Using a Novel Behavioral Stimuli-Response Framework to Defend against Adversarial Cyberspace Participants

Daniel Bilar ∗
Brendan Saltaformaggio †

∗ Director of Research
Siege Technologies
Manchester, New Hampshire, USA

† Department of Computer Science
University of New Orleans
New Orleans, Louisiana, USA

3rd International Conference on Cyber Conflict
Cooperative Cyber Defence Centre of Excellence
Tallinn, Estonia
June 8, 2011

Siege Technologies (Manchester, NH, USA)

Who We Are

Company Founded in 2009. Privately held R&D company with offices in Manchester (NH), Reston (VA) and Rome (NY). Founders have 85 years of combined contractor or government experience

Focus Computer Security, Information Operations, Information Warfare, Computer Network Operations

People 10 scientists/engineers, half of which are PhDs (practitioners, not just eggheads)

Whom We Work With

DoD, Intelligence Community, and commercial entities

What We Do

Advanced System Testing / Red Teaming, Defense Engineering, Software Development and Analysis, Code Analysis / Reverse Engineering, Special Application Support, Hypervisors

Speaker

A bit about me

Domicile Born in the US, grew up in Germany, France, Switzerland. Came to the US for post-secondary studies (BA, M.Eng. PhD, post-doc)

Education Business, law, economics; philosophy, theology, history, political science, computer science; operations research, industrial engineering, engineering sciences

Work White goods salesman, software engineer, financial analyst, law and engineering consultant, university professor, research director

General Research Area: Security Studies

Background As PhD student, founding member of the Institute for Security and Technology Studies at Dartmouth (counter-terror, defense research for US DoJ and US DHS)

Security Studies Solutions cannot be mere math/technical - must span different dimensions such as psychology, technology, computer science, operations research, history, law, sociology and economics. See (good & bad) Aaron Barr

Previous Academic Funding AFRL, DoD/NSA, Navy SPAWAR, LA BoR / NASA

Talk Roadmap

Status Quo

Classic AV byte-pattern matching has reached a dead end with modern malware.
AV is in practice almost useless - dirty secret known to practitioners for a decade.

Why? Problem Setup Favors Adversary

They pose hard problems Through design dissimulation techniques, their functionality and intent difficult to ascertain
We are easy Targets situated on a predominantly WYSIWYG "gameboard"
$\rightarrow$ **Defenses forced to solve time-intensive (minutes, hours, days) halting-type problems** while adversarial cyberspace participants do not
Hence, have to **turn tables** to achieve acceptable (subsecond, seconds) response times

Autonomous Baiting, Control and Deception (ABCD)

Inversion of Problem Setup Morph adversary's view of gameboard, increase adversarial participant's footprint, noise levels, effectiveness, decision complexity
Bait, Control and Deceive Repeated dynamic stimuli-response game, framework decides probabilistically nature of participant and engages appropriate defensive measures
End vision AI-assisted, sub-second decision cycle, autonomic framework that probabilistically determines, impedes, quarantines, subverts, possibly attributes and possibly inoculates against suspected adversarial cyberspace participants

Detection Rates: Malware Increasingly Resistant

Bad: Empirical AV Results

Report Date	AV Signature Update	MW Corpus Date	False Negative (%)
2011/05	Feb. 22nd	Feb. 23rd -Mar. 3rd	[39-77]
2011/02	Feb. 22nd	Feb. 10th	[0.2-15.6]
2010/011	Aug. 16th	Aug. 17th -24th	[38-63]
2010/08	Aug. 16th	Aug. 6th	[0.2-19.1]
2010/05	Feb. 10th	Feb. 11th -18th	[37-89]
2010/02	Feb. 10th	Feb. 3rd	[0.4-19.2]
2009/011	Aug. 10th	Aug. 11th -17th	[26-68]
2009/08	Aug. 10th	Aug. 10th	[0.2-15.2]
2009/05	Feb. 9th	Feb. 9th -16th	[31-86]
2009/02	Feb. 9th	Feb. 1st	[0.2-15.1]
2008/11	Aug. 4th	Aug. 4th -11th	[29-81]
2008/08	Aug. 4th	Aug. 1st	[0.4-13.5]
2008/05	Feb. 4th	Feb. 5th -12th	[26-94]
2008/02	Feb. 4th	Feb. 2nd	[0.2-12.3]

Table: Empirical miss rates for 9-16 well-known AV products. After freezing update signatures *for one week*, best AV missed between 30-40 % of new malware, the worst missed 65-77 %

Worse: Theoretical Findings

Detection of interactive malware at least in complexity class $NP^{NP^{NP_{oracle}}_{oracle}}$ [EF05, JF08]
Blacklisting Deadend Infeasibility of modeling polymorphic shellcode [YSS07]

1[st] Fingerprint: Win32 API Calls

Synopsis: Look at Frequency of Calls

Observe and record Win32 API calls made by malicious code during execution, then compare them to calls made by other malicious code to find similarities

Goal

Classify malware quickly into a family
Set of variants make up a family

Main Result (2005) [Rie05]

Simple (tuned) Vector Space Model yields over 80% correct classification
Behaviorial angle seems promising

2^{nd} Fingerprint: Opcode Frequency

Synopsis: Look at Machine Instruction Makeup

Statically disassemble the binary, tabulate the opcode frequencies and construct a statistical fingerprint with a subset of said opcodes

Goal

Compare opcode fingerprint across non-malicious software and malware classes for quick identification purposes

Main Result (2006) [Bil07b]

For differentiation purposes, infrequent opcodes explain more data variation than common ones
Static makeup Not good enough as discriminator.
Exacerbating: **ROP** [RBSS09][CSR10], 'malicious computation' (Sept. 2010: Adobe 0-day CVE-2010-2883 used ROP attack to bypass DEP)

3^{rd} Fingerprint: Callgraph Properties

Synopsis: Look at Control Flow

Represent executables as callgraph, and construct graph-structural fingerprint for software classes.
Callgraph is relationship-graph of function calls

Goal

Compare 'graph structure' fingerprint of unknown binaries across non-malicious software and malware classes

Main Result (2007) [Bil07a]

Malware tends to have a lower basic block count, implying a simpler functionality: Limited goals, interaction → fewer branches
Behavioral Angle Leverage simpler decision structure to 'outplay' malware?

Callgraph: sub_402400 (Backdoor.Win32.Livup)

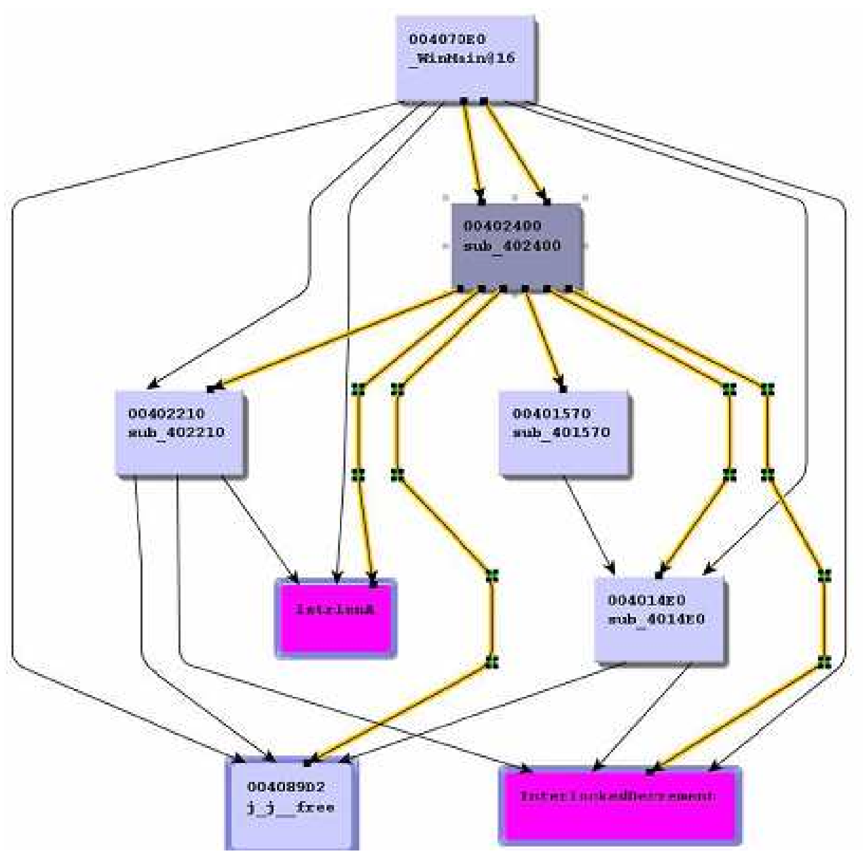

Figure: Callgraph of sub_402400: Indegree 2, outdegree 6

Metrics Collected

Total function count of executable
Indegree of functions (for sub_402400 two callers)
Outdegree of functions (for sub_402400 six callees)
Function 'type' as normal, import, library, thunk
In- and out-degree of a given function

Flowgraph: sub_402400 (Backdoor.Win32.Livup)

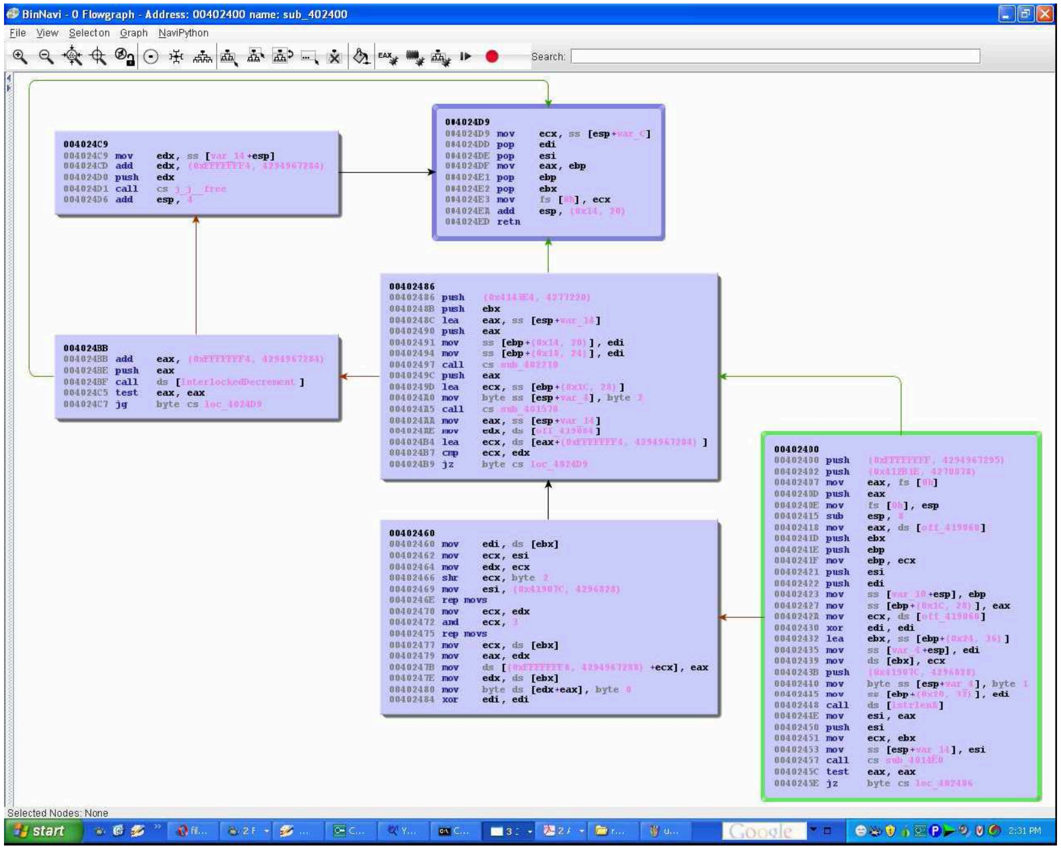

Figure: Backdoor.Win32.Livup.c: Flowgraph of sub_402400, consisting of six basic blocks. The loc_402486 basic block is located in the middle of the flowgraph given above. It consists of 16 instructions, of which two are calls to other functions

Metrics Collected

Basic block count of function

Instruction count of a given basic block

Example: loc_402486

```
402486 push (0x4143E4, 4277220)
40248B push ebx
40248C lea eax, ss [esp + var_14]
402490 push eax
402491 mov ss [ebp + (0x14, 20)], edi
402494 mov ss [ebp + (0x18, 24)], edi
402497 call cs sub_402210
40249C push eax
40249D lea ecx, ss [ebp + (0x1c, 28)]
4024A0 mov byte ss [esp + var_4], byte 2
4024A5 call cs sub_401570
4024AA mov eax, ss [esp + var_14]
4024AE mov edx, ds [off_419064]
4024B4 lea ecx, ds [eax + (0xF4, 429)]
4024B7 cmp ecx, edx
4024B9 jz byte cs loc_4024D9
```

Callgraph: Degree Distribution

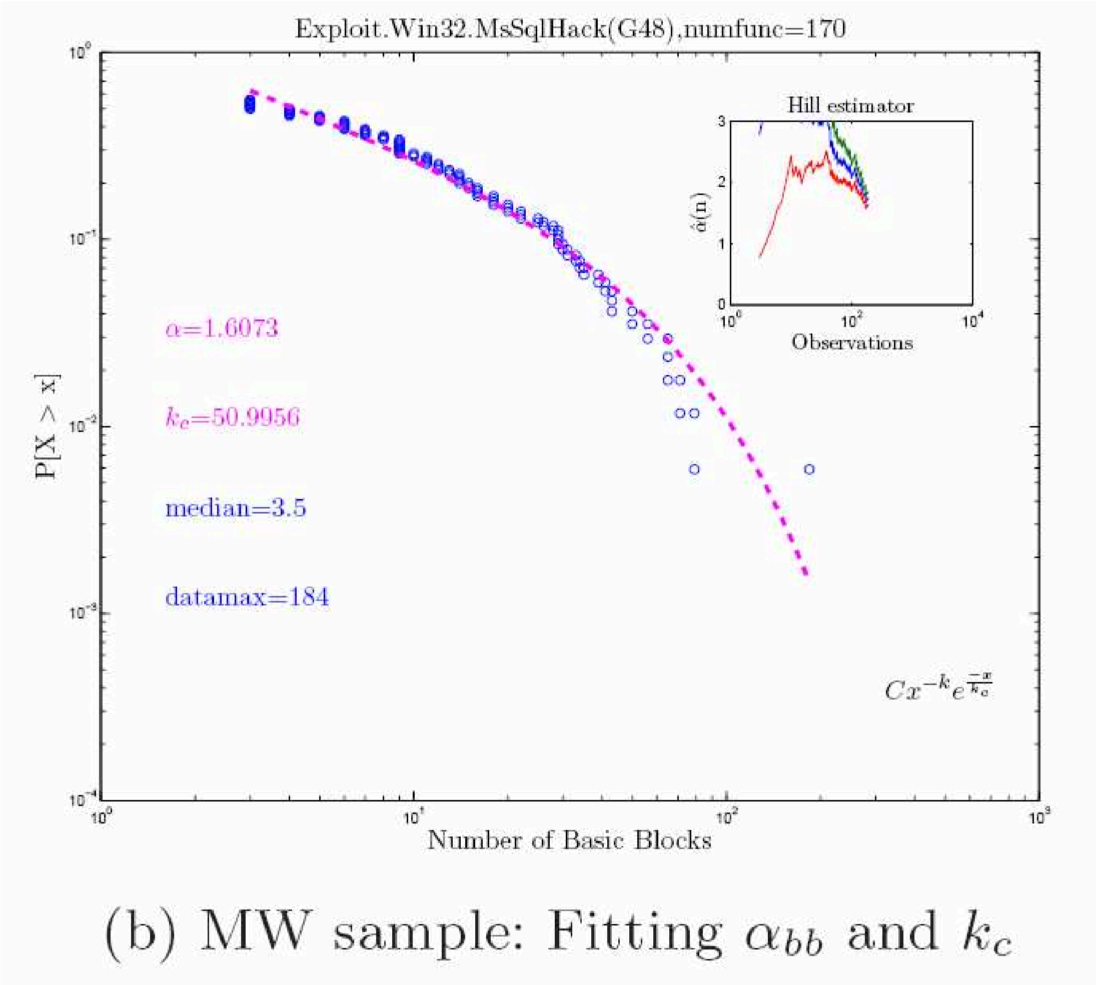

Figure: Pareto fitted ECCDF with Hill estimator $\hat{\alpha}(n)$

Power (Pareto) Law

Investigate whether indegree $d_{indeg}(f)$, outdegree $d_{outdeg}(f)$ and basic block count $d_{bb}(f)$ distributions of executable's functions follows a truncated power law of form

$$P_{d_*(f)}(m) \sim m^{\alpha_{d_*(f)}} e^{-\frac{m}{k_c}}$$

with α a power law exponent, k_c distribution cutoff point, $\hat{\alpha}(n)$ Hill estimator (inset) used for consistency check [CSN09]

Callgraph: Differentiation Results

class	Basic Block	Indegree	Outdegree
t	2.57	1.04	-0.47
Goodware	N(1.634,0.3)	N(2.02, 0.3)	N(1.69,0.307)
Malware	N(1.7,0.3)	N(2.08,0.45)	N (1.68,0.35)

Table: Only one statistically relevant difference found: Basic block distribution metric $\mu_{\text{malware}}(k_{bb}) \neq \mu_{\text{goodware}}(k_{bb})$ via Wilcoxon Rank Sum

Interpretation

Malware tends to have **a lower basic block count**, implying a simpler functionality: Less interaction, fewer branches, limited functionality

Idea

Kasparov wins because he can think 5-7 chess moves ahead. Can we **leverage malware's simpler decision structure to outplay it?**

Conceptual: Actively Morphing, Game-Playing Defense Framework

Idea: Subversion of Decision Loop

Interactive, morphing framework to manipulate, mislead and contain MW.
Infer MW internal decision points, then change the environment (i.e. passive environmental morphing and active environmental stimuli) → **manipulate observables** malware might use for its decisions.
Environment plays an iterative, seemingly cooperative, mixed strategy, multi-player game.
Goal Subvert MW's internal control structure and goad it into a position favorable to the defense.

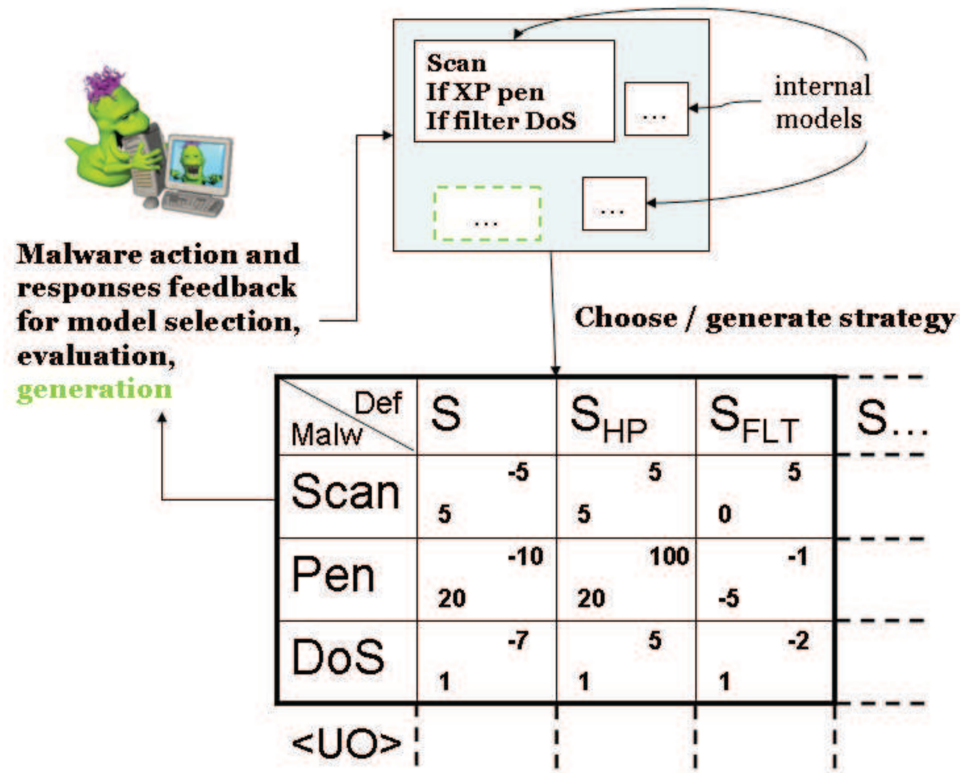

Figure: The environment and the malware can be seen as engaged in an *iterative, seemingly cooperative, possibly mixed strategy, possibly multi-player game.* Can I identify, quantify and deploy strategies (i.e. passive environmental morphing and active environmental stimuli) to goad malware into a payoff corner?

Related Inspiration, Data and Work

Inspiration

OODA (1995) Strategy concept for information warfare developed by USAF

VM Architecture Randomization (2004) Calculated 31 available architecture entropy bits for use against code injection attacks [HLS05]

Conficker A (2008) Exits upon detection of Ukrainian keyboard locale [PSY09].

Data

Environmental Awareness of Malware 2008 study (6200 samples) found disproportionate deterrence value of imitating VMs and debuggers through light-weight registry key insertions, system call hooking [CAM+08]

Work

Nepenthes (2006) Scalable hybridization of low- and high-interaction honeynets [BKH+06]

Wolfsting (2010) Run baseline trace, then provide malware with resources it wants (files, registry keys, processes) [Mul10]

Blocking Games (2011) Nash equilibria computable in poly-time through combinatorial tools (blocking pairs of matrices) [Gue11]

Morphing the Gameboard: The ABCD-ACP project

Characteristics

Continuous Evolution and Adaptation of interaction strategies through algorithms (machines) and intuition (human crowdsourcing)
Resilience against subversive participants seeking to undermine strategies
Continuous increase in decision cycle speed Aggressive optimization over all framework components, workflow and bottlenecks
Stability Guarantees DoD network sizes through rigorous mathematical analysis and simulation

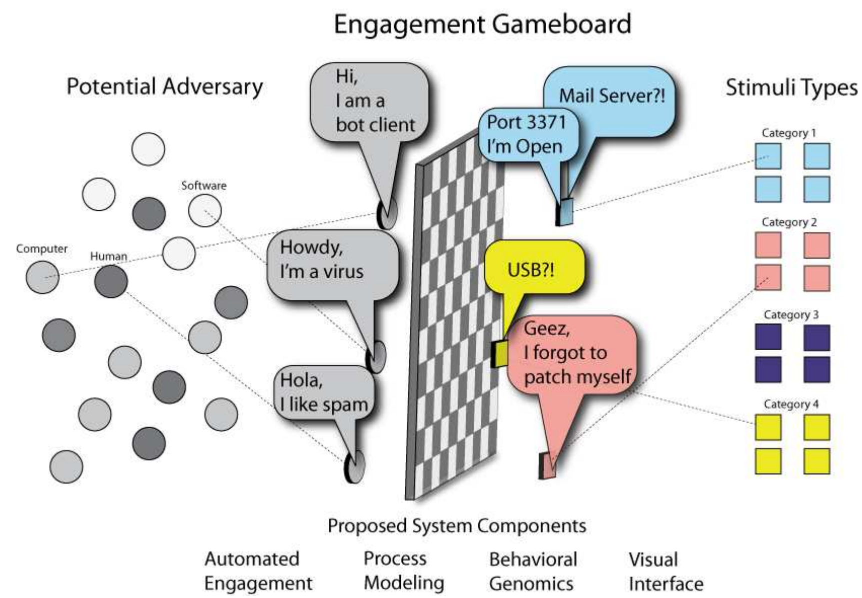

Figure: Notional Gameboard. Stimuli (e.g. fake network drives, fake processes with names of popular applications, AutoCad files etc.) are deployed and participants' responses evaluated

Morphing the Gameboard: Concepts

Overview

Gameboard consists of **virtualized operating environment** into which bait/stimuli are injected to induce potential ACP's (both humans and programs) to 'show their colors'
Morphing Influence ACP's perception of environment, and goad it into a position favorable to the defense
Baits/Stimuli Gameboard-morphing actions taken by Defender to induce behavioral responses from participants. Specificity (low false positives are desired: Does it flag benevolent participants as adversarial?) and sensitivity (low false negatives are desired: Does it miss adversarial participants?)
Probabilistic identification via stimuli/responses 'game'. Weigh different hypotheses (ex: loglikelihood Bayesian odds) consistent with aggregate evidence whether a participant's observed behavior can be classified as adversarial (Whewell's 19th century 'Consillience of Induction' [Sny08])

Working Hypotheses

1. From observations of triggered stimuli/responses, **uncertainty anent unknown intent** can be reduced. In particular, potential adversarial participants can be probabilistically identified.
2. Defender can control the dynamic behavior of ACPs by influencing what Participants perceive within the Gameboard

Morphing the Gameboard: Concepts

Game

Players Participants versus Defender play repeated, dynamic, imperfect information, non-cooperative stimuli-response game
Participants Potentially adversarial programs or humans on the Gameboard. All Participants (benign or malicious) are situated within the Gameboard
Defender Situated outside the Gameboard to hide footprint. Ability to introduce (real or perceived) baits/stimuli, change macroscopic Gameboard parameters, gauge responses and initiate defensive moves.

Defensive Actions

Defender Conversation consists of a high level scenario which is either preemptively engaged, chosen by the user, or activated by other defensive systems. Conversation examples include “Worm”, “Rootkit”, “Bot”, “Trojan”, “Trusted Insider”,“Hapless User”
Defender Scenario informs one or more engagement types. Engagement types include “Offer spread vectors”, “Offer confidentiality vector”, “Offer reconnaissance vector”,“Present weakened defense”, “Change system parameter”
Engagement Strategy dynamically chosen for each engagement type. Game tree aggregate of baits (stimuli) and participants. Depending on responses, next bait/stimuli chosen.

Morphing the Gameboard: Baits

Bait Portfolio

Bait	Bait actions	Malware Ex.	False Positive
Dummy processes	Inject false antivirus programs into the OS process list and monitor for halt in execution	Conficker (kills AV processes), Bugbear (shuts down various AV processes), Vundo (disables Norton AV)	low
Network Shares	Mounts and removes network shares on the client then monitors for activity	MyWife.d (attempt to delete System files on shared network drives), Lovgate (copies itself to all network drives on an infected computer), Conficker (infects all registered drives)	medium
Files	Monitors critical or bait (.doc, .xls, .cad) files	Mydoom.b (alters host file to block web traffic), MyWife.d (deletes AV system programs), Waledac.a (scans local drives for email adds)	low
User action	Executes normal user behavior on the client system and monitors for unusual execution	Mydoom.b (diverts network traffic thus altering what is expected to appear), Vundo (eat up system resources - slows program execution)	high
Thread Injection	Continually checks number of threads for any changes	Poisonivy, Pandex (injects code into 'explorer.exe' or 'msnmsgr.exe')	very low

Morphing the Gameboard: Defender

Defender Goals

Mission assurance/continuity Defender should not self-sabotage or sabotage benign Participants. Mission continuity constraints include but are not limited to: sustain mission availability, confidentiality, integrity, command & control and more.
Actionable Information Gain Defender's responses geared towards reducing uncertainty and learning more about potential ACP (e.g. by migrating ACP into a highly instrumented environment).
Defender Stealth Potentially adversarial participant should remain unaware of Defender's observation and manipulation of ACP's perception of Gameboard
Subversion Defender responds in such a way as to 'repurpose' ACP
Participant Attribution Defender responds in such a way that attribution of adversarial behavior source is made more likely (e.g. smart watermarking/ poisoning of data)
Inoculation Defender may be able to model ACP observed behavior (ex. PQS models [CB04]) to build a vaccine, supplementing efforts in the realm of byte code signatures

Defender Action

Abstract Categories Collberg's [*primitives*] (cover, duplicate, split/merge, reorder, map, indirect, mimic, advertise, detect/ response, dynamic) [CN09]
Quarantine [*Indirect*] Defender moves ACP to an instrumented but isolated platform in order to learn more about its behavior.
(Self-)terminate [*Tamperproof*] Defender terminates ACP or induces its self-termination. In addition, Defender may simulate termination of benign components as a strategic mimetic move (such as unlinking it from the process table).
Scarcity [*Mimicry, Tamperproof*] Defender presents 'critica' or 'strained' Gameboard state in an effort to violate ACP's expectations (e.g. 99% memory utilization, heavy network congestion, no heap space left).
Subversion [*Tamperproof*]:
Data-taint/poison potential ACP in order to create **an attribution trail** (e.g email bugs in .pst file). Especially important for military defense systems and kinetic retaliation, where attackers try to plausibly deny responsibility through one of more levels of indirection.

Theoretical and Implementation Challenges Ahead

"A problem worthy of attack proves its worth by fighting back"

Bait Specificity and Sensitivity Need empirical quantification with robust bait portfolio

Multiple ACPs Implicitly assume just one ACP operating at a time. Multiple ACPs give Discrete Source Separation Problem. Promising approach is Process Query Systems [CB06]

Computational Learning Need to analyze and control the rate of convergence. Informal goal is ACP identification with 2-4 bait/stimuli/response moves. Learning through interaction as validation mechanism (ex. PAC or Vapnik-Chervonenkis theory)

Stochastic Imperfect Information Game Payoff tied to knowledge, varies over time, retroactive. Is this analytically solvable?

Morphing Fundamentals System state, entropy measures

Performance Transitioning to production systems multi-objective optimization challenge (speed, stability, management). Scaling to 100,000s of virtualized hosts on infrastructure clouds poses non-linear problems [Kot11]

Morphing Ground Truth: System's Degrees of Freedom

System State and Entropy Measures

Defense goal is not to maximally confuse ACP, but to manipulate malware's decision tree by controlling its cross-entropy calculus D^x of perceived target/environment. Requires *appropriate state representation of Gameboard and entities*, since this directly determines cross-entropy measure D^x

Ex: If system's governing distribution (probability of given realization) $\mathbb{P} = P(n_i | q_i, N, s, I)$ s.t. prior probabilities q_i, number of entities N, number of states s with $\sum_{i=1}^{s} n_i = N$ and background information I is *multinomial* with $\mathbb{P} = N! \prod_{i=1}^{s} \frac{q_i^{n_i}}{n_i!}$, then cross-entropy to manipulate is Kullback-Leibler

$$D^x_{KL} = \sum_{i=1}^{s} \left(p_i N^{-1} \ln N! + p_i \ln q_i - N^{-1} \ln((p_i N)!) \right)$$

However, if system is not governed by multinomial $\mathbb{P}$ (e.g. Bose-Einstein system's $\mathbb{P}_{BE}$ is multivariate negative hypergeometric), D^x_{BE} is not KL

Cross-entropy D^x_{KL} and Shannon entropy not universal, do not apply to every system [Niv07]

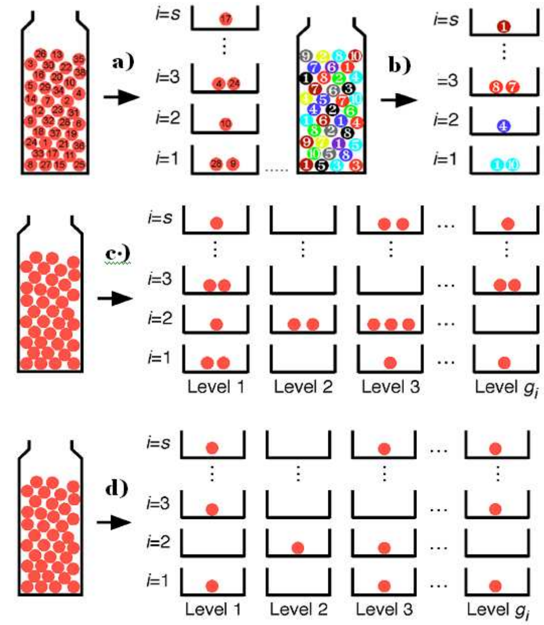

Figure: Model of Maxwell-Boltzmann (a-b), (c) Bose-Einstein and (d) Fermi-Dirac systems

a) N distinguishable balls to s disting. boxes, with n_i of each state $\longrightarrow \mathbb{P}_{MB}$ is multinomial
b) Urn has M disting. balls, with m_i of each state, sample N balls with replacement with n_i in each state $\longrightarrow \mathbb{P}_{MB}$ is multinomial

c) Balls indistinguishable, $\binom{g_i + n_i - 1}{n_i}$ permutations of n_i indisting. balls in g_i disting. boxes $\longrightarrow \mathbb{P}_{BE}$ is multivariate negative hypergeometric
d) Balls indistinguishable, max. 1 in each level, $\binom{g_i}{n_i}$ permutations of n_i indisting. balls in g_i disting. boxes with $n_i \in \{0, 1\} \longrightarrow \mathbb{P}_{FD}$ is multivariate hypergeometric

Future Future

End Vision of ABCD-ACP

'Skynet' AI-assisted, microsecond decision cycle, autonomic stimuli response framework that probabilistically determines, impedes, quarantines, subverts, possibly attributes and possibly inoculates against suspected adversarial cyberspace participants

Human Symbiosis Co-evolution into an autonomous defense 'alter ego' for human decision makers

Coupled with stress (emotion) sensors poised to take over when judgment is deemed to be too affected by emotions andor information overload

$\longrightarrow$ Spirit of USAF Science & Technology 2010-2030 [Dah10])

Complements Efforts In Other Military Domains

DARPA's Integrated Battle Command (BAA 05-14) Give decision aids for battle ops

DARPA's Real-Time Adversarial Intelligence & Decision Making (BAA 04-16) Help battlefield commander with threat predictions in tactical operation

Israel's Virtual Battle Management AI Robotic AI defense system take over from flesh-and-blood operators. In event of doomsday strike, system handles attacks that exceed physiological limits of human command

Why Emphasis on Autonomous Decision?

Human Operator is Subsystem Possible to degrade and subvert end system through subsystem attacks. See CCD COE 2009 "On n^{th} Order Attacks" [Bil09]

Subsystem Subversion: n^{th} Order Attacks

Objective

Induce Instabilities in mission-sustaining ancillary systems that ultimately degrade, disable or subvert end system
n: Degree of relation 0th order targets the end system, 1st order targets an ancillary system of the end system, 2nd order an ancillary system of the ancillary system etc.

Systems

Definition A whole that functions by virtue of interaction between constitutive components. Defined by relationships. Components may be other systems. Key points: Open, isomorphic laws
Nature Technical, algorithmic, societal, psychological, ideological, economic, biological and ecological
Examples Resource allocation / throughput / stability control, manufacturing, visualization environments, social welfare systems, voting systems, data / goods / energy generation/ transmission/ distribution, reputation management, entropy externalization, business models and economic systems

Systems, Attacks and Assumption Violation

Assumptions

Fundamentally, attacks work because they **violate assumptions**

Finite (i.e real life engineered or evolved) systems **incorporate implicit/explicit assumptions** into structure, functionality, language

System **geared towards 'expected', 'typical'** cases

Assumptions reflect those 'designed-for' cases

Intuitive Examples of Attacks and Assumption Violations

Man-in-Middle Attacks Identity assumption violated

Race Condition Attacks Ordering assumption violated

BGP Routing Attacks Trust assumption violated

Generative Mechanism and Assumptions

Optimization process incorporating tradeoffs between objective functions and resource constraints under uncertainty

Some **assumptions generated by optimization** process

Optimization Process: Highly Optimized Tolerance

HOT Background

Generative first-principles approach proposed to account for power laws $P(m) \sim m^{\alpha} e^{-\frac{m}{k_c}}$ in natural/engineered systems [CSN07, CD00]

Optimization model incorporates tradeoffs between objective functions and resource constraints in probabilistic environments

Used Forest, internet traffic, power and immune systems

Pertinent Trait

Robust towards common perturbations, **but fragile** towards rare events

Inducing 'rare events' in ancillary systems is goal of n^{th} order attack

Probability, Loss, Resource Optimization Problem [MCD05]

$$\min J \tag{1}$$

subject to

$$\sum r_i \leq R \tag{2}$$

where

$$J = \sum p_i l_i \tag{3}$$

$$l_i = f(r_i) \tag{4}$$

$$1 \leq i \leq M \tag{5}$$

M events (Eq. 5) occurring iid with probability p_i incurring loss l_i (Eq. 3)

Sum-product is objective function to be minimized (Eq. 1)

Resources r_i are hedged against losses l_i, with normalizing $f(r_i) = -\log r_i$ (Eq. 4), subject to resource bounds R (Eq. 2).

Subsystem Attacks: Examples

Target Ancillary System to Subvert End Systems [Bil10]

P2P Networks RoQ attacks can be mounted against distributed hash tables used for efficient routing in structured P2P networks through join/leave collusions and bogus peer newcomer notifications

Power Grid Load balancing in electricity grids relies on accurate state estimation. Data integrity attacks on a chosen subset of sensors make these estimates unreliable, which could push such feedback systems into unstable state

Democracy Voting systems assume honest participants vote their actual preference. In elections with more than two candidates, system can be undermined by strategic voting, targeting the ranking process subsystem

Trusted Code Second-order control-flow subversion attack termed return-oriented programming (ROP) induce innocuous code to perform malicious computations

Financial Exchange Advent of high-frequency trading infrastructures (physically collocated, hence low latency) gave rise to trading approaches (first- and second-order degradation and subversion attacks) targeting the Efficient Market Hypothesis and its subsystems

Signals from Above

AF Chief Scientist Werner Dahms on USAF Science & Technology 2010-2030 [Dah10]

Augmentation of Human Performance Use of highly adaptable autonomous systems to provide significant time-domain operational advantages over adversaries limited to human planning and decision speeds

Massive virtualization Agile hypervisors, inherent polymorphism complicate adversary's ability to plan and coordinate attacks by reducing time over which networks remain static, and intruder to leave behind greater forensic evidence for attribution.

Resilience Make systems more difficult to exploit once entry is gained; cyber resilience to maintain mission assurance across entire spectrum of cyber threat levels, including large-scale overt attacks

Symbiotic Cyber-Physical-Human Augmentation through increased use of autonomous systems and close coupling of humans and automated systems

Direct augmentation of humans via drugs or implants to improve memory, alertness, cognition, or visual/aural acuity, screening (brainwave patterns or genetic correlators)

2011 IEEE Symposium on Computational Intelligence in Cyber Security (April 2011)

Mission Assurance Track Explore theoretical and applied research work in the academic, industrial, and military research communities related to mission assurance.

Selected Topics Mission representation, modeling, simulation, visualization, impact estimation and situational awareness; Decision making and decision support; Engineering for mission assurance and resilience strategies.

How Scientists Relax

Little Humor

Infrared spectroscopy on a vexing problem of our times: *Truly* comparing apples and oranges.

Thank You

Thank you for your time and the consideration. I appreciate being back at the CCD COE in beautiful Tallinn ⌣̈

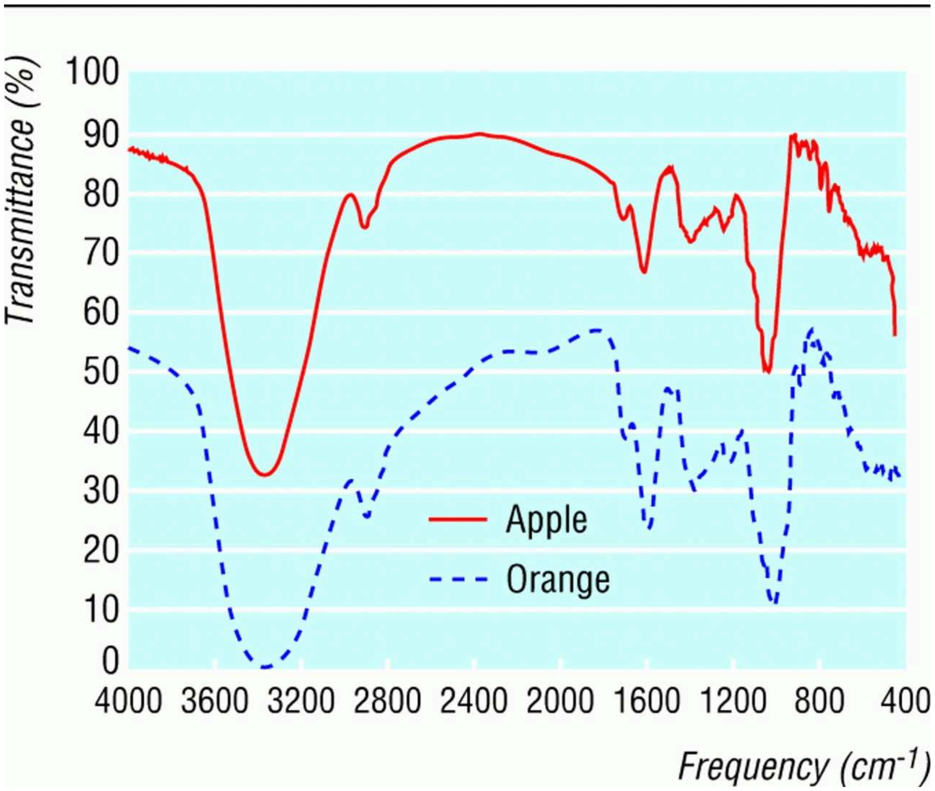

Figure: A spectrographic analysis of ground, desiccated samples of a Granny Smith apple and a Sunkist navel orange. Picture from [San95]

References I

Daniel Bilar, *On callgraphs and generative mechanisms*, Journal in Computer Virology **3** (2007), no. 4.

______, *Opcodes as predictor for malware*, International Journal of E-Security and Digital Forensics **1** (2007), no. 2.

______, *On nth order attacks*, The virtual battlefield : Perspectives on cyber warfare (Christian Czosseck and Kenneth Geers, eds.), IOS Press, 2009, pp. 262–281.

______, *Degradation and subversion through subsystem attacks*, IEEE Security & Privacy **8** (2010), no. 4, 70–73.

P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling, *The nepenthes platform: An efficient approach to collect malware*, Lecture notes in computer science (2006), 165–184.

X. Chen, J. Andersen, Z.M. Mao, M. Bailey, and J. Nazario, *Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware*, Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008. IEEE International Conference on, IEEE, 2008, pp. 177–186.

George Cybenko and Vincent Berk, *An overview of process query systems*, Proc. SPIE, vol. 5403, 2004.

______, *Process detection in homeland security and defense applications*, Proc. SPIE **6201** (2006).

Jean Carlson and John Doyle, *Highly Optimized Tolerance: Robustness and Design in Complex Systems*, Physical Review Letters **84** (2000), no. 11, 2529+.

References II

- Christian Collberg and Jasvir Nagra, *Surreptitious software: Obfuscation, watermarking, and tamperproofing for software protection*, Addison-Wesley Professional, 2009.
- Aaron Clauset, Cosma R. Shalizi, and Mark Newman, *Power-Law Distributions in Empirical Data*, SIAM Reviews (2007).
- Aaron Clauset, Cosma R. Shalizi, and M. E. J. Newman, *Power-law distributions in empirical data*, SIAM Review **51** (2009), no. 4, 661+.
- Stephen Checkoway, Hovav Shacham, and Eric Rescorla, *Are text-only data formats safe? or, use this LaTeX class file to pwn your computer*, Proceedings of LEET 2010 (Michael Bailey, ed.), USENIX, April 2010, To appear.
- Werner Dahms, *Technology Horizons: A Vision for Air Force Science & Technology During 2010-2030*, Tech. report, USAF Science and Technology, May 2010, http://www.aviationweek.com/media/pdf/UnmannedHorizons/Technologys
- Éric Filiol, *Computer viruses: from theory to applications*, Springer, 2005.
- Assane Gueye, *A game theoretical approach to communication security*, Ph.D. thesis, UCLA, Spring 2011.
- D.A. Holland, A.T. Lim, and M.I. Seltzer, *An architecture a day keeps the hacker away*, ACM SIGARCH Computer Architecture News **33** (2005), no. 1, 34–41.
- Gregoire Jacob and Eric Filiol, *Malware As Interaction Machines*, J. Comp. Vir. **4** (2008), no. 2.

References III

- Evangelos Kotsovinos, *Virtualization: Blessing or curse?*, CACM **54** (2011), no. 1, 61–65.
- Lisa Manning, Jean Carlson, and John Doyle, *Highly Optimized Tolerance and Power Laws in Dense and Sparse Resource Regimes*, Physical Review E **72** (2005), no. 1, 16108+.
- Vikram Mulukutla, *Wolfsting: Extending online dynamic malware analysis systems by engaging malware*, Master's thesis, North Carolina State University, 2010.
- Robert K. Niven, *Combinatorial Information Theory: I. Philosophical Basis of Cross-Entropy and Entropy*, ArXiv (2007).
- P. Porras, H. Saidi, and V. Yegneswaran, *Conficker C Analysis*, Tech. report, SRI International Technical Report, 2009.
- Ryan Roemer, Erik Buchanan, Hovav Shacham, and Stefan Savage, *Return-oriented programming: Systems, languages, and applications*, 2009, In review.
- Chris Ries, *Automated identification of malicious code variants*, J. Comput. Small Coll. **20** (2005), no. 5, 140–141.
- Scott Sandford, *Apples and oranges: a comparison*, Annals of Improbable Research **1** (1995), no. 3.
- L.J. Snyder, *"the whole box of tools": William whewell and the logic of induction*, British Logic in the Nineteenth Century (2008), 163.
- Michael E. Locasto Yingbo Song and Salvatore J. Stolfo, *On the infeasibility of modelling polymorphic shellcode*, ACM CCS, 2007, pp. 541–551.