

## Worm Spread Models

- > They should help us understand how worms spread
- > Hopefully point out some weakness in the worm spread mechanism and help us design defenses
  - > Epidemic model
  - > Kermack-McKendrick model
  - > Two-factors model

1

## Epidemic Model

- > A host is in one of two states:
  - > Susceptible – has the vulnerability that the worm exploits, so it can be infected
  - > Infectious – was infected and is now trying to infect other hosts



2

## Epidemic Model

Where  $I(t)$  is number of infectious hosts at time  $t$ ,  $S(t)$  is number of susceptible hosts at time  $t$ ,  $\beta$  is the infection rate

$$\frac{dI(t)}{dt} = \beta * I(t) * S(t)$$

$N = I(t) + S(t)$   $N$  is population size

$$\frac{dI(t)}{dt} = \beta * I(t) * (N - I(t))$$

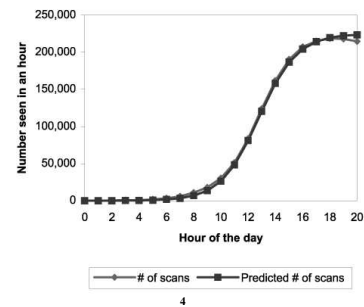
Dividing both sides by  $N^2$  we get the equation that models the fraction of the infected hosts —  $i(t) = I(t)/N$

$$\frac{di(t)}{dt} = \beta * i(t) * (1 - i(t))$$

Solving this equation gives:  
(for some constant of integration  $T$ )

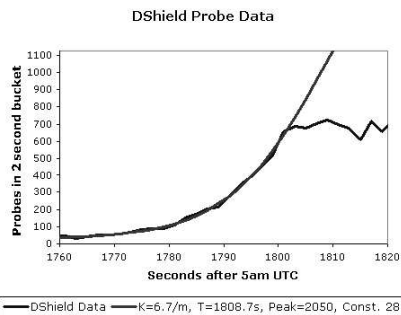
$$i(t) = \frac{e^{\beta(t-T)}}{1 + e^{\beta(t-T)}}$$

## Epidemic Model – Code Red



4

## Epidemic Model - Slammer



## Kermack-McKendrick Model

- > A host is in one of three states:
  - > Susceptible – has the vulnerability that the worm exploits, so it can be infected
  - > Infectious – was infected and is now trying to infect other hosts
  - > Removed – immune to the worm, or recovered from the infection, or disconnected from the network



6

### Kermack-McKendrick Model

$$\frac{dI(t)}{dt} = \beta * S(t) * I(t) - \frac{dR(t)}{dt}$$

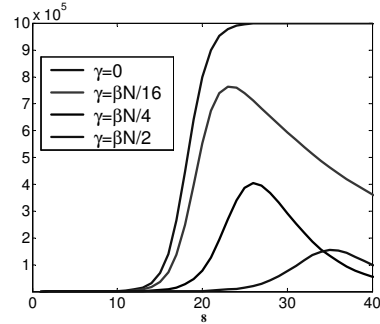
Where R(t) is number of infectious hosts that have been removed at time t

$$\frac{dR(t)}{dt} = \gamma * I(t)$$

$$S(t) + I(t) + R(t) = N$$

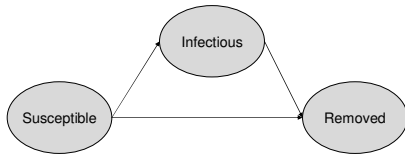
7

### Kermack-McKendrick Model



### Two-Factor Model

- Host can also be removed from susceptible state due to human action
- Congestion makes worm spread slow down so beta is variable — beta(t)



9

### Two-Factor Model

$$\frac{dS(t)}{dt} = -\beta(t) * S(t) * I(t) - \frac{dQ(t)}{dt}$$

Where Q(t) is number of susceptible hosts that have been removed at time t

$$\frac{dR(t)}{dt} = \gamma * I(t)$$

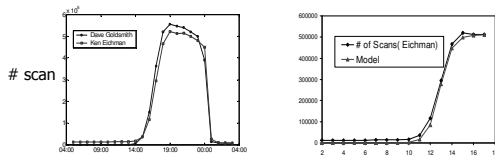
$$\frac{dQ(t)}{dt} = \mu * S(t) * (I(t) + R(t))$$

$$S(t) + I(t) + R(t) + Q(t) = N$$

$$\beta(t) = \beta_0 * (1 - I(t) / N)^\eta$$

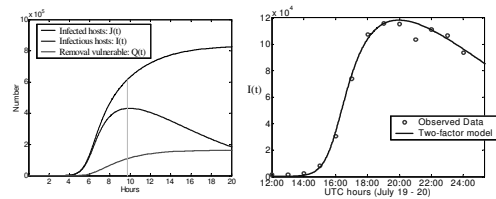
10

### Code Red – Epidemic Model



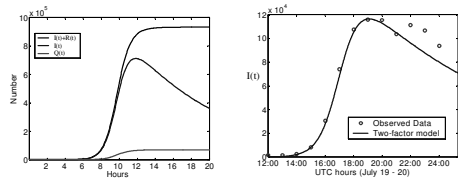
11

### Code Red – Two-Factor Model



12

### Code Red – Two-Factor Model With fixed $\beta$



13

### Worm Defense

- Three factors define worm spread:
  - Size of vulnerable population
    - Prevention – patch vulnerabilities, increase heterogeneity
  - Rate of infection (scanning and propagation strategy)
    - Deploy firewalls
    - Distribute worm signatures
  - Length of infectious period
    - Patch vulnerabilities after the outbreak

14

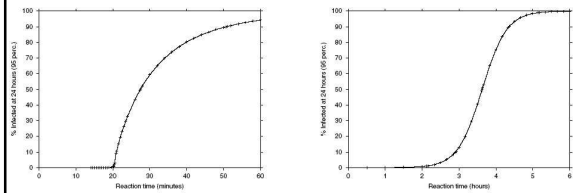
### How Well Can Containment Do?

- This depends on several factors:
  - Reaction time
  - Containment strategy – address blacklisting and content filtering
  - Deployment scenario – where is response deployed
- Evaluate effect of containment 24 hours after the onset

"Internet Quarantine: Requirements for Containing Self-Propagating Code",  
Proceedings of INFOCOM 2003, D. Moore, C. Shannon, G. Voelker, S. Savage

15

### How Well Can Containment Do? Code Red



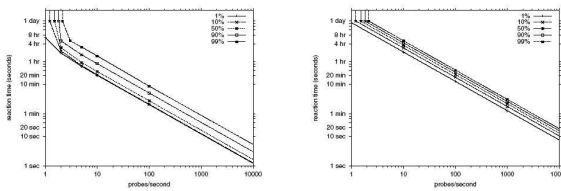
(a) Address Blacklisting

(b) Content Filtering

Idealized deployment

16

### How Well Can Containment Do? Depending on Worm Aggressiveness



(a) Address Blacklisting

(b) Content Filtering

Idealized deployment

17

### How Well Can Containment Do? Depending on Deployment Pattern

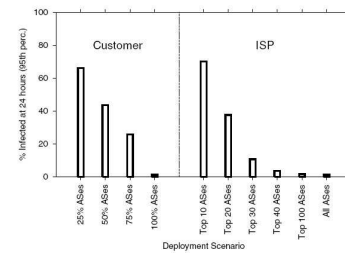
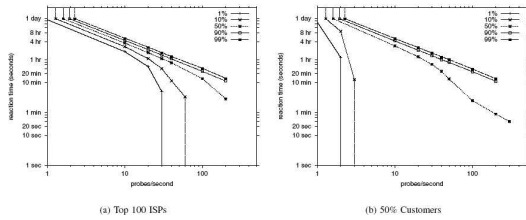


Fig. 4. Containment effectiveness as a function of deployment scenario.

18

## How Well Can Containment Do? Depending on Deployment Pattern And Worm Aggressiveness



19

## How Well Can Containment Do?

- Reaction time needs to be within minutes, if not seconds
- We need to use content filtering
- We need to have extensive deployment on the Internet

20

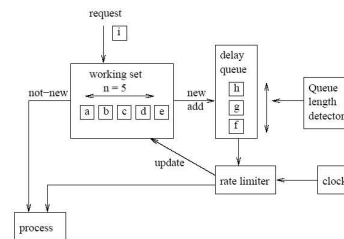
## Detecting and Stopping Worm Spread

- Monitor outgoing connection attempts *to new hosts*
- When rate exceeds 5 per second, put the remaining requests in a queue
- When number of requests in a queue exceeds 100 stop all communication
- A very neat idea how to test worm spread in a controlled manner – open inviting application on an unused port, only spread to machines that run this

"Implementing and testing a virus throttle", Proceedings of Usenix Security Symposium 2003, J. Twycross, M. Williamson

21

## Detecting and Stopping Worm Spread



22

## Detecting and Stopping Worm Spread

connections per second	stopping time	allowed connections
<i>Nimda</i>		
120	0.28s	1
<i>Test Worm</i>		
20	5.44s	5
40	2.34s	2
60	1.37s	1
80	1.04s	1
100	0.91s	1
150	0.21s	0
200	0.02s	0
<i>SQLSlammer</i>		
850	0.02s	0

Figure 3: Average time taken by the throttle to stop real and test worms

23

## Detecting and Stopping Worm Spread

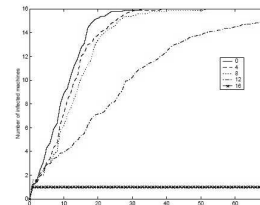


Figure 4: Infection times for different numbers of installed virus throttles (Nimda)

24

## Detecting and Stopping Worm Spread

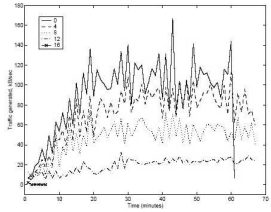


Figure 5: Traffic loads for different numbers of installed virus throttles (Nimda)

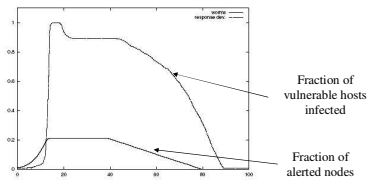
25

## Cooperative Strategies for Worm Defense

- Organizations share alerts and worm signatures with their “friends”
  - Severity of alerts is increased as more infection attempts are detected
  - Each host has a severity threshold after which it deploys response
- Alerts spread just like worm does
  - Must be faster to overtake worm spread
  - After some time of no new infection detections, alerts will be removed

“Cooperative Response Strategies for Large-Scale Attack Mitigation”,  
Proceedings of DISCEX 2003, D. Nojiri, J. Rogge, K. Levitt

## Cooperative Strategies for Worm Defense



Every node has 16 friends, worm spreads slowly

27

## Cooperative Strategies for Worm Defense

- As number of friends increases, response is faster
- Propagating false alarms may be a problem

28