

---

# Modeling and Containment of Search Worms Targeting Web Applications

Jingyu Hua\*, Kouichi Sakurai  
Information Technology & Security Lab  
Kyushu Univ.

Speaker: Jingyu Hua  
Email: [huajingyu@gmail.com](mailto:huajingyu@gmail.com)

\* He is partly supported by the Grant of Graduate school of ISEE of Kyushu University for Supporting Students' Overseas Traveling and the China Governmental Scholarship.

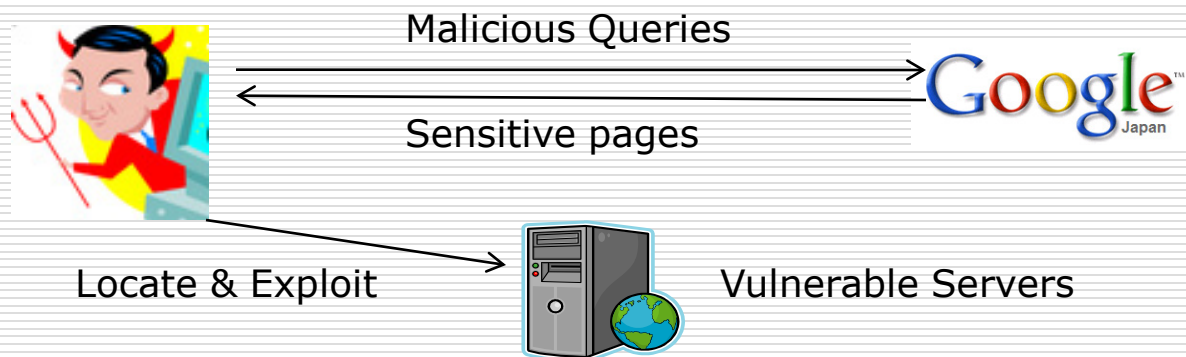
# Outline

---

- Background: What are Search Worms?
- Motivation: Modeling & Containing
- Modeling of Search Worms
- Containment of Search Worms
- Conclusion
- Future Work

# Background-Google Hacking

Google is a great tool for Hackers!!!



Google

filetype:inc intext:mysql\_connect

Google 搜索 高级

所有网页  中文网页  简体中文网页

网页 [+ 打开百宝箱...](#)

搜索 filetype:inc intext:mysql\_connect 获得约 1,130 条结果，以下是第 1-10 条。

[functions.inc - Index of / - \[ 翻译此页 \]](#)

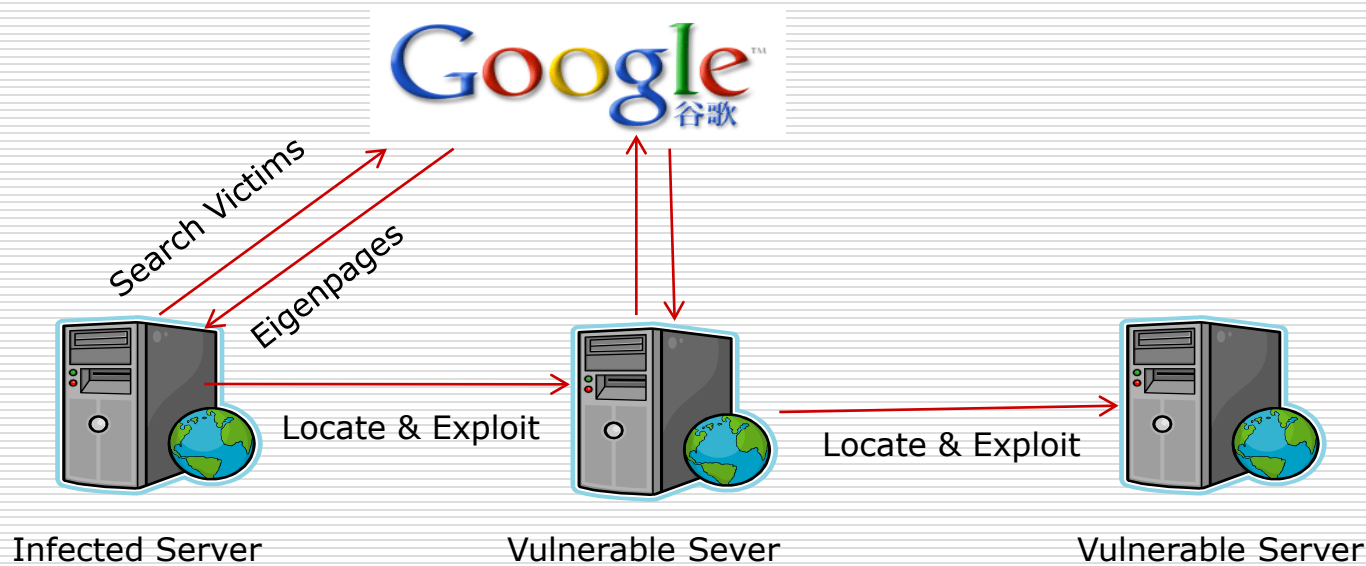
```
<?php function ListRedTree($Start) { $Node=array(); $DBLink=mysql_connect("localhost",  
"guest"); mysql_select_db("menutest",$DBLink); $Query="SELECT ID, ...
```

[hclements.com/Menu/functions.inc - 网页快照](#) - [网页快照](#) - [网页快照](#)

Google Dorks

**Eigenpages: dangerous pages that can disclose server vulnerabilities**

# Background-Search Worm



**Search Worm:** a worm uses search engines to locate targets

## Example:

Santy: targeting phpBB bulletin system.

Google Dork: *allinurl: viewtopic.php*

Released on 12/20/2004, infected **40,000** servers in **2** days

# Motivation

---

- Search Worms are dangerous! So, It deserves our research.
- Two common problems in the field of worm studying:
  - How to model the propagation of such worms ?  
( $I(t) = ?$ )
    - Study the spreading characteristics
    - Study the effects of containment strategies
  - How to contain the propagation of such worms?

---

# Topic 1

## Modeling of Search Worms

---

# Modeling of the Search Worms

## A virtual search worm

---

- Vulnerable servers leak eigenpages containing specific *keywords* to search engine.  
 $N$ : num of servers containing the eigenpages (Suspicious servers)  
 $V$ : num of really exploitable servers (Vulnerable servers)
  
- Propagation steps of infected servers:
  1. Search "*special-keywords* and *random-keywords*" in a search engine  $\rightarrow m$  search results
  2. Choose  $\delta$  pages among the total  $m$  search results to scan.
  3. Once a server is infected, it begins this infection cycle, too.

# Modeling of the Search Worms

## Effects of Eigenpage Distribution

---

- Obviously, servers contain more eigenpages are more likely to be exploited.
  
- Two attacker-favorable assumptions:
  - $m$  search results are randomly selected from all the eigenpages on the web
  - $\delta$  targets are randomly selected from the  $m$  search results



# Modeling of the Search Worms

## Effects of Eigenpage Distribution

---

- **U-Model:** eigenpages are uniformly distributed on servers

The number of infected servers by the end of time tick  $t$ :

$$I(t) = I(t-1) + [V - I(t-1)] \left[ 1 - \left(1 - \frac{1}{N}\right)^{\delta I(t-1)} \right]$$

Newly infected servers during the time tick  $t$

Remained vulnerable servers by the end of the time tick  $t-1$

The probability that a specific server is hit by a scan during the time tick  $t$

$V$  is the total count of servers really suffering the vulnerabilities among the  $N$  servers containing eigenpages

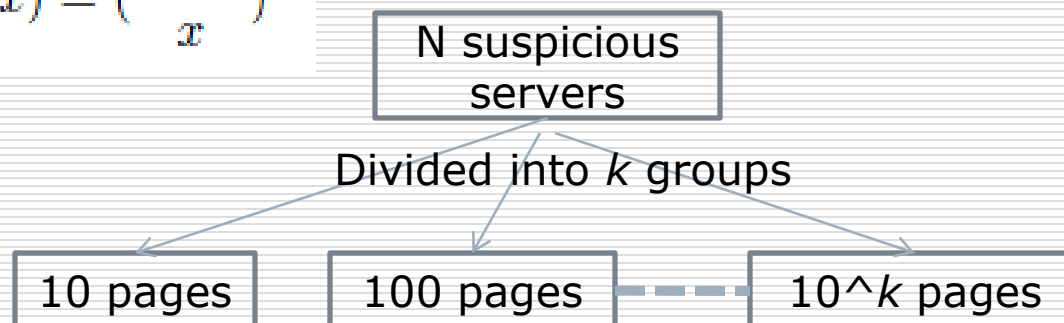
# Modeling of the Search Worms

## Effects of Eigenpage Distribution

- **PL-Model:** eigenpages follow a **power law distribution**.

The probability that the number of eigenpages  $p$  on a suspicious server is greater than  $x$  is

$$\text{prob}(p > x) = \left(\frac{p_{\min}}{x}\right)^\sigma$$



$$a_i = N \cdot [\text{prob}(p > p_i) - \text{prob}(p > p_{i+1})] = N \cdot \left[ \left(\frac{p_{\min}}{p_i}\right)^\sigma - \left(\frac{p_{\min}}{p_{i+1}}\right)^\sigma \right]$$

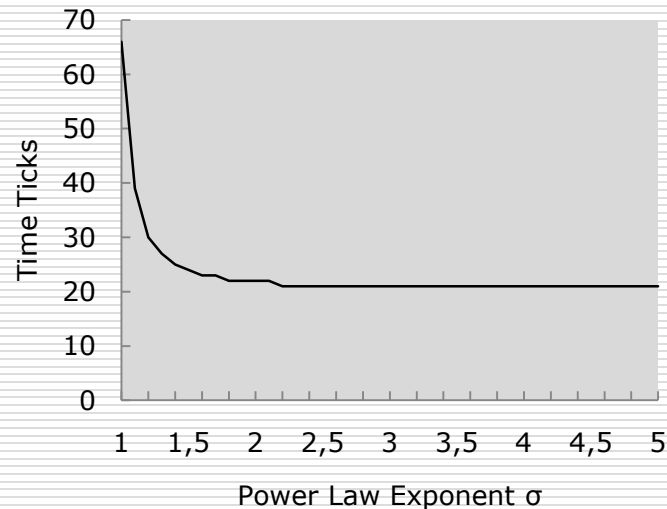
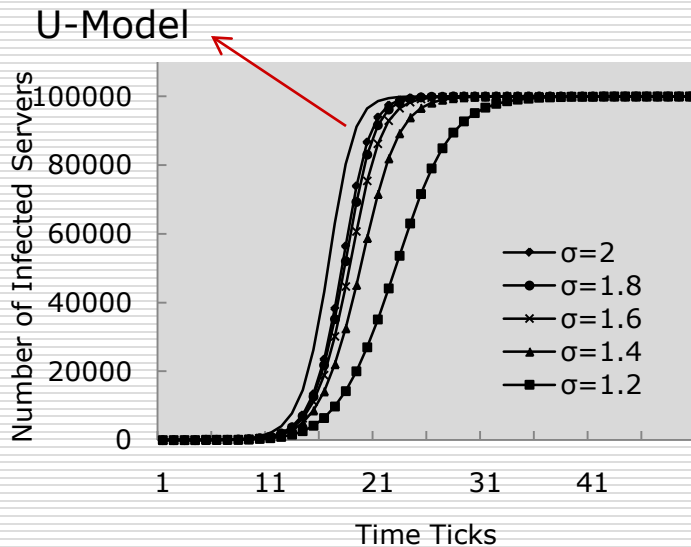
# Modeling of the Search Worms

## Effects of Eigenpage Distribution

### □ PL-Model:

Newly infected servers in the  $i$ -th group during the time tick  $t$

$$\begin{cases} I(t, i) = I(t - 1, i) + [V_i - I(t - 1, i)][1 - (1 - \frac{p_i}{P})^{\delta I(t-1)}] \\ I(t) = \sum_{i=1}^k I(t, i) \end{cases}$$



Time to infect  
95% vulnerable  
servers for the  
PL-Model

# Modeling of the Search Worms

## Effects of Eigenpage Distribution

---

### □ Proposition:

Among different distributions of eigenpages, the uniform distribution optimizes the performance of search worms.

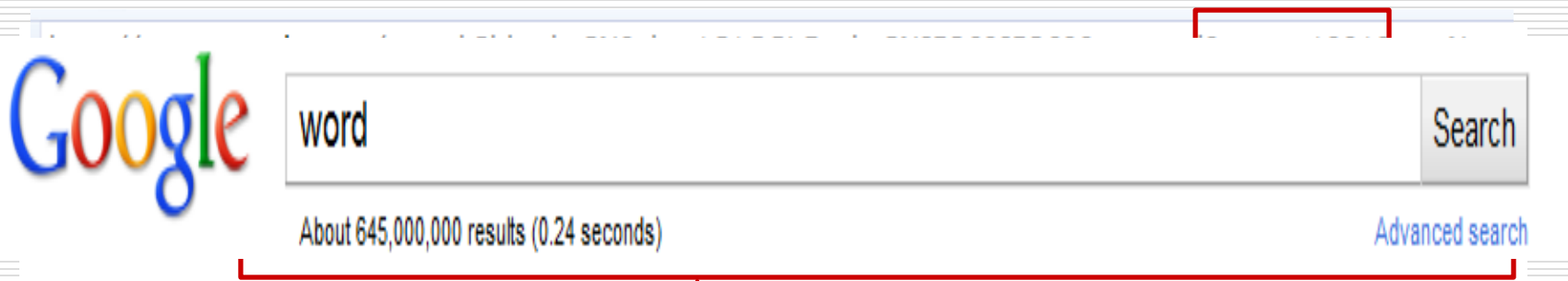
We proved this conclusion by using the mean value inequality

# Modeling of the Search Worms

## Effects of Page Ranking

---

- ❑ Search results are ranked according to Keyword relevance, page importance....
- ❑ Pages on popular servers are more likely to appear in front → scan collisions.
- ❑ If the second attacker-favorable assumption is true, no scan collisions.



However, the second assumption is impossible, the propagation of search worm will be affected by page ranking

# Modeling of the Search Worms

## Effects of Page Ranking

---

- Page importance:
  - Page Ranking Value (0-10)  
Power Law [Litvak 2007]

$$\text{prob}(PR = k) = \begin{cases} \left(\frac{1}{6^k}\right)^\alpha - \left(\frac{1}{6^{k+1}}\right)^\alpha & 0 \leq k < 10 \\ \left(\frac{1}{6^{10}}\right)^\alpha & k = 10 \end{cases}$$

- Site importance  
We simply assume servers contain more pages are more important

**[Litvak 2007]** Litvak, N., Scheinhardt, W. R. W., Volkovich, Y.: In-degree and PageRank: Why do they follow similar power laws? Internet Math, Vol.4(2-3), pp.175-198 (2007)

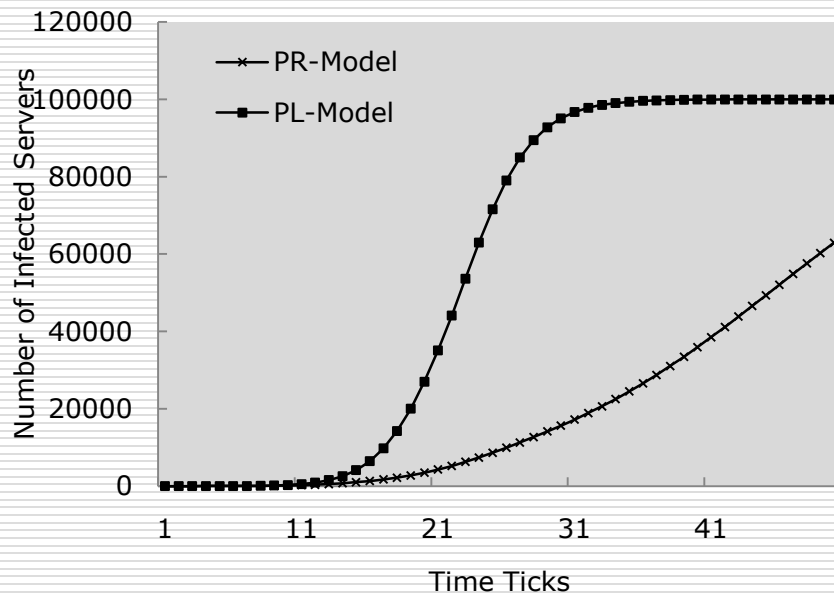
# Modeling of the Search Worms

## Effects of Page Ranking

### □ PR-Model:

An infected server selects the  $\delta$  top-ranking

$$\begin{cases} I(t, i) = I(t-1, i) + [V_i - I(t-1, i)][1 - (1 - \frac{1}{a_i})^{\delta I(t-1)}] \\ I(t) = \sum_{i=1}^k I(t, i) \end{cases}$$



Page ranking slows down the spreading of the search worm

---

## Topic 2: Containment of Search Worms

---



# Containment of the Search Worm

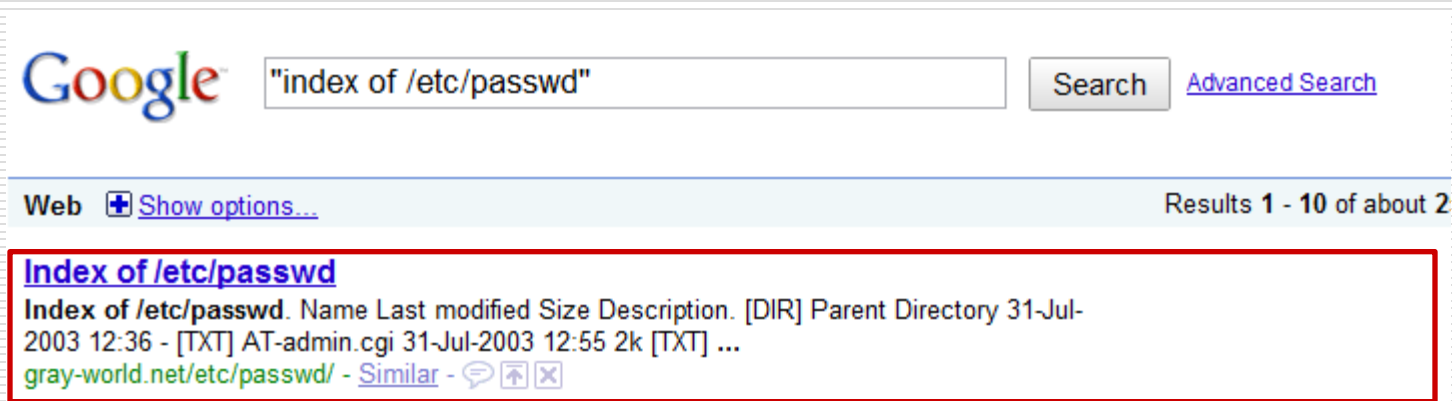
---

- The goal to model the search worm is to help developing an efficient containment system.
- We introduce a conceptual containment system based on honey-page insertion.
- We use our propagation models to analyze this system.

# Containment of the Search Worm

---

## □ Honey Page



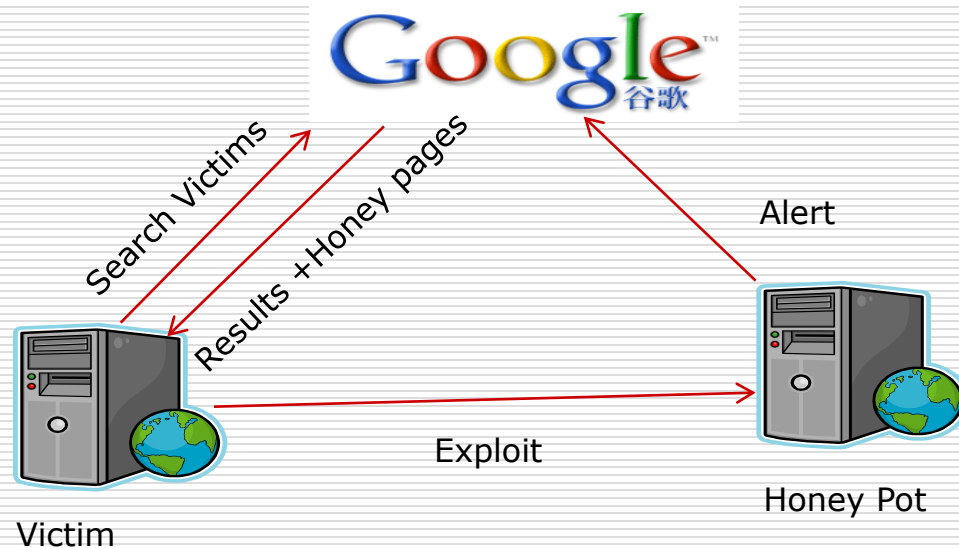
The screenshot shows a Google search interface. The search bar contains the text "index of /etc/passwd". Below the search bar, there is a "Search" button and a link to "Advanced Search". The search results are displayed below, with the first result highlighted by a red box. The result is titled "Index of /etc/passwd" and includes the following text: "Index of /etc/passwd. Name Last modified Size Description. [DIR] Parent Directory 31-Jul-2003 12:36 - [TXT] AT-admin.cgi 31-Jul-2003 12:55 2k [TXT] ...". Below the text, there is a link to "gray-world.net/etc/passwd/" and a "Similar" link. A red arrow points from the bottom of the red box to the text below.

This is a honey page pointing to a honey pot

# Containment of the Search Worm

## □ Containment based on honey-page insertion:

Search engine randomly inserts honey-pages into search results for any query.



When the search engine receives an alert, it denies further queries from detected victims

# Containment of the Search Worm

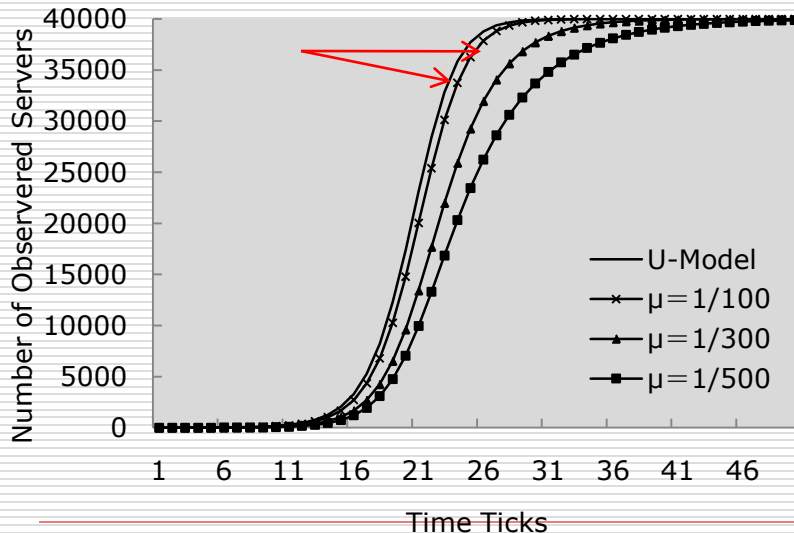
□ Is such a strategy possible?

$$D_{t+1} = D_t + (I_t - D_t) [1 - (1 - \mu)^\delta]$$

Honey page insert rate

# of detected infected nodes by the time t+1

Probability that an infected sever scans a honey page during an infection cycle



An infected sever can be induced to scan honeypots in a very short time after it is infected even if the insert rate is very small.

# Containment of the Search Worm

---

## □ Two questions:

### ■ Containment requirement $\rightarrow$ Insert Rate?

$$\mu = 1 - \left[ 1 + \frac{\gamma \varepsilon \delta}{\ln(1 - \gamma)} \right]^{1/\delta}$$

$\gamma$  is the final prevalence rate:  
ration of infected vulnerable  
servers

### ■ Is arbitrary requirement can be reached?

$\varepsilon \delta \leq 1$  No limitation, arbitrary requirement can be reached

$\varepsilon = \frac{V}{N}$  is the density of vulnerable servers.

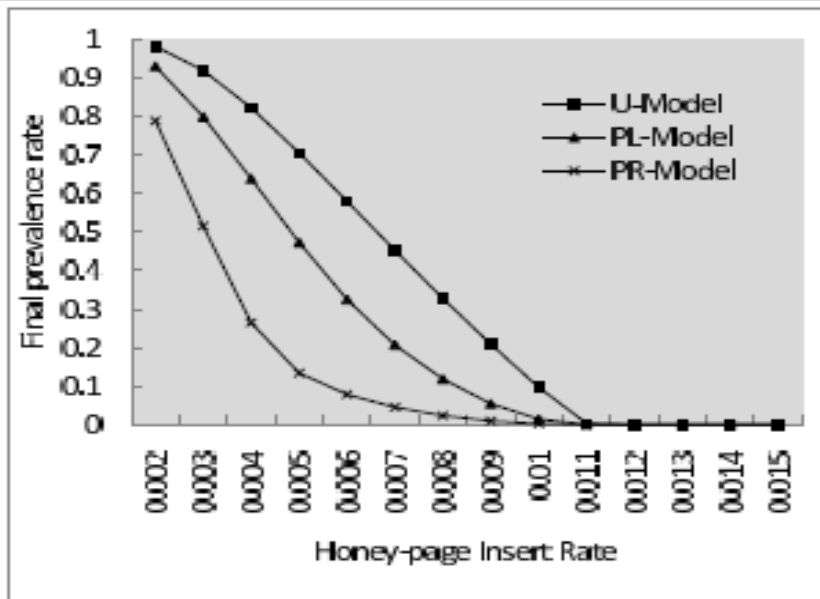
# Containment of the Search Worm

## □ Effectiveness for the Santy worm

$N=6,000,000$     $V=40,000$

To contain the final prevalence rate below 1%

Insert rate  $\rightarrow 0.011$



2 honey pages in every 100 search results can stop the spreading of the Santy worm at its early age!

# Conclusion

---

- Modeling of the Search Worm
  - Eigenpages Distribution: Uniform distribution optimizes the spreading
  - Page Ranking: slow downs the spreading
- Containment of the Search Worm
  - Honey page insertion
  - A small insert rate can lead a good containment effect

# Challenging Future Work

---

- Worm may validate the truth of the search results. Then, how to disguise honey pages as true ones both in URL and contents?
- Our current conclusions are based on simulation. Real experiments are required to verify them.



# Question & Answer

---

Thanks for your attention!