Honey@home

Spiros Antonatos
antonat@ics.forth.gr

Distributed Computing Systems Lab (DCS)
Institute of Computer Science (ICS)
Foundation for Research and Technology Hellas (FORTH)
A few words about NoAH

- Network of Affined Honeypots
- EU-funded 3 year project (2005-2008)
- Develop an infrastructure to detect and provide early warning of cyberattacks
- Gather and analyse information about the nature of these attacks
- More info at http://www.fp6-noah.org
Outline

• Introduction
• Motivation
• Honey@home
• Architecture
• Challenges and how to face them
What are honeypots?

- Computer systems that do not provide production services
- Listening to unused IP address space
- Intentionally made vulnerable
- Closely monitored to analyse attacks directed to them
- Usually run inside a containment environment
  - Virtual machines
Motivation (1/2)

- There is unused IP address space
  - Large universities and research centers
    - UCSD, allocated a /8, only few thousands used
    - FORTH
    - CSD
  } Allocated a /16 each
  } utilization under 40%

- Organizations and private companies
- Public domain bodies
- Upscale home users
- NAT-based home networks
  - 192.168.*.*
Motivation (2/2)

• Social aspect
  – **Empower the people**
  – With minimal installation overhead
  – Minimal runtime overhead

• Appropriate for organizations
  – Who want to contribute
  – But do not have the technical knowledge
    • To install/maintain a full-fledged honeypot
Honey@home

- Enables willing users and organizations to effortlessly participate in a distributed honeypot infrastructure
  - No configuration needed, install and run
  - Both Windows and Linux platforms
- Runs in the background, sends all traffic from the dark space to NoAH core for processing
- Attacker think they communicate with a home computer but actually talks with honeypots
Install...

Welcome to the installation process.

Select installation folder:
- C:\Program Files

WARNING: This is an unauthorized duplicate or criminal penalty.

Install Honey@Home:
- Everyone
- Just me

The installer will follow the selected file path.

To install in this location, you need to agree:

- Everyone
- Just me

Please wait...

Honey@Home has been successfully installed.

Click "Close" to exit.

Please use Windows Update to check for any critical updates to the .NET Framework.

22 January 2007

20th TF-CSIRT meeting

Spiros Antonatos
...and run

1. Running at the background

2. Creating a new virtual interface

3. Getting an IP address from DHCP server
Features

• Can obtain address from DHCP or statically
• BPF filters can be used
  – Useful to get traffic from the whole unused subnet
• NAT detection and automatic port forwarding
  – Mostly for DSL users and small enterprises that are behind NAT
• Graphic overview of traffic statistics captured by the client
• Automatic updates
- Honey@home clients connect to NoAH core
- Honeyd as front-end to filter out scans
  - Filters out scans and unfinished connections
- Honeyd hands off connection to Argos
- Argos is an instrumented virtual machine able to catch zero-day exploits
  - Detects when code coming from the network is executed
  - [http://www.few.vu.nl/argos/](http://www.few.vu.nl/argos/)
Challenges

• We cannot trust clients
  – Anyone will be able to set up honey@home
• Clients must not know the address of honeypots
  – Honeypots may become victims of direct attacks
• Addresses of clients must also remain hidden
  – Attacker can use their black space for flooding
  – Or blacklist them to make NoAH core blind
• Computer-based mass installation of honey@home mockup clients should be prevented
Hiding honeypots and clients

- Use of anonymous communication system
- Onion routing is an attractive solution
  - Prevents eavesdropping attacks
  - Based on a set of centralized nodes (onion routers)
  - Even when a router is compromised, privacy is preserved
- Tor, an implementation of second generation onion routing
How onion routing works

- Sender chooses a random sequence of routers
  - Some routers are honest, some controlled by attacker
  - Sender controls the length of the path
- Routing info for each link encrypted with router’s public key
- Each router learns only the identity of the next router

Alice → R1 → R2 → R3 → R4 → Bob

(R2, k1)_{pk(R1)}, (R3, k2)_{pk(R2)}, (R4, k3)_{pk(R3)}, {B, k4}_{pk(R4)}, {M}k4, {M}k3, {M}k2, {M}k1
Hidden services

• In previous examples, Alice needed to know the address of Bob
  – That is client needs to know the address of honeypots
  – **We need to hide our honeypots**

• Tor offers hidden services
  – Clients only need to know an identifier for the hidden service
  – This identifier is a DNS name in the form of “xyz.onion”
  – “.onion” is routable only through Tor
Hidden services in action

*We created a hidden service that actually forwards to Google.com*
Preventing automatic installation

• Goal: prevent attacker from deploying clients to its botnet

• CAPTCHAs as a proposed solution
  – Instruct human to solve a visual puzzle
  – Puzzle cannot be identified by a computer
  – Puzzle can also be an audio clip
Enhancing CAPTCHAs

- Attacker may post the image to his site and use visitors to solve it
- Adding animation to avoid “CAPTCHA” laundry
- User clicks on the correct (animated) answer to continue with the registration
  - Animation prevents users to provide static responses, like “I clicked the upper left corner”
- We use the Java applet technology
Enhancing CAPTCHAs

honey@home

Home
Downloads
Registration Page
Documentation
Database
Links
Related Projects
NOAH
Argos
HoneyD

ANIMATED CAPTCHA TEST

Click on the apple to continue with registration.
Questions?
Honey@home

Spiros Antonatos
antonat@ics.forth.gr

Distributed Computing Systems Lab (DCS)
Institute of Computer Science (ICS)
Foundation for Research and Technology Hellas (FORTH)