Detecting Deception in the Context of Web 2.0.

Annarita Giani,
EECS, University of California, Berkeley, CA

Paul Thompson,
CS Dept. Dartmouth College, Hanover, NH
Outline

1. Motivation and Terminology
2. Process Query System (PQS) Approach
3. Detection of a complex attack
4. Conclusion and Acknowledgments
Cognitive Hacking

The user's attention is focused on the channel. The attacker exploits this fact and uses malicious information in the channel to mislead her.

1. Attacker: Makes a fake web site
2. Misleading information from a web site
3. Victim: Acts on the information from the web site
4. Attacker: Obtains advantages from user actions
Jonathan Lebed.

He spread fake rumors about stocks. Investors driven to buy shares of that stock inflating its price

The SEC wanted to prosecuted him for stock fraud. Was allowed to keep $500,000 from his “illegal” stock proceeds.

"Subj: THE MOST UNDERVERUED STOCK EVER
"Date: 2/03/00 3:43pm Pacific Standard Time
"From: LebedTG1

"FTEC is starting to break out! Next week, this thing will EXPLODE. . . .
"Currently FTEC is trading for just $2 1/2! I am expecting to see FTEC at $20 VERY SOON.
"Let me explain why. . . .
"The FTEC offices are extremely busy. . . . I am hearing that a number of HUGE deals are being worked on. Once we get some news from FTEC and the word gets out about the company . . . it will take-off to MUCH HIGHER LEVELS!
"I see little risk when purchasing FTEC at these DIRT-CHEAP PRICES. FTEC is making TREMENDOUS PROFITS and is trading UNDER BOOK VALUE!!!
Covert Channels

The user's attention is unaware of the channel. The attacker uses a medium not perceived as a communication channel to transfer information.

Attacker: Codes data into inter-packet delays, taking care to avoid drawing the attention of the user.

User: does not see inter-packet delay as a communication channel and does not notice any communication.
Phishing

The user's attention is attracted by the exploit. The information is used to lure the victim into using a new channel and then to create a false perception of reality with the goal of exploiting the user’s behavior.

1. Send a fake email

2. Misleading email to get user attention


4. Bogus web site

First name, Last name
Account # SSN
Cognitive Channels

A cognitive channel is a communication channel between the user and the technology being used. It conveys what the user sees, reads, hears, types, etc.

Focus of the current protection and detection approaches

The cognitive channel is the weakest link in the whole framework. Little investigation has been done on detecting attacks on this channel.
Cognitive Attacks

Our definition is from an engineering point of view.

Cognitive attacks are computer attacks over a cognitive channel. They exploit the attention of the user to manipulate her perception of reality and/or gain advantages.

COGNITIVE HACKING. The user's attention is focused on the channel. The attacker exploits this fact and uses malicious information to mislead her.

COVERT CHANNELS. The user is unaware of the channel. The attacker uses a medium not perceived as a communication channel to transfer information.

PHISHING. The user's attention is attracted by the exploit. The information is used to lure the victim into using a new channel and then to create a false perception of reality with the goal of exploiting the user's behavior.
The Need to Correlate Events

- Large amount of sensors for network monitoring
  - Intrusion Detection Systems
  - Network traces
  - File Integrity Checkers

- Large amount of Alerts
  - Overloaded operators
  - Hard to make sense of alarms

- Need a principled way of combining alerts
  - Reduce false alarms
  - Discover multistage attacks
Outline

1. Motivation and Terminology
2. Process Query System (PQS) Approach
3. Detection of a complex attack
4. Conclusion and Acknowledgments
Process Query System

Observable events coming from sensors

Models

Model $M_1$

Model $M_2$

... Model $M_k$

Hypothesis

Likelihood $L_1$

Likelihood $L_2$

... Likelihood $L_k$

PQS ENGINE

Tracking Algorithms
Framework for Process Detection

An Environment consists of

- Multiple Processes
  - \( \lambda_1 = \) router failure
  - \( \lambda_2 = \) worm

Events that produce

Indicators and Warnings

- 129.170.46.3 is at high risk
- 129.170.46.33 is a stepping stone
- 129.170.46.3 is at high risk

Hypotheses

- Hypothesis 1
- Hypothesis 2

FORWARD PROBLEM

1. Real World
2. Process Detection (PQS)
3. Unlabelled Sensor Reports
4. that detect complex attacks and anticipate the next steps
5. that PQS resolves into
6. that are used for control

INVERSE PROBLEM
Hierarchical PQS Architecture

Scanning
- Snort
- IP Tables

Infection
- Snort
- Tripwire

Data Access
- Samba

Exfiltration
- Flow and Covert Channel Sensor

RESULTS
Hidden Discrete Event System Models

Dynamical systems with discrete state spaces that are:

- **Causal** - next state depends only on the past
- **Hidden** – states are not directly observed
- **Observable** - observations conditioned on hidden state are independent of previous states

Example. Hidden Markov Model

- **N States**
- **M Observation symbols**
- State transition Probability Matrix, **A**
- Observation Symbols Distribution, **B**
- Initial State Distribution **π**

HDESM models are general
HDESM Process Detection Problem

Identifying and tracking several (casual discrete state) stochastic processes (HDESM’s) that are only partially observable.

TWO MAIN CLASSES OF PROBLEMS

**Hidden State Estimation**: Determine the “best” hidden states sequence of a particular process that accounts for a given sequence of observations.

**Discrete Sources Separation**: Determine the “most likely” process-to-observation association.
Discrete Source Separation Problem

Catalog of Processes

HDES M Example (HMM):
- 3 states + transition probabilities
- n observable events: a, b, c, d, e, …
- \( \text{Pr( state | observable event )} \) given/known

Observed event sequence:
…abcbbbaaaababbabcccbdddbebdbabcbae….

Which combination of which process models “best” accounts for the observations? Events not associated with a known process are “ANOMALIES”.
An analogy....

What does hbeolnjouolor mean?

Events are: h b e o l n j o u o l o r
Models = French + English words (+ grammars!)

hbeolnjouolor = hello + bonjour

Intermediate hypotheses include tracks: ho + be
PQS in Computer Security

Internet

DMZ

BRIDGE

WWW

Mail

WS

WinXP

LINUX

DIB:s

BGP

IPTables

Snort

Tripwire

Samhain

Observations

PQS ENGINE

Models

Worm

Exfiltration

Phishing
Outline

1. Motivation and Terminology
2. Process Query System (PQS) Approach
3. Detection of a complex attack
4. Conclusion and Acknowledgments
Complex Phishing Attack Steps

1. .... visits a web page.
   - inserts username and password.
   - (the same used to access his machine)

2. records username and password

3. accesses user machine using username and password

4. uploads some code

5. attacks the victim

6. downloads some data

Web page, Madame X

Attacker

Victim

Stepping stone

100.20.3.127

165.17.8.126

51.251.22.183

100.10.20.9
Complex Phishing Attack Observables

1. RECON
   SNORT: KICKASS_PORN
   DRAGON: PORN HARD CORE

2. ATTEMPT
   SNORT:政策违例
   NON-STANDARD-PROTOCOL

3. DATA UPLOAD
   FLOW SENSOR
   Sept 29 11:23:56

4. ATTEMPT (ATTACK RESPONSE)
   SNORT POTENTIAL BAD TRAFFIC
   Sept 29 11:24:06

5. DATA DOWNLOAD
   FLOW SENSOR
   Sept 29 11:24:07
Flow Sensor

- Based on the *libpcap* interface for packet capturing.
- Packets with the same source IP, destination IP, source port, destination port, protocol are aggregated into the same flow.

  - Timestamp of the last packet
  - # packets from Source to Destination
  - # packets from Destination to Source
  - # bytes from Source to Destination
  - # bytes from Destination to Source
  - Array containing delays in microseconds between packets in the flow

We did not use *Netflow* only because it does not have all the fields that we need.
## Two Models Based on the Flow Sensor

### Low and Slow UPLOAD

<table>
<thead>
<tr>
<th>Volume</th>
<th>Packets</th>
<th>Duration</th>
<th>Balance</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiny: 1-128b</td>
<td>4: 10-99</td>
<td>4: 1000-10000 s</td>
<td>Out</td>
<td>&gt;80</td>
</tr>
<tr>
<td>Small: 128b-1Kb</td>
<td>5: 100-999</td>
<td>5: 10000-100000 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6: &gt; 1000</td>
<td>6: &gt; 100000 s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### UPLOAD

<table>
<thead>
<tr>
<th>Volume</th>
<th>Packets</th>
<th>Duration</th>
<th>Balance</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiny: 1-128b</td>
<td>1: one packet</td>
<td>0: &lt; 1 s</td>
<td>Out</td>
<td>&gt;80</td>
</tr>
<tr>
<td>Small: 128b-1Kb</td>
<td>2: two pckts</td>
<td>1: 1-10 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medium: 1Kb-100Kb</td>
<td>3: 3-9</td>
<td>2: 10-100 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Large: &gt; 100Kb</td>
<td>4: 10-99</td>
<td>3: 100-1000 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5: 100-999</td>
<td>4: 1000-10000 s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6: &gt; 1000</td>
<td>5: 10000-100000 s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

W2SP2007 – Oakland, CA – May 24, 2007
Phishing Attack Model 1 – very specific
Phishing Attack Model 2 – less specific

1. RECON or ATTEMPT or COMPROMISE
2. UPLOAD dst, src
3. ATTEMPT dst, A
4. ATTEMPT dst, !src
5. UPLOAD dst, src
6. ATTEMPT dst, !src
7. DOWNLOAD src

W2SP2007 – Oakland, CA – May 24, 2007
Phishing Attack Model 3 – more general
Phishing Attack Model 3 – Most general

Strict models reduce false positives, but less strict models can detect unknown attack sequences.
Outline

1. Motivation and Terminology
2. Process Query System (PQS) Approach
3. Detection of a complex attack
4. Conclusion and Acknowledgments
Contribution

• Identification of a new generation of threats
• Need for new paradigms of combining alerts (observations)
• Process Query System (PQS) based approaches to detect complex attacks and covert channels
• Need of reducing the gap between user perception and what technology means (maybe explicit information about the real status of the system).
Many thanks to professor George Cybenko (Thayer School of Engineering at Dartmouth College) and professor Shankar Sastry (EECS, UC Berkeley).

agiani@eecs.berkeley.edu