scanning and wants to get to  $[Pen, S]$  penetrating a real host. The defense wants to engage sequential strategies such that the malware penetrates a fake host  $[Pen, S_{HP}]$ , thereby giving the illusion of a win for the malware while learning more about it. Again, the defense wants to iteratively control, not necessarily minimize malware's hoped-for entropy reduction. Strategies may not be fixed and dynamically generated as PQS models adapt to the malware responses, as denoted by  $S_{...}$  (new defense strategy) and  $<UO>$  (unknown observation).

## 6. OPEN QUESTIONS

Though PQS models have been constructed a priori that describe complex cyberattacks (differentiating between simultaneous sophisticated attacks on a target network [4]), dynamic inference and modeling the malware's hypothesis structure is an open area. The attacker and defense's payoffs depend on the relative entropy induced by the respec-

tive strategies; thus, a suitable entropy measure should be investigated [6]. Finally, the active defense framework presupposes the ability to classify stimuli as probabilistically originating from malware, i.e we may need fuzzier categories than under attack/not under attack.

## 7. REFERENCES

- [1] G. Cybenko and V. Berk. Process detection in homeland security and defense applications. *Proceedings of SPIE: Sensors and C3I Technologies for Homeland Security and Homeland Defense*, 6201, 2006.
- [2] E. Filiol. Formalisation and implementation aspects of  $k$ -ary (malicious) codes. *Journal in Computer Virology*, 3(2):75–86, 2007.
- [3] D. Goldin and P. Wegner. The Church-Turing Thesis: Breaking the Myth. *LNCS: New Computational Paradigms*, pages 152–168, 2005.
- [4] I. Gregorio-de Souza and V. Berk. Detection of complex cyber attacks. *Proceedings of SPIE*, 6201, 2006.
- [5] G. Jacob and E. Filiol. Malware as interaction machines: a new framework for behavior modelling. *Journal in Computer Virology*, 4(2), 2008.
- [6] R. K. Niven. Combinatorial Information Theory: I. Philosophical Basis of Cross-Entropy and Entropy. *ArXiv*, April 2007.