

## Finding and stopping worms at backbone speeds

Georgios Portokalidis & Herbert Bos  
Vrije Universiteit Amsterdam

### SENTINELS

#### Two main approaches to worm detection

1. flow-based
2. payload inspection

#### Flow based approaches:

- fast
- inaccurate

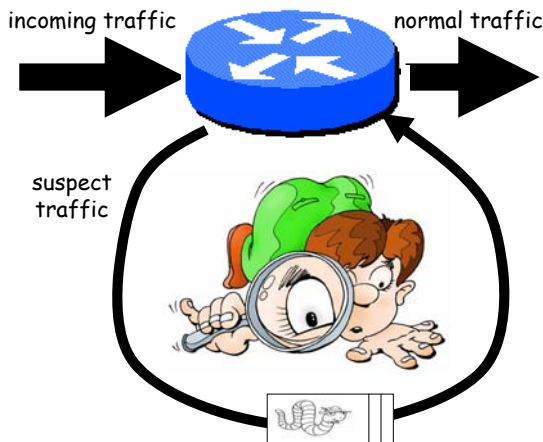
#### Content inspection

- slow
- accurate



#### 1 DeWorm combines the two approaches:

"We use a flow-based approach as a first tier filter which directs traffic towards a deep scan probe"



- #### 2 content inspection:
- look for similarity, not exact match
  - use reports from multiple sites

#### Advantages

- false positive rate of flow-based approach is less of a problem
- by distribution we can increase confidence and accuracy of signature generation
  - use reports from other sites to confirm
  - use signatures from other sites to find the smallest subset of the signature possible
  - by counting incidents we get a grip on the spread and virulence of individual worms

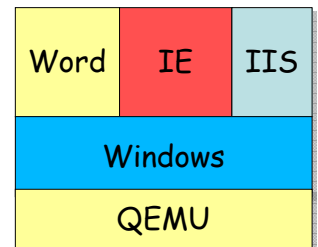
- #### 3 In addition, need to detect zero-day worms
- possibly encrypted
  - vulnerability may be unknown

→ Can only be done at edge of the network

Current design uses a deep scan approach that:

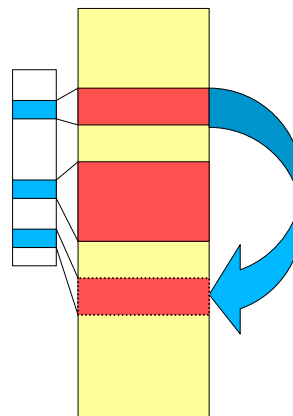
- catches all buffer overflow exploits (including zero-day and polymorphic/encrypted)
- also catches format string attacks
- generates no false positives

- #### 4 → run OS in x86 emulator
- modify emulator  
slowdown: 10-20x  
target: 2-4  
protect kernel, apps, *everything*



- #### 5 → use memory tainting
- all data coming from the network is tracked and marked as 'tainted'

track all operations that move tainted data



- copy to register
- arithmetic operations
- moves to/from memory
- etc.

- trigger at illegal use  
e.g. use of external data as jump target  
→ generate signature

network+host-based  
major challenge