HIVE: an Open Infrastructure for Malware Collection and Analysis

Davide Cavalca[1] Emanuele Goldoni[2]

University of Pavia, Italy

[1] Department of Computer Engineering and Systems Science
[2] Department of Electronics

1st Workshop on Open Source Software
for Computer and Network Forensics
2008

Goals

- a forensics approach to Internet malware and botnets
- self-spreading malware study and classification
- monitoring of attack trends and targets
- botnets behavior, structure and evolution

To achieve these goals we built an **automated** infrastructure for malware collection and analysis.

Outline

1. Introduction
   - Malware
   - Honeypot
2. HIVE
   - Architectural design
   - Experimental setup
   - Preliminary results
3. Conclusions

Outline

1. Introduction
   - Malware
   - Honeypot
2. HIVE
   - Architectural design
   - Experimental setup
   - Preliminary results
3. Conclusions

Malware

Malware = **mal**icious soft**ware**

- unwanted software with an agenda
    - virus
    - worm
    - trojan horse
    - spyware
    - ..
- malware spreads
    - automatically, relying on software bugs to self-replicate itself on new computer systems
    - manually, employing social engineering techniques against the users
- malware types
    - strictly destructive
    - for profit
        - SPAM and phishing
        - ransom requests
        - botnet construction

Botnet

- distributed network of autonomous programs (**bot**)
- created spreading *ad hoc* malware
    - infected computers turn into **zombie** systems
    - stealth behavior
- the attacker (**botherder**) remotely controls its botnet
    - using IRC or HTTP (centralized botnet)
    - using peer-to-peer protocols (distributed botnet)
- ...and rents its services to the best offer
    - criminal organizations
    - SPAM and advertisement
    - phishing
    - Distributed DoS attacks
    - "data mining"
- self-sustaining and reliable source of income

Honeypot

- decoy computer system designed to attract external attacks
  - human: study attacker behavior
  - automated: collect the malware binary code
- no valuable data (fake data sometimes used as bait)
- used to study attacks dynamics and attacker's tools
- sits on an otherwise unused IP space (**darknet**)
- **honeynet** = a network of honeypots

Honeypot: types

Low interaction honeypot

- software simulation of a computer system
- efficient: a single machine can simulate a large network
- not so effective: attack can fail due to simulation mishaps
- quick and easy to deploy, low TCO

High interaction honeypot

- a *real* vulnerable computer system
- very effective: the attacker compromises an actual system
- expensive to deploy and maintain, higher TCO
- legal liability issues

State of the art

We currently have:

- several low-interaction honeypot implementations
  - but there is no standardized framework for high-interaction honeypots
  - most works on the subject tend to reinvent the wheel
- a number of analysis services for malware samples

What we lack is an integrated framework encompassing the collection of samples, the analysis of malware and the monitoring of detected threats.

Outline

1. Introduction
   - Malware
   - Honeypot
2. HIVE
   - Architectural design
   - Experimental setup
   - Preliminary results
3. Conclusions

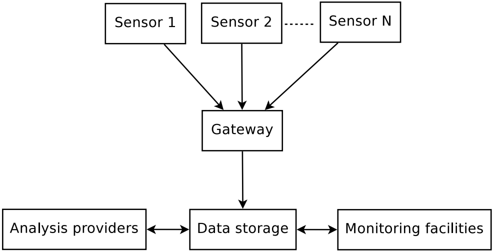
HIVE

HIVE = Honeypot Infrastructure in Virtualized Environment

- integrated infrastructure for malware collection and analysis
- fully automated
- open architecture
  - easy to interact with
  - easy to extend with new tools
- based on proven Open Source software

HIVE: architecture

- three-layered architecture
  - sensors (honeypot)
  - gateway
  - data storage and analysis
- layers are fully decoupled
- extensible and scalable

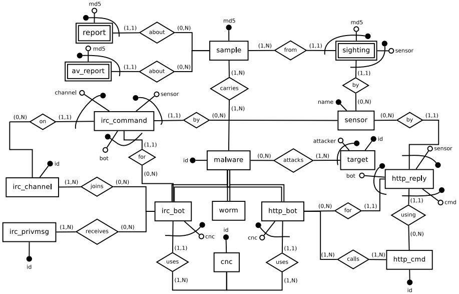


HIVE: honeynet

- a combination of low and high interaction systems
- extensive use of virtualization techniques (VirtualBox)
- automated self-maintenance
    - malware samples collection
    - honeypot systems rebuild

HIVE: database

- use of a relational DBMS (**PostgreSQL**)
  - malware samples storage
  - central repository for all acquired data
  - database views allow aggregate high-level data reporting



- automated samples analysis
  - static analysis: antivirus
  - behavioral analysis
    - CWSandbox
    - Anubis

HIVE: monitoring facilities

Malware analysis provides information on botnets

- C&C address location
- botnet login information

Computer programs disguised as zombies can then infiltrate the botnets

- zombie activity monitoring
- attacks issued
- botnet size and expansion
- attacker behavior and targets

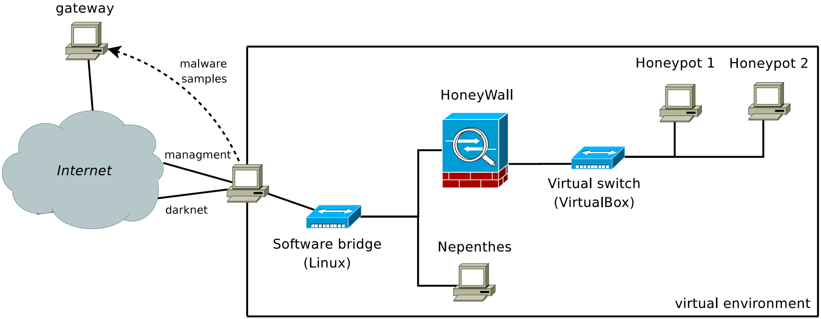
Tools developed

- **infiltrator** (originally by Göbel) for IRC botnets
- **httpmole** for HTTP botnets

Experimental setup

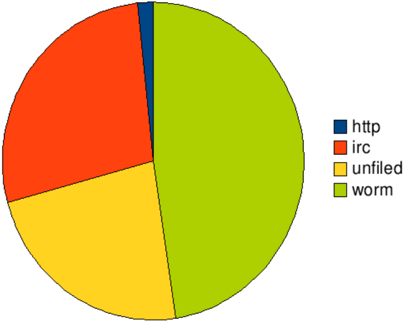
- three-systems virtual honeynet
  - Windows XP
  - Windows 2000 Server
  - Nepenthes (low interaction)
  - deployed on a single physical machine
- darknet: three contiguous IPs on an unprotected commercial network
  - perimetral defense systems can dramatically lower the honeynet efficency

HIVE: honeynet



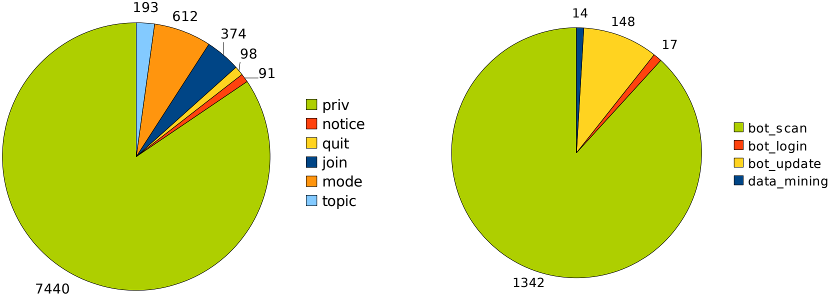
Results: collected malware

- during a month of operation
  - we collected more than 14k malware samples...
  - with 13k unique samples
  - over 50 centralized botnets were monitored



Results: botnets monitoring

Commands detected on the IRC botnets control channels



Outline

1. Introduction
   - Malware
   - Honeypot
2. HIVE
   - Architectural design
   - Experimental setup
   - Preliminary results
3. Conclusions

Conclusions and future works

To summarize

- botnets are an actual and widespread menace
- HIVE has proved to be an efficient tool for malware collection and analysis

In the future

- reporting engine
- HoneyWall integration
- peer-to-peer botnet study

Availability

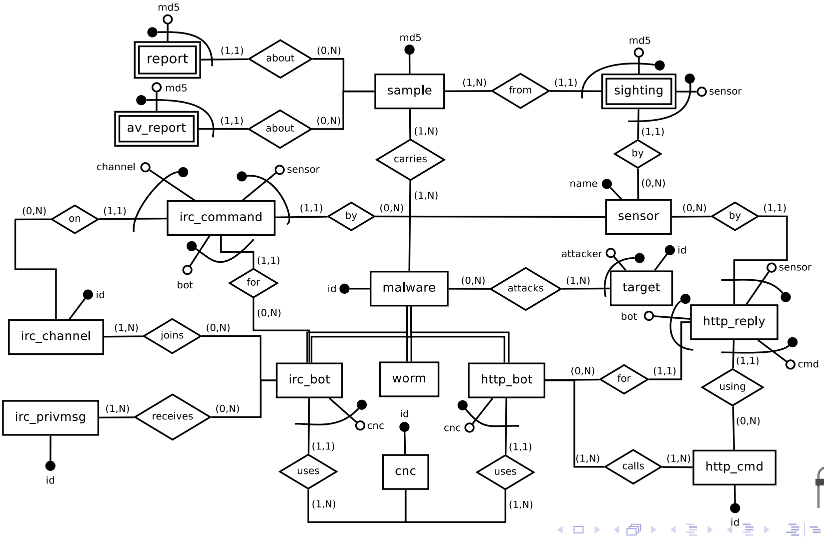
The HIVE software and this presentation can be downloaded from

`http://netlab-mn.unipv.it/hive`

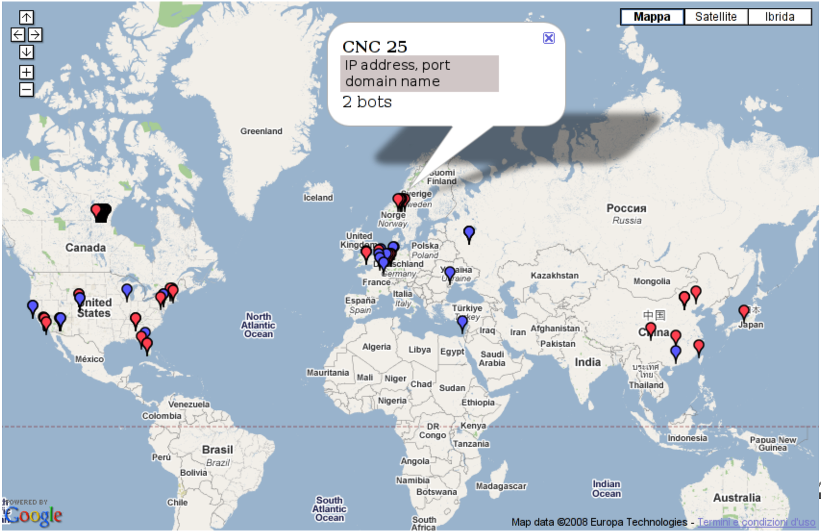
Future updates will be published at the same location.

Questions?

Simplified Entity-Relationship diagram



Botnets C&C map



CNC 25

IP address, port
domain name

2 bots

Chart: sightings

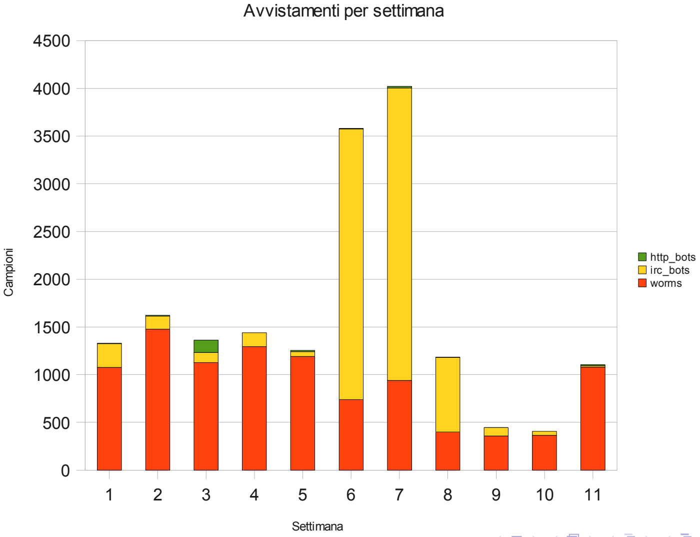
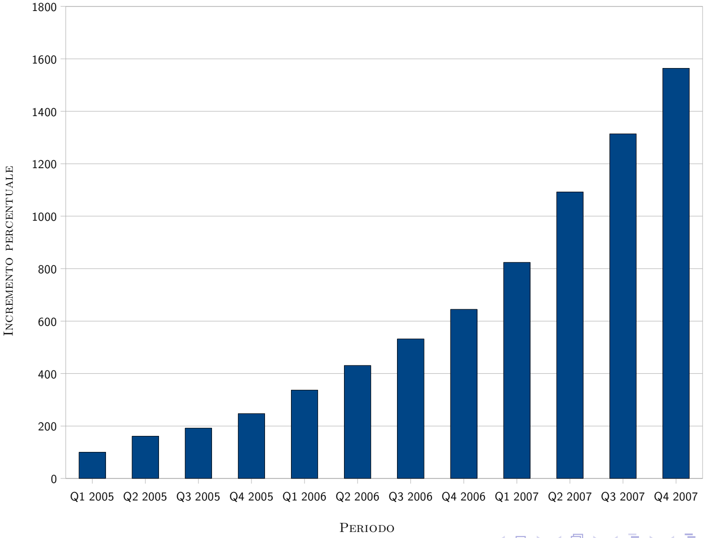


Chart: malware advance


Incremento percentuale
Q1 2005
Q2 2005
Q3 2005
Q4 2005
Q1 2006
Q2 2006
Q3 2006
Q4 2006
Q1 2007
Q2 2007
Q3 2007
Q4 2007
Periodo