Using Strategy Objectives for Network Security Analysis

Elie Bursztein
Stanford University / LSV, Ens-Cachan
Inscrypt 2009
Introduction

Work purpose

Analyzing and anticipating computer networks attacks.
Network complexity: The Pentagon Case

Huge network

- 15 000 LAN Networks
- 7 000 000 Computers

Huge Security problems

- Flash Drive banned due to a virus spread (Nov 2008).
- 1500 computers taken (Jun 2007)
Attack Complexity

- email propagation of malicious code
- "stealth"/advanced scanning techniques
- widespread attacks using NNTP to distribute attack
- widespread attacks on DNS infrastructure
- executable code attacks (against browsers)
- automated widespread attacks
- GUI intruder tools
- hijacking sessions
- Internet social engineering attacks
- automated probes/scans
- techniques to analyze code for vulnerabilities without source code
- DDOS attacks
- increase in worms
- sophisticated command & control
- anti-forensic techniques
- home users targeted
- distributed attack tools
- increase in wide-scale Trojan horse distribution
- Windows-based remote controllable Trojans (Back Orifice)

1990 - 2004

Intruder Knowledge

Elie Bursztein
Some Epic Failures

- 2004 Bouygues Telecom: 2 servers downs → 3 200 000 cellphones down
- 2005 Japan Mitsubishi: 1 computer infected → 40 MB of confidential reports leaked on a P2P network
- 2007 Apple: 1 computer in the production line infected → 150 000 ipods infected by the trojan RavMonE.exe
Outline

Network Security
Attacks

Game

Automated Analysis

Conclusion
A **vulnerability** is a software *bug* that can be *exploited* by attacker to gain privilege.

An **exploit** is the piece of software that *takes advantage* of a software *bug*.

A **0day exploit** is an exploit for an undisclosed vulnerability.
Vulnerabilities as Step stones

- Large networks may suffer multiple vulnerabilities
- Patches and counter-measures need to be prioritized
- A minor vulnerability can turn into a major hole when used as a step-stone
Illustration of a Complex attack

- Exploit a bug in Firefox
- Install a trojan
- Stealth the web server password
- Upload a rogue page
- Stealth all user password
The Need for Automation

Attack analysis can’t be done by hand: network and attack are just too complex and big for that.

We need models and tools for this!
Attack Graph Frameworks

- 1998: Use of model-checking for host security [RS98]
- 2000: Use of model-checking for network [RA00]
- 2004: First complete framework that constructs the attack scenario [SW04]
- 2005: Mulval [Ou05] a framework based on Datalog.
- 2006: NetSpa [ALI06] a framework that scale up to 50 000 nodes.
Time is the Essence

Network security is a race between Intruder and Administrator.

Windows of vulnerability

![Diagram showing the relationship between exploit released, patch released, and windows of vulnerability over time.]

Elie Bursztein
The Need for Time

Without time *meaningless* actions are allowed in the model.

- Administrator can patch 1000 services instantly.
- Intruder can compromise 1000 services before the administrator have a chance to react.

Without time concurrent actions can’t be modeled. Ex: *Administrator may patch a service while Intruder tries to exploit it.*
Time and Game

Model
Timed automaton game [AFHMS].

Property
Property can be written in Timed Alternating-Time Temporal Logic [AHK06].
Collateral Effects

- Email
- DNS
- Web
- Internet
- DDOS Attack

Relationships:
- Email → DNS: Dommage collatéral
- DNS → Web: Dommage collatéral
- Internet → DNS: DDOS Attack
Outline

Network Security

Game
Structure
Rules

Strategy

Automated Analysis

Conclusion
Dual layer structure

The **Upper-layer** is the timed automaton game, the **Lower-layer** represents the network state.
Dual layer structure

The **Upper-layer** is the timed automaton game, the **Lower-layer** represents the network state.
Lower-layer: the network state

The lower layer is composed of

- The dependency graph
- A set of states (atomic proposition)
Web Service Receipt

To build a web service you need:

- A HTTP frontend to serve the data
Web Service Receipt

To build a web service you need:

- A HTTP frontend to serve the data
- A SQL backend to store the data
Web Service Receipt

To build a web service you need:

- A HTTP frontend to serve the data
- A SQL backend to store the data
- A way to administrate the service
To build a web service you need:

- A HTTP frontend to serve the data
- A SQL backend to store the data
- A way to administrate the service
The Dependency graph

- SQL
- HTTP
- HTTP2
- SSH

Using Strategy Objectives for Network Security Analysis
<table>
<thead>
<tr>
<th></th>
<th>SSH</th>
<th>SQL</th>
<th>HTTP1</th>
<th>HTTP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vulnerable</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Compromised</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
</tbody>
</table>
Rule Syntax

Rule syntax:

\[ \Gamma : \textbf{Pre} \varphi_{pre} \rightarrow \Delta, p, a, c \]

\textbf{Effect} \varphi_{eff}

- \( \varphi_{pre} \): Preconditions.
Rule Syntax

Rule syntax:

\[ \Gamma : \text{Pre} \varphi_{pre} \rightarrow \Delta, p, a, c \]

Effect \( \varphi_{eff} \)

- \( \varphi_{pre} \): Preconditions.
- \( \Delta \): Time required to complete the action.
- \( p \): The player that executes the rule.
- \( a \): Rule name.
- \( c \): Rule cost.
- \( \varphi_{eff} \): Effects.
Rule Syntax:

\[ \Gamma : \text{Pre } \varphi_{pre} \rightarrow \Delta, p, a, c \]

\textbf{Effect } \varphi_{eff}

- \( \varphi_{pre} \): Preconditions.
- \( \Delta \): Time required to complete the action. 
- \( p \): The player that executes the rule.
Rule Syntax

Rule syntax:

\[ \Gamma : \text{Pre} \varphi_{pre} \rightarrow \Delta, p, a, c \]

Effect \( \varphi_{eff} \)

- \( \varphi_{pre} \): Preconditions.
- \( \Delta \): Time required to complete the action.
- \( p \): The player that executes the rule.
- \( a \): Rule name.
Rule Syntax

Rule syntax:

\[ \Gamma : \text{Pre } \varphi_{pre} \rightarrow \Delta, p, a, c \]

**Effect** \( \varphi_{eff} \)

- \( \varphi_{pre} \): Preconditions.
- \( \Delta \): Time required to complete the action.
- \( p \): The player that executes the rule.
- \( a \): Rule name.
- \( c \): Rule cost.
Rule Syntax:

\[ \Gamma : \text{Pre } \varphi_{\text{pre}} \rightarrow \Delta, p, a, c \]

**Effect** \( \varphi_{\text{eff}} \)

- \( \varphi_{\text{pre}} \): Preconditions.
- \( \Delta \): Time required to complete the action.
- \( p \): The player that executes the rule.
- \( a \): Rule name.
- \( c \): Rule cost.
- \( \varphi_{\text{eff}} \): Effects.
Rule Syntax:

\[
\Gamma : \text{Pre } \varphi_{pre} \quad \rightarrow \quad \Delta, p, a, c \\
\text{Effect } \varphi_{eff}
\]

- \(\varphi_{pre}\): Preconditions.
- \(\Delta\): Time required to complete the action.
- \(p\): The player that executes the rule.
- \(a\): Rule name.
- \(c\): Rule cost.
- \(\varphi_{eff}\): Effects.
Rules Syntax

\[ F ::= A \quad \text{atomic propositions, in } \mathcal{A} \]
\[ \top \quad \text{true} \]
\[ \neg F \quad \text{negation} \]
\[ F \land F \quad \text{conjunction} \]
\[ \lozenge F \]
◊ **Semantics**

◊ **Vulnerable**: One of the successors is vulnerable.
Rule Example

\[ \Gamma: \text{Pre Vulnerable} \rightarrow 4, A, \text{Patch, 500} \]
\[ \text{Effect } \neg\text{Vulnerable } \land \neg\text{Compromise} \]
The Element of Surprise

if the opponent alters the service state *during the player rule execution* then the player is taken by *surprise*!
Decidability

Model-checking TATL over anticipation games is EXPTIME-Complete [BGL, ASIA’07].
Outline

Network Security

Game

Strategy
  What is a strategy?
  Using strategy
  Play Example

Automated Analysis

Conclusion
From counter-example to strategy

- An attack is a counter-example.
From counter-example to strategy

- An attack is a counter-example.
- Typically you end-up with many counter-examples.
From counter-example to strategy

- An attack is a counter-example.
- Typically you end-up with many counter-examples.

The problem
Which counter-example should the administrator look at first?
From counter-example to strategy

- An attack is a counter-example.
- Typically you end-up with many counter-examples.

The problem
Which counter-example should the administrator look at first?
- Which attack is the most devastating?
- What service to patch first?
Costs and Rewards

To find the most meaningful counter-example we need some additional informations.
Costs and Rewards

To find the *most meaningful* counter-example we need some additional informations.

- **Cost**: Each action has a cost.
Costs and Rewards

To find the most meaningful counter-example we need some additional informations.

- Cost: Each action has a cost.
- Reward: Each network asset has a value.
Costs and Rewards

To find the most meaningful counter-example we need some additional informations.

- Cost: Each action has a cost.
- Reward: Each network asset has a value.

\[
\mathcal{O} ::= O \quad \text{Objective } \in \phi \\
\mid O \land O \\
\mid \text{MAX}(O) \quad \text{maximize the value} \\
\mid \text{MIN}(O) \quad \text{minimize the value} \\
\mid O < x \quad x \in \mathbb{N} \\
\mid O > x \quad x \in \mathbb{N}
\]
Relation between Cost and Time

Assumption

The faster an action is, the more costly it is.

Real world examples of this assumption:

- Exploit: 0day versus Public exploit.
- Response team: 24/24h versus 8h /day
Definition

Strategy objectives are a tuple:

\[ S = (Na, Pl, Ob, Or, Co) \]

- **Na**: Strategy name
- **Pl**: The player
- **Ob**: Numerical objectives
- **Or**: Strict preference order
- **Co**: Constraints.

Example

\[ S = (Patch, A, Min(Cost) \land Max(OCost), OCost > Cost, \neg Compromised) \]
Computing Assets value

- Using the same value for each asset.
- Assigning value by hand.
- Computing automatically the value with a ranking algorithm [EB, INSCRYPT’08].
Which Objectives to choose?

- Minimizing cost (patch)
- Maximizing reward (attack)
Which Objectives to choose?

- Minimizing cost (patch)
- Maximizing reward (attack)

Wrong answer!
Player performs the best when his opponent makes mistakes. 
Game theory classical optimal criterion such as Nash equilibrium and Pareto optimality are not applicable.
The notion of dominant strategy was informally introduced in biology [H67] in 1967.

(Strictly) Dominant Strategy

The (strictly) dominant strategy is the player strategy that beats the maximum number of (every) opponent strategies.
The Lower Layer

<table>
<thead>
<tr>
<th></th>
<th>SSH</th>
<th>SQL</th>
<th>HTTP1</th>
<th>HTTP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vulnerable</td>
<td>⊤</td>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
<tr>
<td>Compromised</td>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
<td>⊥</td>
</tr>
</tbody>
</table>
The Lower Layer

<table>
<thead>
<tr>
<th></th>
<th>SSH</th>
<th>SQL</th>
<th>HTTP1</th>
<th>HTTP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1</td>
<td>100</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Elie Bursztein
Using Strategy Objectives for Network Security

What is a strategy?
Using strategy
Play Example
Intruder Rules

\[ \Gamma : \text{Pre } Vulnerable \land \neg \text{Compromise} \rightarrow 2, I, \text{Exploit 0day, 20000} \]

Effect Compromise
Intruder Rules

Γ : \textbf{Pre} Vulnerable \land \neg \textbf{Compromise} 
→ 2, I, Exploit 0day, 20000
\textbf{Effect} Compromise

Γ : \textbf{Pre} Vulnerable \land \neg \textbf{Compromise} 
→ 10, I, Exploit Public, 500
\textbf{Effect} Compromise
Intruder Rules

$$\Gamma : \text{Pre} \ Vulnerable \land \neg\text{Compromise}$$
$$\rightarrow 2, I, \text{Exploit 0day, 20000}$$
Effect Compromise

$$\Gamma : \text{Pre} \ Vulnerable \land \neg\text{Compromise}$$
$$\rightarrow 10, I, \text{Exploit Public, 500}$$
Effect Compromise

$$\Gamma : \text{Pre} \neg\text{Compromise} \land \text{Compromised}$$
$$\rightarrow 1, I, \text{Propagation, 5000}$$
Effect Compromise
Intruder Rules

\[ \Gamma : \text{Pre } \text{Vulnerable} \land \neg\text{Compromise} \rightarrow 2, I, \text{Exploit 0day}, 20000 \]

Effect Compromise

\[ \Gamma : \text{Pre } \text{Vulnerable} \land \neg\text{Compromise} \rightarrow 10, I, \text{Exploit Public}, 500 \]

Effect Compromise

\[ \Gamma : \text{Pre } \neg\text{Compromise} \land \Diamond\text{Compromised} \rightarrow 1, I, \text{Propagation}, 5000 \]

Effect Compromise
Administrator Rules

\[ \Gamma : \text{Pre } \text{Vulnerable} \rightarrow 4, A, \text{Patch, 500} \]

**Effect** $\neg \text{Vulnerable} \land \neg \text{Compromise}$
$S = (\text{Attack}, I, \text{MAX}(\text{Reward}) \wedge \text{Max}(\text{OCost}), \text{OCost} > \text{Reward}, \Diamond \text{Compromised})$
What is a strategy?

Using strategy

Play Example

SQL

HTTP

HTTP2

SSH
What is a strategy?

Using strategy

Play Example

<table>
<thead>
<tr>
<th>T</th>
<th>P</th>
<th>Action</th>
<th>Rule</th>
<th>Target</th>
<th>Succ</th>
<th>Payoff</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A</td>
<td>choose</td>
<td>Patch</td>
<td>SSH</td>
<td>⊥</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>I</td>
<td>choose</td>
<td>Exp 0 Day</td>
<td>SSH</td>
<td>⊥</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
What is a strategy?

Using strategy

Play Example

<table>
<thead>
<tr>
<th>T</th>
<th>P</th>
<th>Action</th>
<th>Rule</th>
<th>Target</th>
<th>Succ</th>
<th>Payoff</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>I</td>
<td>execute</td>
<td>Exp 0 Day</td>
<td>SSH</td>
<td>↓</td>
<td>1</td>
<td>20000</td>
</tr>
</tbody>
</table>
### What is a strategy?

- Using strategy
- Play Example

<table>
<thead>
<tr>
<th>T</th>
<th>Action</th>
<th>Rule</th>
<th>Target</th>
<th>Succ</th>
<th>Payoff</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>In Progress</td>
<td>Patch SSH</td>
<td>↓</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>I choose propagation</td>
<td>SQL SSH</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
What is a strategy?
Using strategy
Play Example

<table>
<thead>
<tr>
<th>T</th>
<th>P</th>
<th>Action</th>
<th>Rule</th>
<th>Target</th>
<th>Succ</th>
<th>Payoff</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>I</td>
<td>execute propagation</td>
<td>SQL</td>
<td>SSH</td>
<td>101</td>
<td>25000</td>
<td></td>
</tr>
</tbody>
</table>
What is a strategy?

Using strategy

Play Example

<table>
<thead>
<tr>
<th>T</th>
<th>P</th>
<th>Action</th>
<th>Rule</th>
<th>Target</th>
<th>Succ</th>
<th>Payoff</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>In Progress</td>
<td>Patch</td>
<td>SSH</td>
<td>⊥</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>I choose propagation</td>
<td>HTTP1</td>
<td>SQL</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
What is a strategy?
Using strategy
Play Example

<table>
<thead>
<tr>
<th>T</th>
<th>P</th>
<th>Action</th>
<th>Rule</th>
<th>Target</th>
<th>Succ</th>
<th>Payoff</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>I</td>
<td><strong>execute</strong></td>
<td>propagation</td>
<td>HTTP1</td>
<td>SQL 111</td>
<td>30000</td>
<td></td>
</tr>
</tbody>
</table>
What is a strategy?

Using strategy

Play Example

<table>
<thead>
<tr>
<th>T</th>
<th>Action</th>
<th>Rule</th>
<th>Target</th>
<th>Succ</th>
<th>Payoff</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>In Progress</td>
<td>Patch</td>
<td>SSH</td>
<td>⊥</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>I choose propagation</td>
<td>HTTP2</td>
<td>SQL</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
### What is a strategy?

#### Using strategy

#### Play Example

<table>
<thead>
<tr>
<th>T</th>
<th>P</th>
<th>Action</th>
<th>Rule</th>
<th>Target</th>
<th>Succ</th>
<th>Payoff</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>A</td>
<td>execute</td>
<td>Patch</td>
<td>SSH</td>
<td>1</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>InProgress</td>
<td>propagation</td>
<td>HTTP2</td>
<td>SQL</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
What is a strategy?

Using strategy

Play Example

<table>
<thead>
<tr>
<th>T</th>
<th>P</th>
<th>Action</th>
<th>Rule</th>
<th>Target</th>
<th>Succ</th>
<th>Payoff</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>I</td>
<td>execute</td>
<td>propagation</td>
<td>HTTP2</td>
<td>SQL</td>
<td>121</td>
<td>35000</td>
</tr>
</tbody>
</table>
We extended the anticipation game framework [EB,FAST’08] in order to model

- Multiples network cooperation
- Cost over the time (penalty)
- Timeline of events
Outline

Network Security

Game

Automated Analysis

Conclusion
We create an implementation in C (≈ 6500 lines) of the anticipation game framework called NetQi [EB,ATVA’08].
What Is Netqi?

NetQi name come from the English word Net and the Chinese word Qi, which mean vital energy flow. Hence NetQi is a tool designed to analyze the "network vital energy flow" to prevent attacks and failures that can harm this flow (legitimate traffic). It is based on timed game and model-checking theory.

So far it has been successfully used to analyze many network security threats including: Distributed Denial Of Service (DDOS), Network Exploit, Trust Relation Abuse, Information Leak, Password Cracking, Hard drive crash and DNS cache poisoning.

NetQi is not limited to network security and can be used to analyze most situation where interaction with a complex environment can be described as rules such as protein interaction in biology.

Learn how NetQi can help you

Download

The NetQi tools are Open-Source and free for non-commercial applications in academia, research, and for private persons. For commercial applications a commercial license is required.
## Case study

<table>
<thead>
<tr>
<th>Nb Nodes</th>
<th>Nb Dep</th>
<th>Strategy</th>
<th>type</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5200</td>
<td>27000</td>
<td>Defense</td>
<td>Exact</td>
<td>2.4 sec</td>
</tr>
<tr>
<td>5200</td>
<td>27000</td>
<td>Intrusion</td>
<td>Approximate</td>
<td>55 sec</td>
</tr>
</tbody>
</table>
Outline

Network Security

Game

Automated Analysis

Conclusion
In this work we have

- Developed the notion of strategy
- Show how strategy allow to select the most interesting play
- Implemented the model in order to show the effectiveness of the approach.
Perspective

- Finding network key services.
- Using dynamic costs and rewards.
- Modeling various classes of attackers.