Oscillations through Co-evolution: A Manifestation of Moving Target Defense Conficker Case Study

Daniel Bilar (Siege Technologies)  
George Cybenko (Dartmouth College)  
John Murphy (ProQueSys)

CSIIRW 8, ONRL (Oak Ridge, TN)  
January 9, 2013
Support

• Research partially supported by DARPA I2O, DHS, AFRL, DOD, OSD, and AFOSR with UTEP, Ball Aerospace, Pikewerks, Siege
  – All opinions and results expressed are those of authors and not necessarily those of the funding agencies

• Thanks also to V. Berk, I. Gregoriou-de Souza, J.T House, D. Sicilia, G. Stocco, P. Sweeney
Outline of talk 1/2

• **Background**: Studied public data in various domains
  – US border security, computer vulnerability databases, offensive & defensive coevolution of worms (Conficker)
  – Modeled as players in adversarial situation

• **Findings**: Performance metrics **oscillate over time**
  – No asymptotic convergence, not monotonic

• **Claim**: In majority of (adversarial) games, players do *not* compute Nash Equilibriums over (static) strategy sets but use myopically perceived best responses at each time step
  – ‘Classical’ game theory is not the best fit

• **Why**: Not a *stationary* environment! Ongoing sequences of moves, countermoves, deception and strategic adaptation
  – Explains exhibited oscillations and consistent with data
Outline of talk 2/2

• **Problem:** Oscillations modeled by replicator equations
  – Typically 3\textsuperscript{rd} degree, non-linear, analytically difficult
  – Inverse problem of estimating RE parameters from observations of behavior computationally tractable

• **Claim:** Possible to infer players motives, costs and move options by observation of oscillation
  – Not discussed in this talk

• **Contributions of authors**
  – Detailed empirical analysis of players Conficker & environment (Bilar & Murphy)
  – Abstraction of game through Quantitative Attack Graph (Bilar & Cybenko & Murphy)
  – “Asymptotic” cut set theorem (Cybenko) for optimal defense allocation
You know you are working in an adversarial domain when you want to see this kind of progress...

...but instead, you see this ...

...or this ...

Total Number of Vulnerabilities and Exploits Disclosed (all platforms)
Border security...

[Graph showing human apprehensions from 1992 to 2010]
War on drugs...

Drug Apprehensions (Entire SWB) x 1,000,000

Time


0 0.5 1 1.5 2 2.5 3

0 0.5 1 1.5 2 2.5 3
Comments

• “Performance” measures may oscillate (not monotonic)
  – Depends partly on normalization of metrics (see Fig 3.1 in BMC (2012))

• Operating against human adversaries is different than operating against nature

• Games not defined a priori, game details not known
  – Players do not know who the other players are, what their possible moves might be and, perhaps most importantly, what their preferred outcomes or objectives are

• Result: Co-evolution, adaptation as evinced through oscillations
Conficker

• AKA Downup, Downadup, Kido
• Detected November 2008
• Largest worm/botnet infection since 2003
• Infected million’s of machines
• Evolved through 5 versions in several months
• Affected military systems in France, UK etc
• Used many vulnerabilities and techniques
Conficker Versions A-E Host States

Adversarial Dynamics: The Conficker Case Study. Daniel Bilar, George Cybenko and John Murphy
Conficker Timeline

Security ‘hole’ arises long range vulnerability and weaponizable exploit available Sept. 2008

BCA time regime

Race to market

Ecosystem configuration fluctuations give rise to security ‘holes’ -> Race to market

ACA time regime

Ecosystem timeline

After appearance of Conficker A, events on both timelines may be viewed as moves/response to each other -> Adversarial game

PCE time regime

Move with measures in 3 categories
### Examples of Conficker Analysis

**Table 3** Measures implemented by Conficker and Ecosystem between November 2008 and April/May 2009. **Bolded** indicates newly introduced measures. **Strike through** indicates dropped measures.

<table>
<thead>
<tr>
<th>Time Period</th>
<th>Player</th>
<th>Spread/Infect</th>
<th>Update</th>
<th>Armor</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nov 20, 2008 - Dec 28, 2008</td>
<td>11/20/08 Conficker.A</td>
<td>MS08-067, EnvCheck, FetchGeoIP</td>
<td>central rnd250-5 RC4 RSA-1024</td>
<td>obfusc</td>
<td>AVXP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ecosystem response</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AVsigA, DenyGeoIP, MSPatch</td>
<td>blockBiz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec 29, 2008 - Feb 19, 2009</td>
<td>12/29/08 Conficker.B</td>
<td>FetchGeoIP, InclGeoIP, EnvCheck, LocalShare USB, MS08-067</td>
<td>central RC4 RSA-1024 rnd250-8 MSbkcdr MD6v1 RSA-4096</td>
<td>obfusc</td>
<td>DNSblock, AutoUpdDis, AnlsShut</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ecosystem response</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AVsigB, DenyGeoIP</td>
<td>SRI-AB MD6v1</td>
<td></td>
<td>MSBounty</td>
</tr>
</tbody>
</table>
Abstraction of Attack/Defend Game

• Attackers attacks “weakest” paths to achieve goals
  – Weakest according to attackers’ understanding
  – Paths consist of one or more technical steps
  – Can create *completely new paths* and/or steps

• Defenders make some step(s) of the most common/damaging paths harder to traverse
  – Most common/damaging according to defenders’ understanding
  – Users/boss want to create new services so new paths emerge

• Iterate the above over time
Attack Graph for a Critical System

An attacker must traverse a path from the start state to the goal state to succeed.

Each step is a technical means to achieve a subgoal.

Note: This is an actual attack graph on a real but proprietary system.
Each step is a technical means to achieve a subgoal.

An attacker must traverse a path from the start state to the goal state to succeed.

Attacker uses his “shortest” path.
Attack Graph for a Critical System

Each step is a technical means to achieve a subgoal.

An attacker must traverse a path from the start state to the goal state to succeed.

Attacker uses his “shortest” path.

Defender protects a step by increasing its cost.
An attacker must traverse a path from the start state to the goal state to succeed.

Each step is a technical means to achieve a subgoal.

Attacker changes some edges in attack path.
An attacker must traverse a path from the start state to the goal state to succeed.

Each step is a technical means to achieve a subgoal.

Or the attacker picks a completely new path.
An attacker must traverse a path from the start state to the goal state to succeed. Each step is a technical means to achieve a subgoal. Or the attacker creates a new path.
Comments

• Attacks graphs are old technique but hard to build and quantify
  – State space explosions, how to assign edge costs, blind spots, etc
  – Maybe like democracy, worst way except for all others
• Prediction markets: QuERIES provides a technique for quantifying the attack graphs by cost, difficulty, etc
• We will adapt, invest and perform better if we quantify
  – Pursuit-evasion – go to where the prey will be
  – Flu shots anticipate the flu, not respond to current ones
  – Wayne Gretzky – “A good hockey player plays where the puck is. A great hockey player plays where the puck is going to be.”
Attack-Defend Game

- Estimate costs to attacker of traversing attack graph edges – shortest path is the most attractive for an attacker to take

Simple Example – Shortest path in yellow

Real Problem – What is/are the shortest path(s)?

- Optimization problem – maximize the cost of the shortest path from Start to Goal states
- Can formulate this as a linear programming problem – solution is the investment allocation that makes the least cost attack as expensive as possible
Linear Programming Formulation

\[
M = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 \\
\end{bmatrix}
\]

One column per edge
One row per path

\[
u = \begin{bmatrix}
A \\
B \\
C \\
D \\
E \\
\end{bmatrix}
\]

Vector of initial edge costs

\[
x = \begin{bmatrix}
a \\
b \\
c \\
d \\
e \\
\end{bmatrix}
\]

Vector of allocated costs

\[
\text{max } z \text{ such that } \\
M(u + x) \geq z \geq 0 \\
1^* x = K > 0, \ x \geq 0
\]
Example strategies

- Which edges are “best” to invest in? Suppose budget = 1.

- Analysis has shown that optimal investments are ultimately in a “cut set”
Back to Real System

Multiple edges mean multiple attack steps possible

Matrix M has 37 columns and 180 rows

5/5/14
Linear Programming Results Identify High Value Protection Paths for Different Investment Levels

- Result shows benefit from hardening multiple paths according to iterative algorithm
- X-axis shows total budget, Y-axis shows investment in hardening specific paths
- As budget increases, the defensive strategy is diversified, but investment into minimal cut edges continues
- Once the inputs to state 2 are hardened, investment begins in edges 20 and 37
Minimal cost paths for attacker

- Graph shows total cost of minimum-cost path resulting from investment strategy
- Minimum effort required by attacker
- Includes initial edge costs along path
- Slope decreases as investment strategy diversifies into hardening multiple paths
- “Diminishing rate of return”, ROI
Role of minimal cut sets

Each edge has cost 1

You have a budget of 1
Role of minimal cut sets

Invest that 1 unit here

Each edge has cost 1

You have a budget of 1

But this is the minimal cut set
Role of minimal cut sets

Now invest in the minimal cut set
“Asymptotic” Attack Graph Theorem (Cybenko)

If we are given an attack graph with
• a minimal cut set that has e edges
• a large investment budget, K
then
• the optimal budget allocation assigns $\approx K/e$ to each edge in the cut set and;
• the minimal cost path grows like $c + K/e$
where $c$ is a constant
Theorem states that optimal investment is eventually $K/e$ in minimal cut set edges. Initially, optimal investments can occur in other edges.

Linear Programming Results Identify High Value Protection Paths for Different Investment Levels.

Edges 1,2

Edges 20,37
Back to Real System

Multiple edges mean multiple attack steps possible

Matrix M has 37 columns and 180 rows

37 edges
180 paths
12 nodes
e = 6, cut set
Adversarial Dynamics Takeaways 1/2

- “Big data” needed
  - Red and blue forces’ data sets are needed
  - New, non-stationary statistics and estimation are key
  - Adaptation, not static equilibria, describe “solutions”
- “Hidden data” needed
  - Need to capture what players/agents think, not just the outcomes
- Anticipating moves is the way to gain advantage
  - Kasparov who can think 5-6 moves ahead
References

Thank you

Thank you for the kind consideration of these ideas

We are happy to answer questions / field comments 😊

Contact:
• Daniel Bilar: dbilar@acm.org
• George Cybenko: gvc@dartmouth.edu
• John Murphy: jmurphy@proquesys.com
Additional Slides
Oscillations as Manifestation of Adversarial Dynamics

• Evolution is a response to competition
• Competition exists among adversaries
• How do you know you are operating in an “adversarial” domain?
  – Oscillations of performance metrics
• Dynamics can be modeled by replicator equations
  – Typically 3rd order, non-linear (analytically difficult)
• *Inverse problem* of observing behavior and estimating parameters of replicator equation that guide behavior is tractable
• Possible to observe game play and strategy evolution and then make inferences about player’s motives, costs and move options