Advanced Network Security

Botnets

Amir Mahdi Sadeghzadeh, Ph.D.

Most slides have been adapted from Mehdi Kharrazi, CE40-817, Sharif University of Technology, 2015

Definition

- Bots
  - Definition: autonomous programs automatically performing tasks, absent a real user.
- Botnets
  - Definition: networks of autonomous programs capable of acting on instructions.

Rise of Botnets

- 2003: 800-900,000 infected hosts, up to 100K nodes per botnet
- 2006: 5 million distinct bots, but smaller botnets
  - Thousands rather than 100s of thousands per botnet
  - Reasons: evasion, economics, ease of management
  - More bandwidth (1 Mbps and more per host)
- For-profit criminal activity (not just mischief)

Botnets as a Root cause

- Distributed DoS
- Spamming
- Click fraud attacks
- Cheating in online polls/games
- … many others

Botnets – Money matters!

- CPM
  - Cost Per Mille or Cost Per Thousand Impressions
- For regular banners you would get 2-3$/1000 views
- For some ads you would get much higher rate
- Let's say you have an ad for 5$/1000 views
  - If you have it viewed 1 million times, you will make $5000

Denial of Service (DoS)

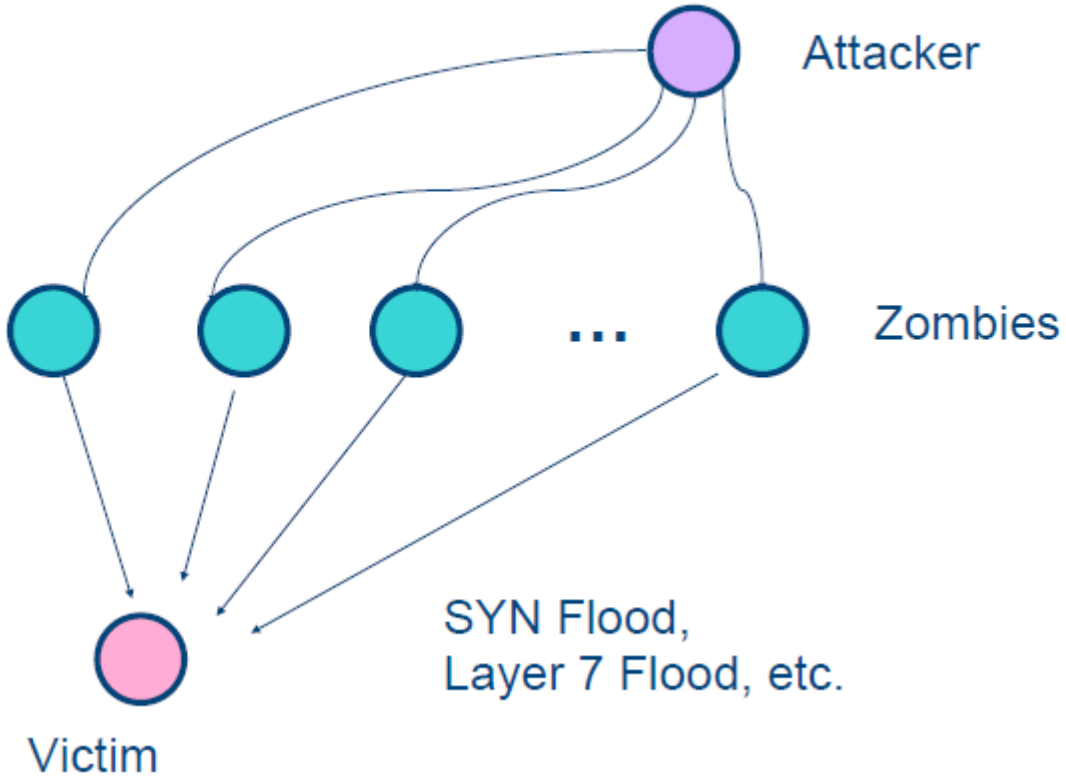
- Goal: overwhelm victim machine and deny service to its legitimate clients
- DoS often exploits networking protocols
  - Smurf: ICMP echo request to broadcast address with spoofed victim's address as source
  - SYN flood: "open TCP connection" request from a spoofed address
  - UDP flood: exhaust bandwidth by sending thousands of bogus UDP packets

Distributed Denial of Service (DDoS)

- Build a botnet of zombies
  - Multi-layer architecture: use some of the zombies as "masters" to control other zombies
- Command zombies to stage a coordinated attack on the victim
  - Does not require spoofing (why?)
  - Even in case of SYN flood, SYN cookies don't help (why?)
- Overwhelm victim with traffic arriving from thousands of different sources

Attack Update

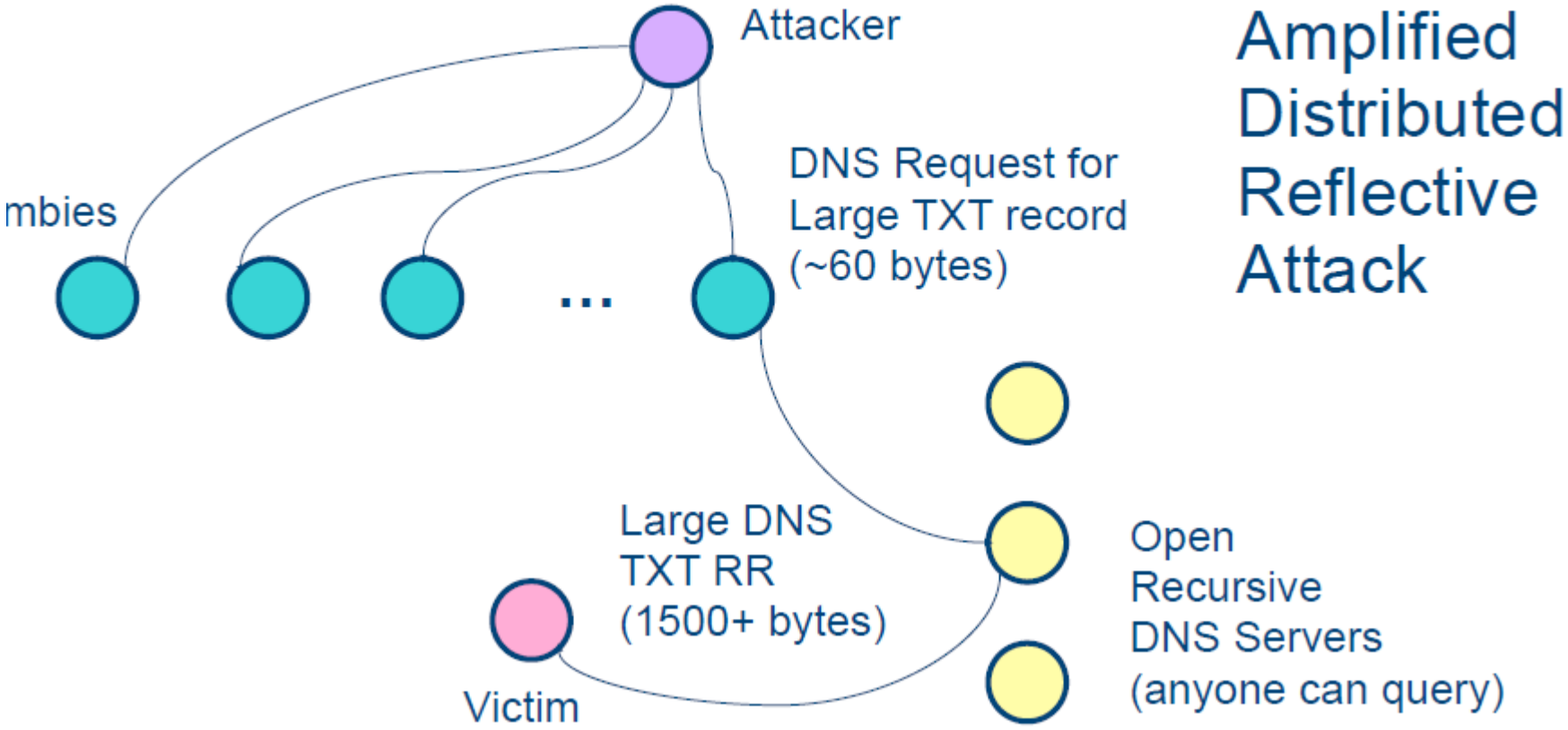
- Botnets of course are used for DDoS



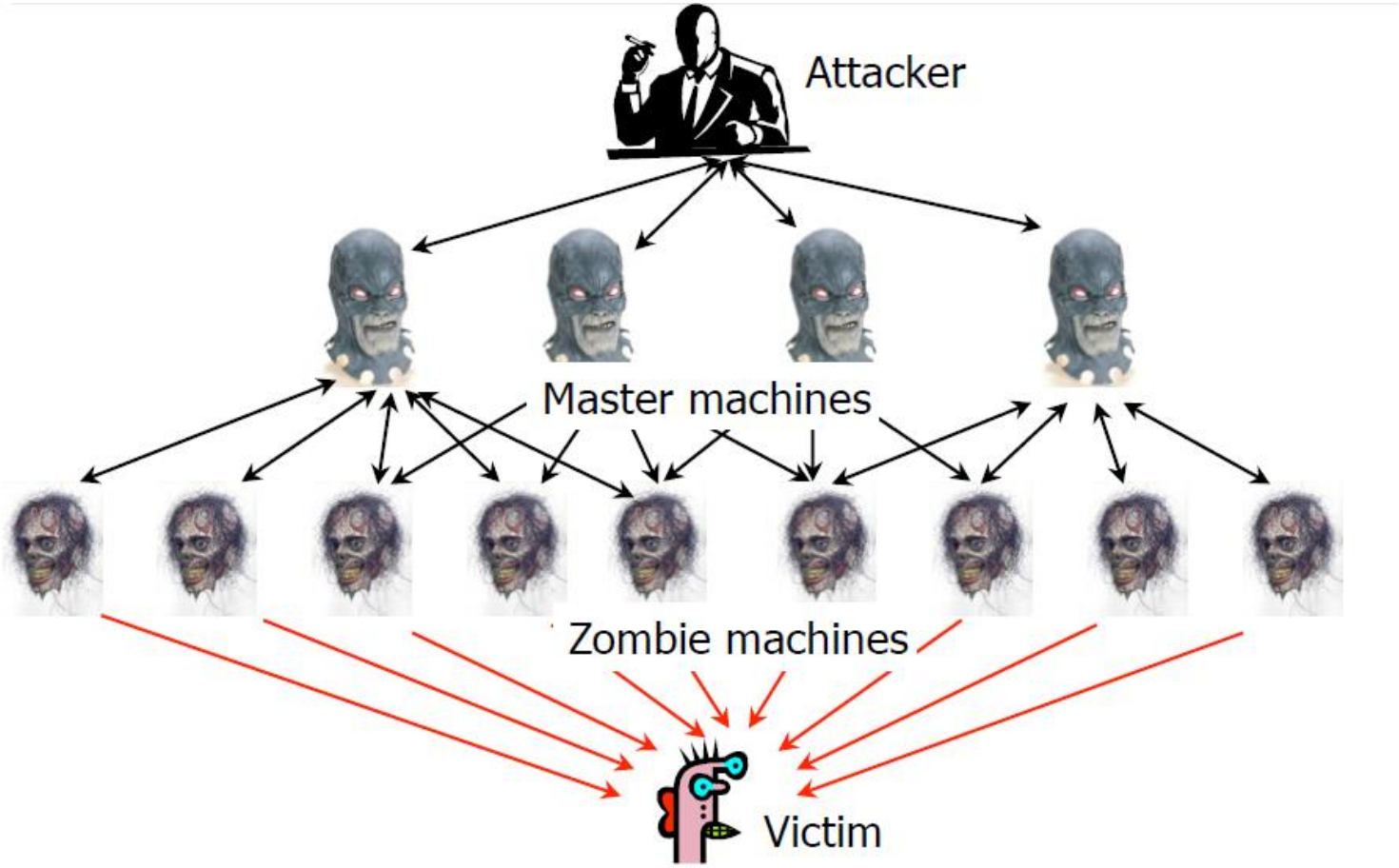
Typical Distributed Denial of Service (DDoS)

Attack Update

- Botnets increasingly used for amplified distributed reflective attacks



DDoS Architecture

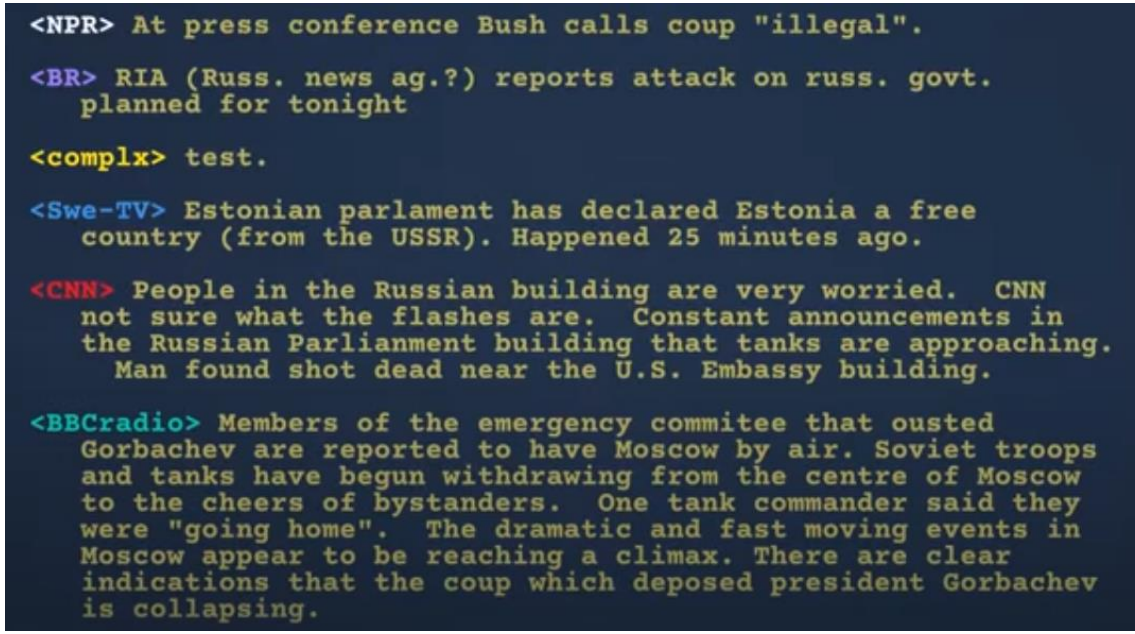


Bot History

- Eggdrop (1993): early IRC bot
- DDoS bots (late 90s): Trin00, TFN, Stacheldracht
- IRC bots (mid-2000s)
  - •Active spreading, multiple propagation vectors
  - •Include worm and trojan functionality
  - •Many mutations and morphs of the same codebase
- Stormbot and Conficker (2007-09)

Life Cycle of an IRC Bot

- Exploit a vulnerability to execute a short program (shellcode) on victim's machine
  - Buffer overflows, email viruses, etc.
- Shellcode downloads and installs the actual bot
- Bot disables firewall and antivirus software
- Bot locates IRC server, connects, joins channel
  - Typically **need DNS** to find out server's IP address
    - Especially if server's original IP address has been blacklisted
  - Password-based and crypto authentication
- Botmaster issues authenticated commands



IRC

Command and Control

```
(12:59:27pm) -- A9-pcgbdv (A9-pcgbdv@140.134.36.124)
has joined (#owned) Users : 1646

(12:59:27pm) (@Attacker) .ddos.synflood 216.209.82.62

(12:59:27pm) -- A6-bpxufrd (A6-bpxufrd@wp95-
81.introweb.nl) has joined (#owned) Users : 1647

(12:59:27pm) -- A9-nzmpah (A9-nzmpah@140.122.200.221)
has left IRC (Connection reset by peer)

(12:59:28pm) (@Attacker) .scan.enable DCOM

(12:59:28pm) -- A9-tzrkeasv (A9-tzrkeas@220.89.66.93)
has joined (#owned) Users : 1650
```

Trin00

- Scan for known buffer overflows in Linux & Solaris
  - Unpatched versions of wu-ftpd, statd, amd, …
- Install attack daemon using remote shell access
- Send commands (victim IP, attack parameters), using plaintext passwords for authentication
  - Attacker to master: TCP, master to zombie: UDP
  - To avoid detection, daemon issues warning if someone connects when master is already authenticated
- August of 1999: a network of 227 Trin00 zombies took U. of Minnesota offline for 3 days

Tribal Flood Network

- Supports multiple DoS attack types
  - Smurf; ICMP, SYN, UDP floods
- Attacker runs masters directly via root backdoor; masters talk to zombies using ICMP echo reply
- List of zombie daemons' IP addresses is encrypted in later versions of TFN master scripts
  - Protects identities of zombies if master is discovered

Stacheldraht

- Combines "best" features of Trin00 and TFN
  - Multiple attack types (like TFN)
- Symmetric encryption for attacker-master connections
- Master daemons can be upgraded on demand
- February 2000: crippled Yahoo, eBay, Amazon, Schwab, E*Trade, CNN, Buy.com, ZDNet
  - Attack on Yahoo consumed more than a Gigabit/sec of bandwidth
  - Sources of attack still unknown

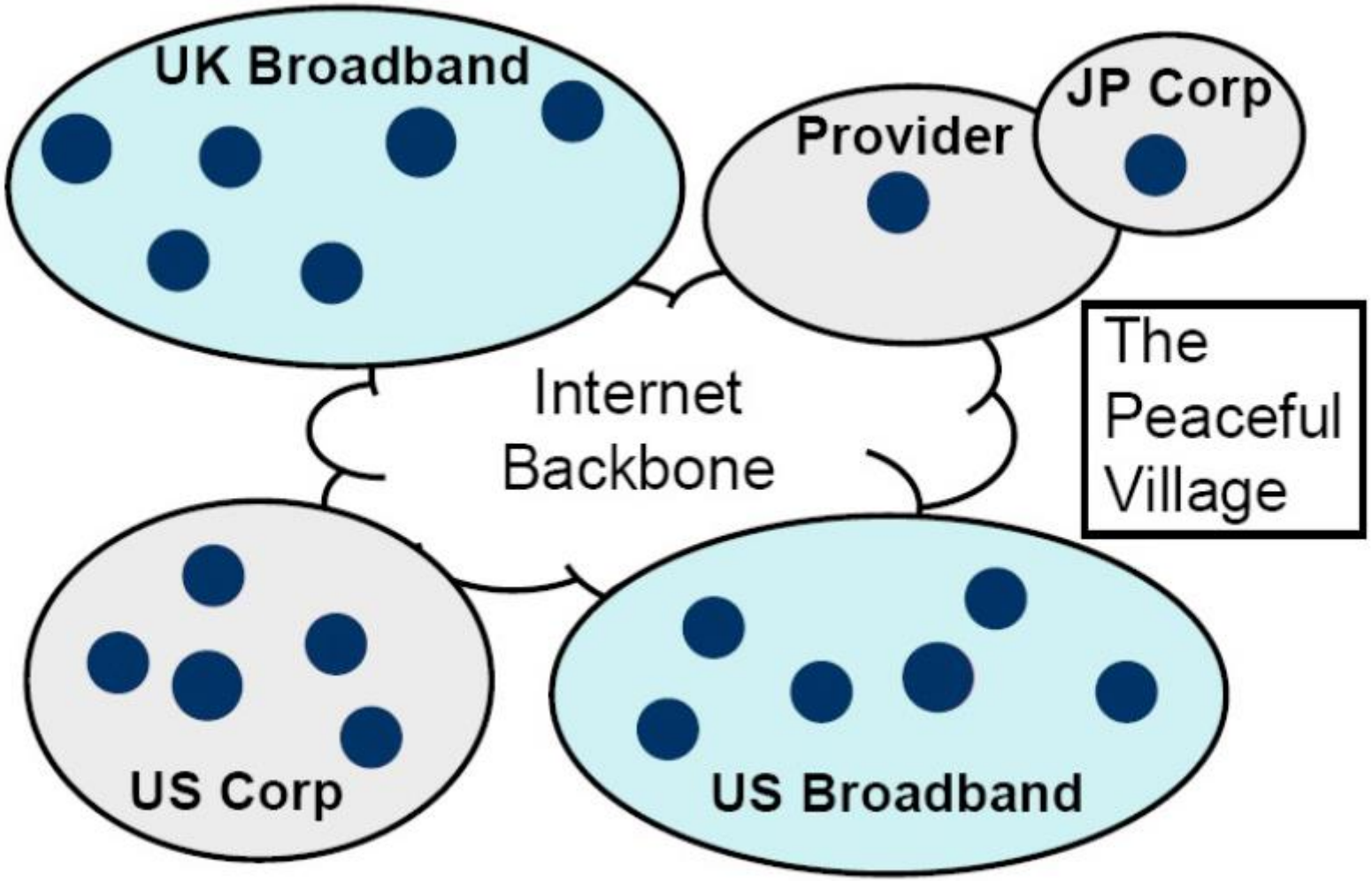
Agobot

- 20,000 lines of C/C++ code
- IRC-based command and control
- Scanning tools, many propagation vectors
- Capable of many DoS flooding types
- Code obfuscation to avoid detection
- Installs sniffer, terminates anti-virus processes, points DNS for anti-virus to localhost

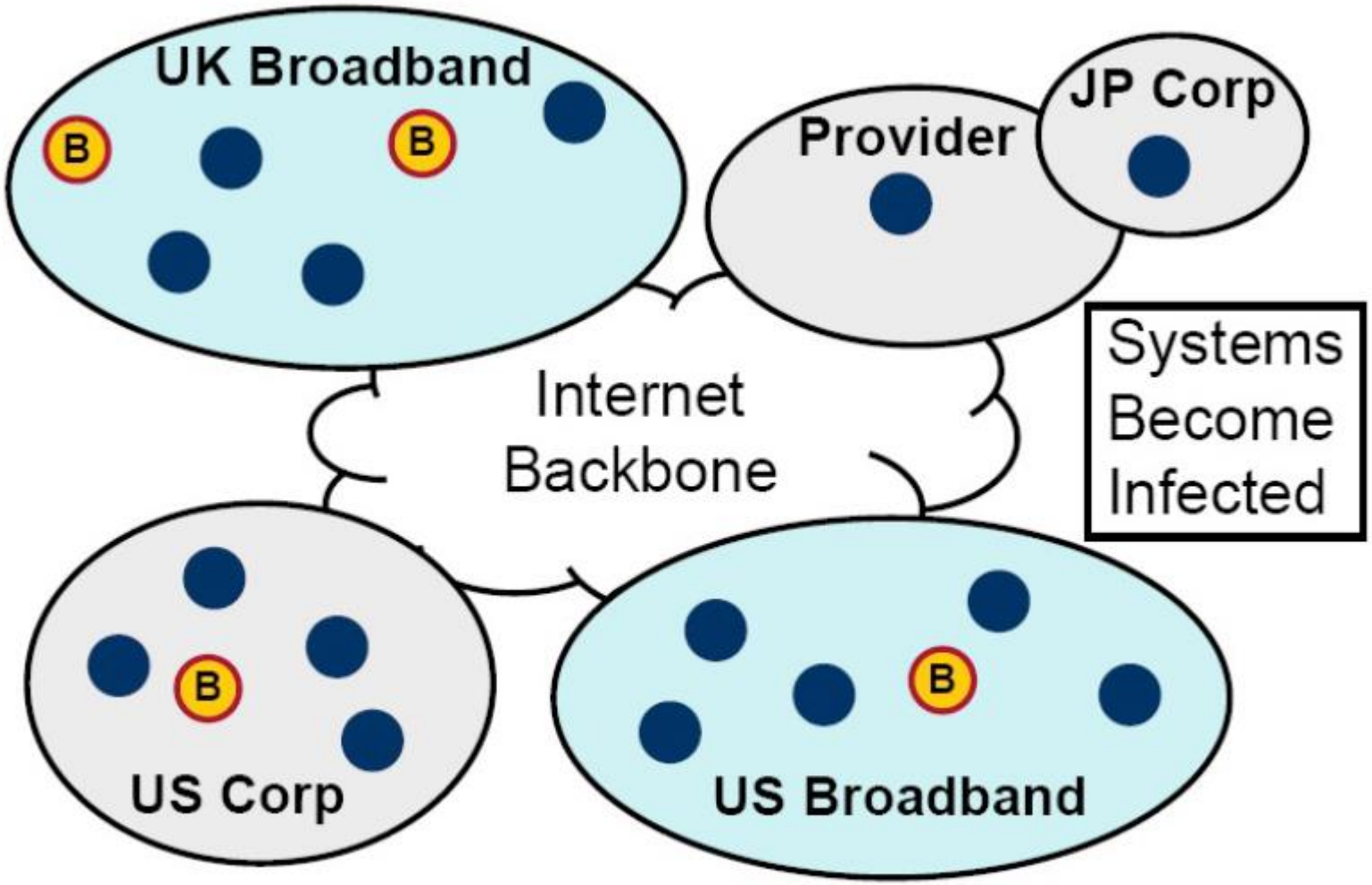
Other Modern Bots

- SDBot / SpyBot
  - Non-malicious, but can be extended for scanning, sniffing, DoS attacks
- GT-Bot
  - Renamed mIRC
  - Scanning, DoS, RPC and NetBIOS exploits
  - Simpler than Agobot
    - 2-3,000 lines of C code
  - Extensible and customizable codebase
- Trend: hybrids of bots, trojans, worms

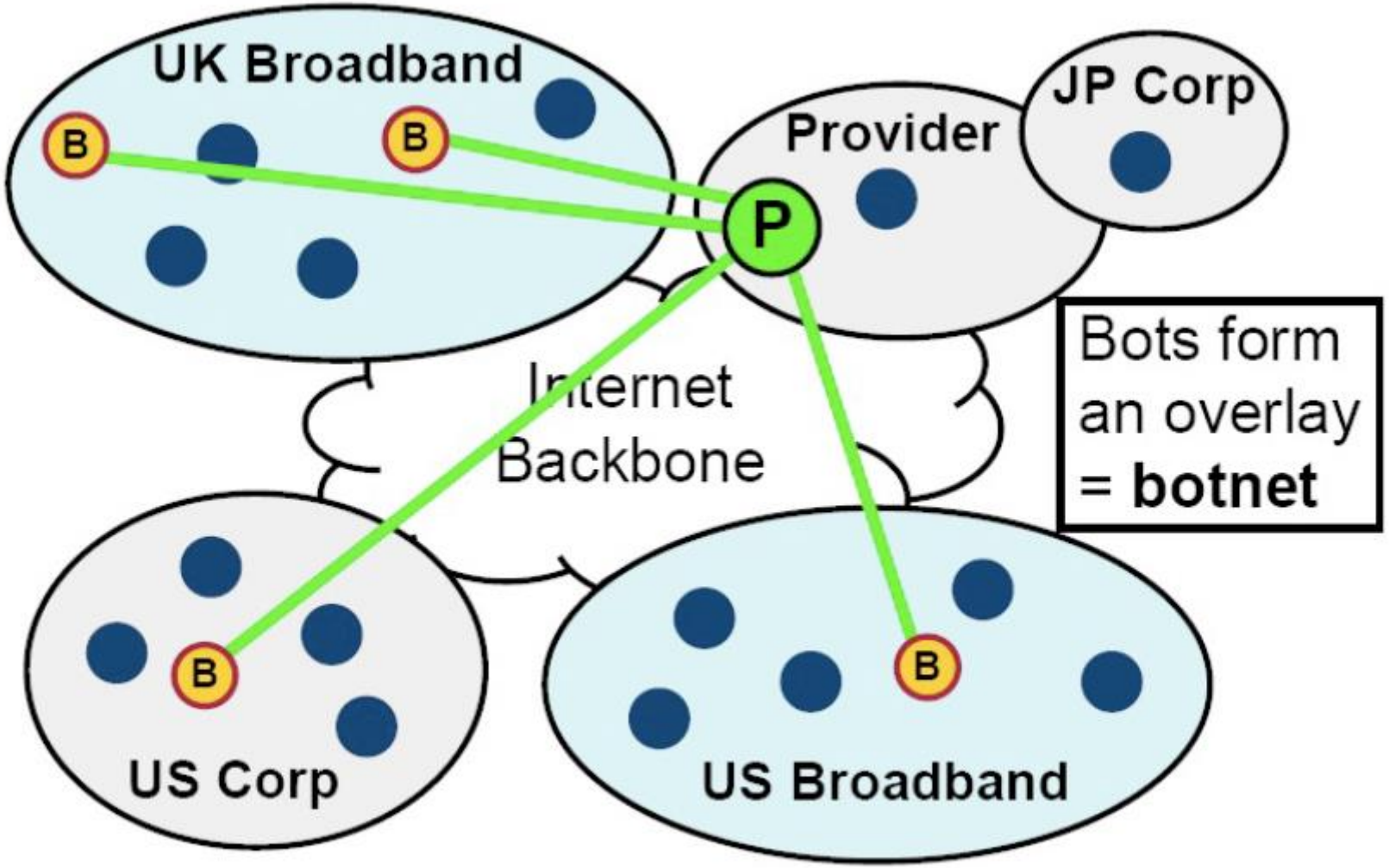
Botnet creation (1/5)



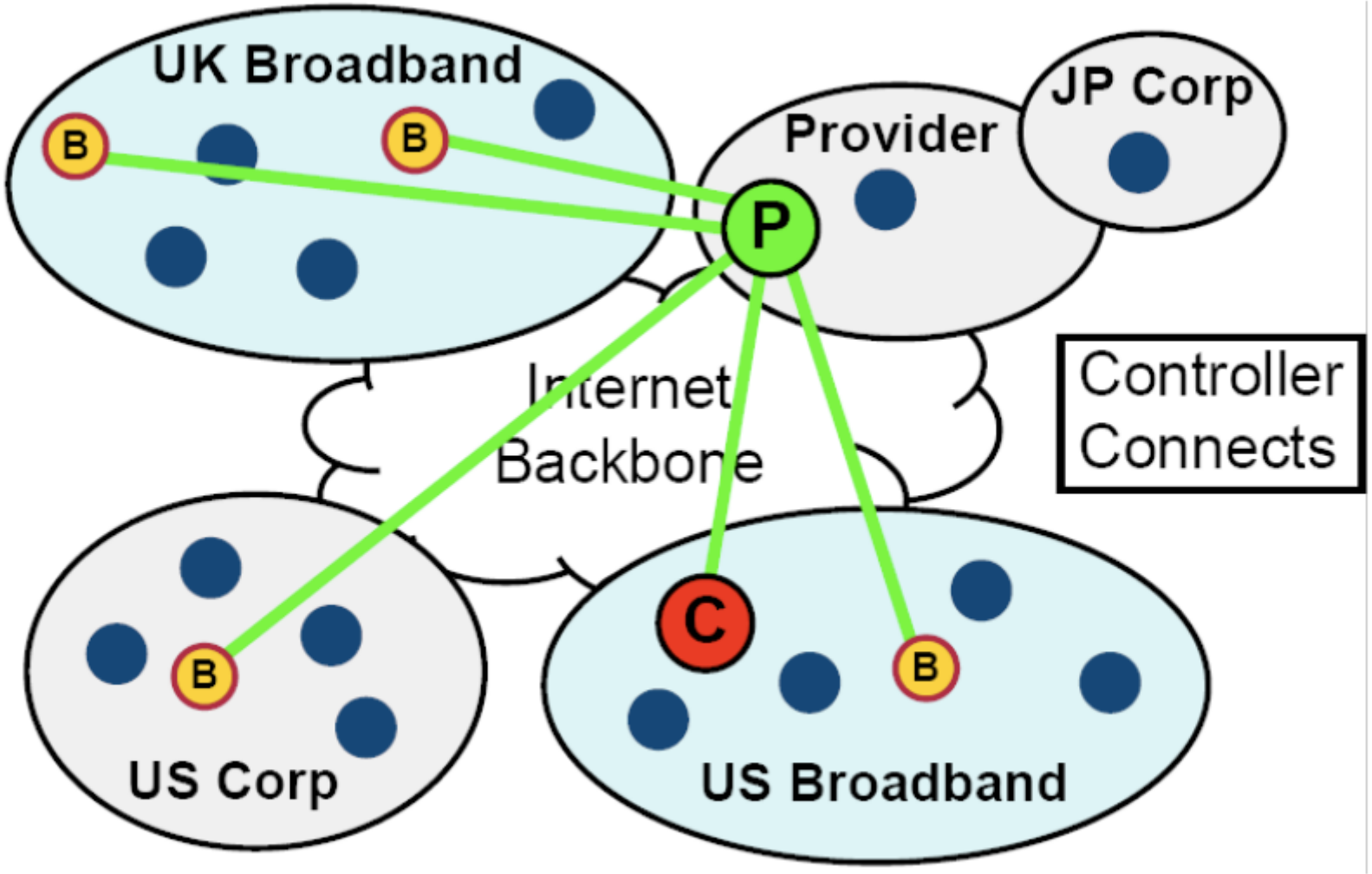
Botnet creation (2/5)



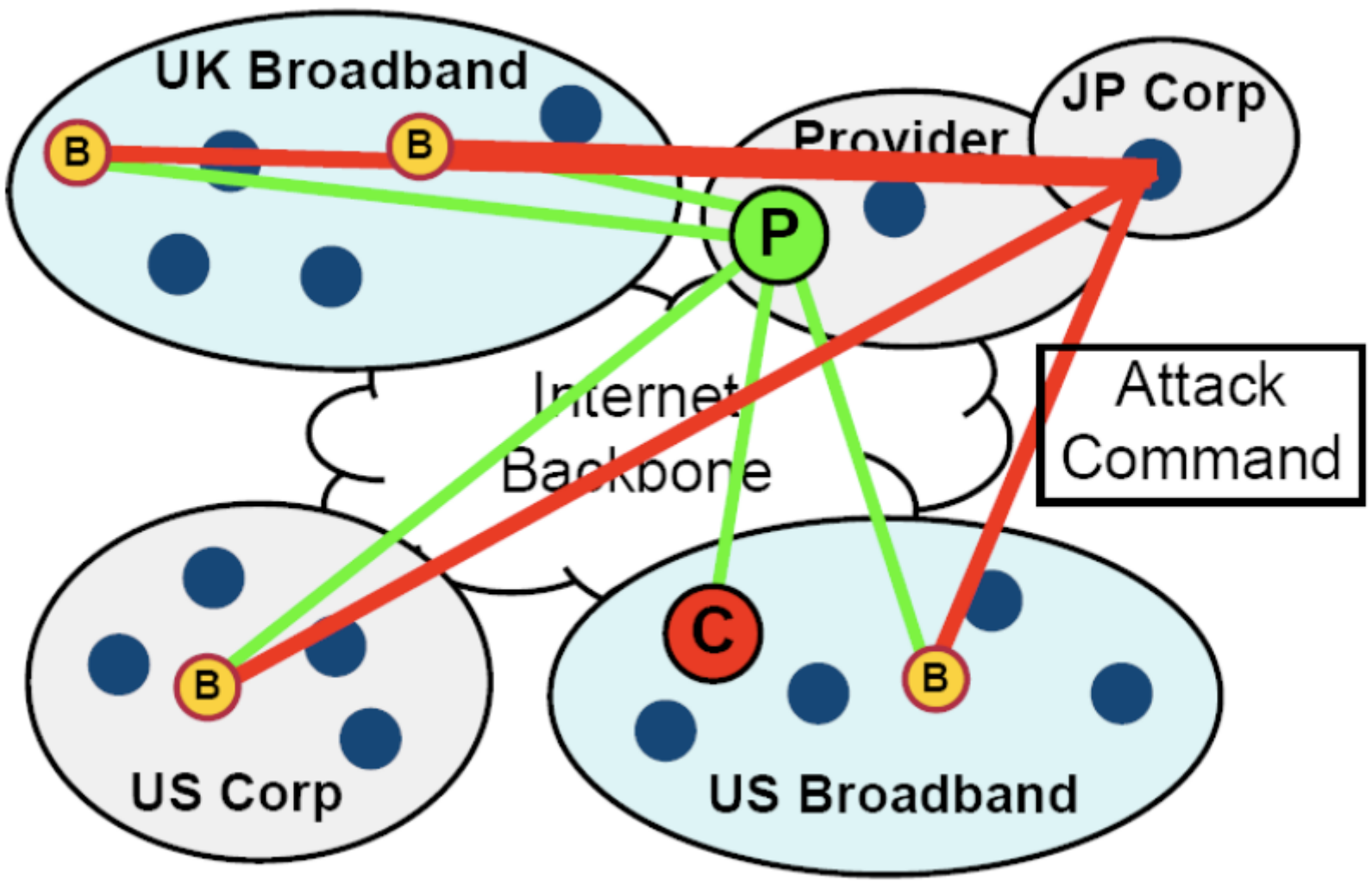
Botnet creation (3/5)



Botnet creation (4/5)



Botnet creation (5/5)



Botnet Propagation – Hiring of new bots

- Email
  - Requires user interaction, social engineering
  - Easiest method; common.
- instant message
  - Various: social eng., file transfer, vulnerabilities
- remote software vulnerability
  - Often, no interaction needed

Botnet Propagation – Hiring of new bots

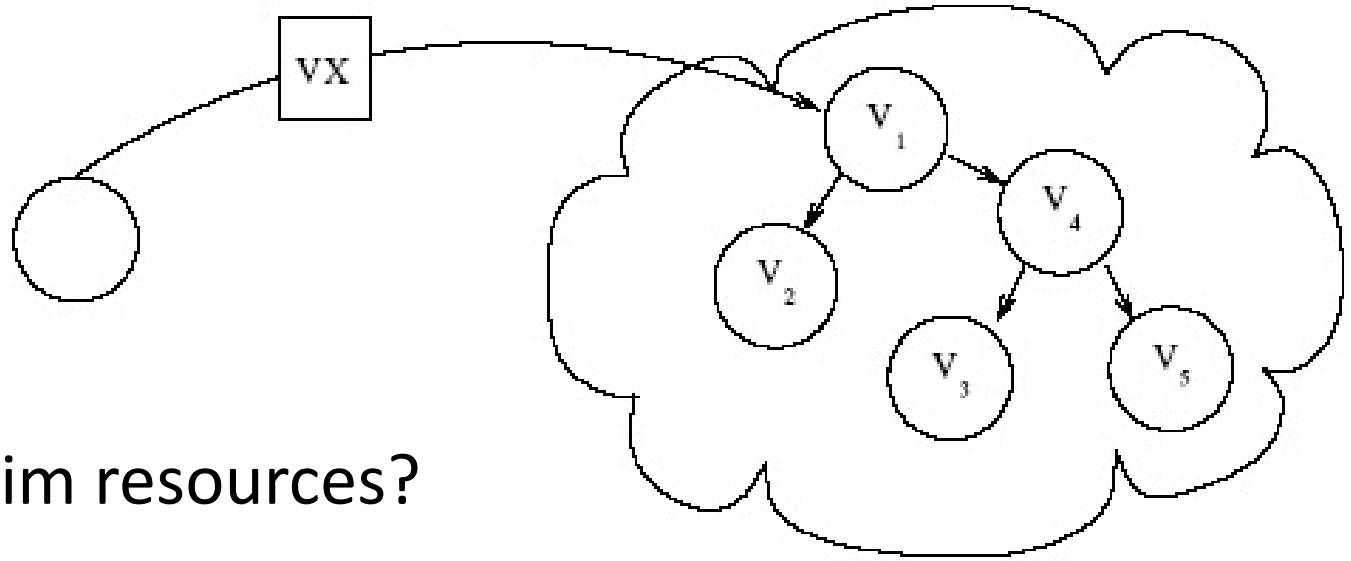
- "seed" botnets
  - Botnets create botnets.
  - Used for upgrades.

Attacker Challenges

- How to rally victims
- Most (> 90%) use DNS

The Rallying Problem

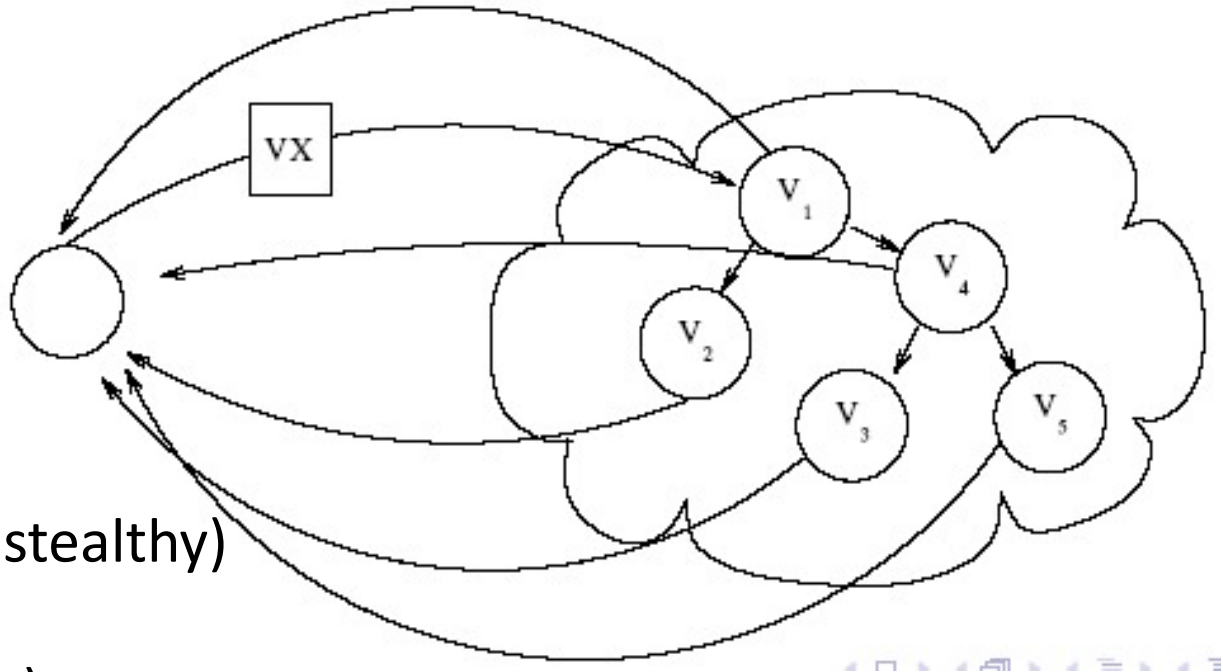
- Suppose we create virus
  - Download vx code; compile
  - Uses email propagation/social engr.
- We mail it...



- What if we want to use victim resources?

Rallying - I

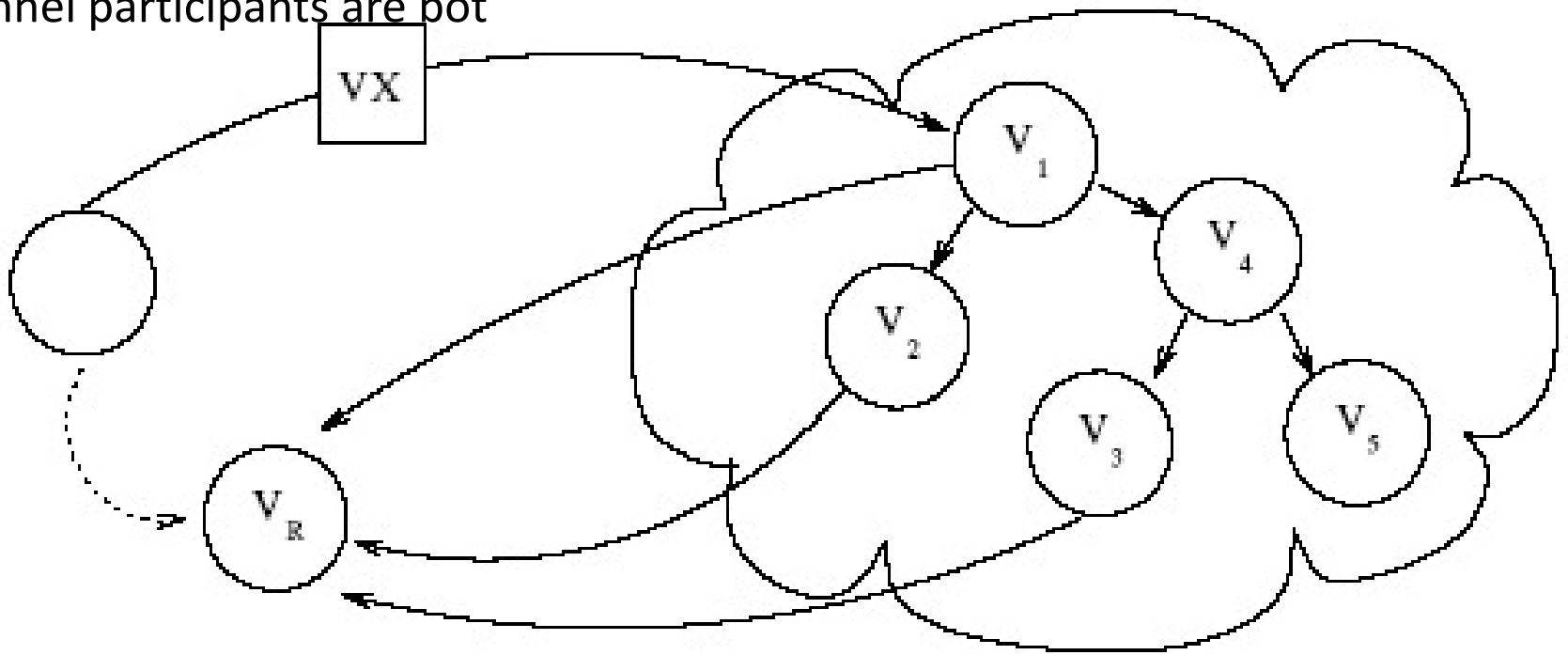
- Naively, we could have victims contact author...



- Problems
  - VX must include author's address (not stealthy)
  - Single rallying point (not robust)
  - VX has hard-coded address (not mobile)
    - Can not change

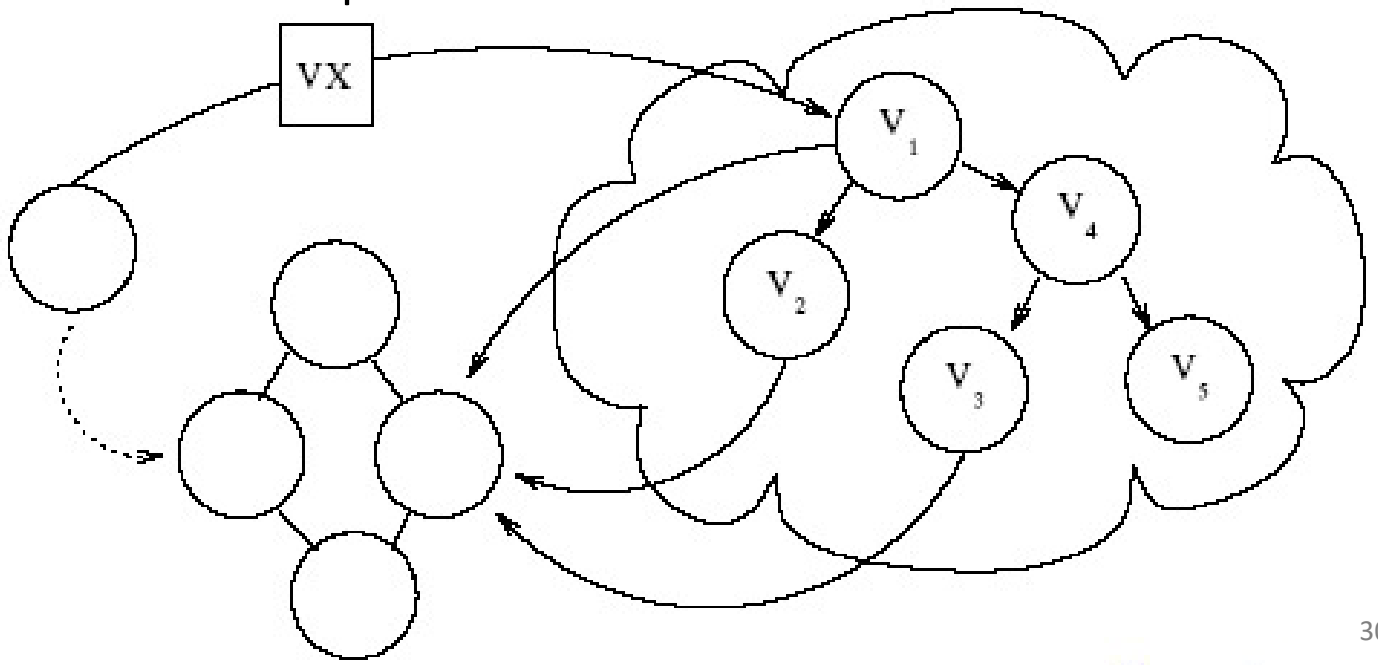
Rallying - II

- The victims could contact a 3rd party, e.g., post to Usenet
  - Some connections dropped, single point of failure (not robust)
  - Rival VXers and AVers obtain list (not stealthy)
    - Public, lasting record of victims (not stealthy)
    - All 3rd party cannel participants are bot



Rallying - III

- The victims could contact a robust service, e.g., IRCd
  - Portability of IRCd DNS (is mobile)
  - No single point of failure (is robust)
  - Rival VXers and AVers id list (not stealthy)
    - Addressed by adjusting protocol adherence or private nature of service.



Rallying – Summary

- A first task of zombies is rallying
  - how can victims contact the master safely?
- Simple, naïve approach:
  - Victims contact single IP, website, ping a server, etc.
  - Easily defeated (ISP intervention, blackhole routing, etc.)
  - Still used by kiddies, first-time malware authors
- Resilient Networks needed
- Open Problem
  - If you had 300K+ bots, what does command and control look like?
  - Botnets usually use ~3,000 users/channel
  - Newer botnets use command and control hierarchy, with botmaster, lieutenants, and individual zombies

Detecting Bots

- Prevent systems from getting infected
- Directly detect bot communications
  - communication between bots and bot controllers
  - e.g. IRC botnets
    - IRC ports (e.g., TCP 6667)
    - Monitor IRC payload for known commands

Detecting Botnet Activity

- Many bots are controlled via IRC and DNS
  - IRC used to issue commands to zombies
  - DNS used by zombies to find the master
- IRC/DNS activity is very visible in the network
- Easily evaded by using encryption and P2P ☹

Detecting Bots (con't)

- Check behavioral characteristics
  - e.g. IRC clients responding very quick may be bots
    - Use Netflow to capture the traffic
- Track the botnet by honeypot
  - Use honeypot to get infected
  - Make new bot and join botnet
    - Create signature

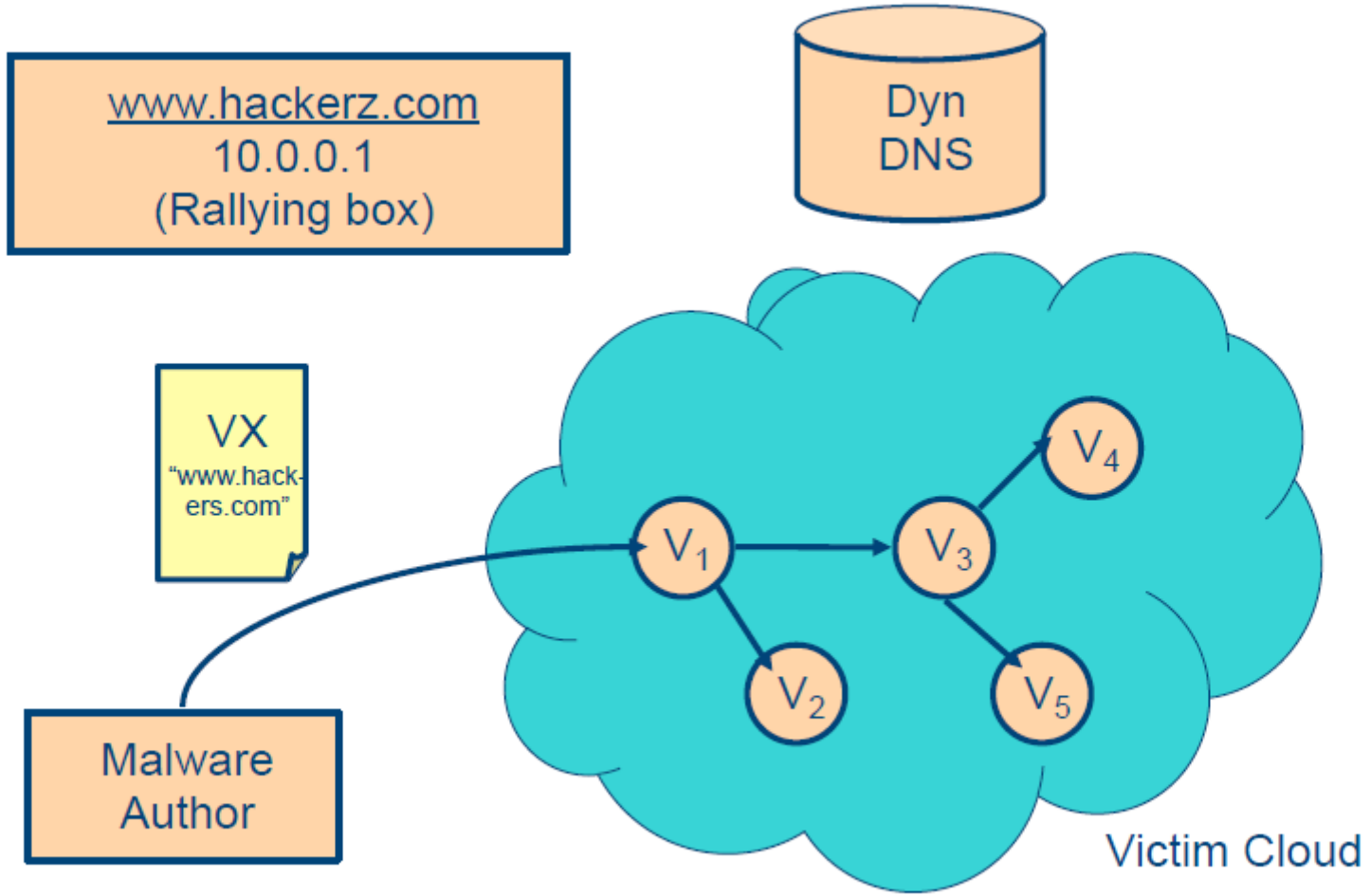
Removal Example

- So, you find a bot army big enough to DDoS cnn.com or similar sites. What now?
- Proceed with caution.
- Bot is reverse engineered
- Always approach channel from the IRC server, or from a proxied address.
- "Remove self" command issued
  - Most bots have such a command, to help evade forensic analysis
  - Locate, and send command, spoofed from the bot master's address.

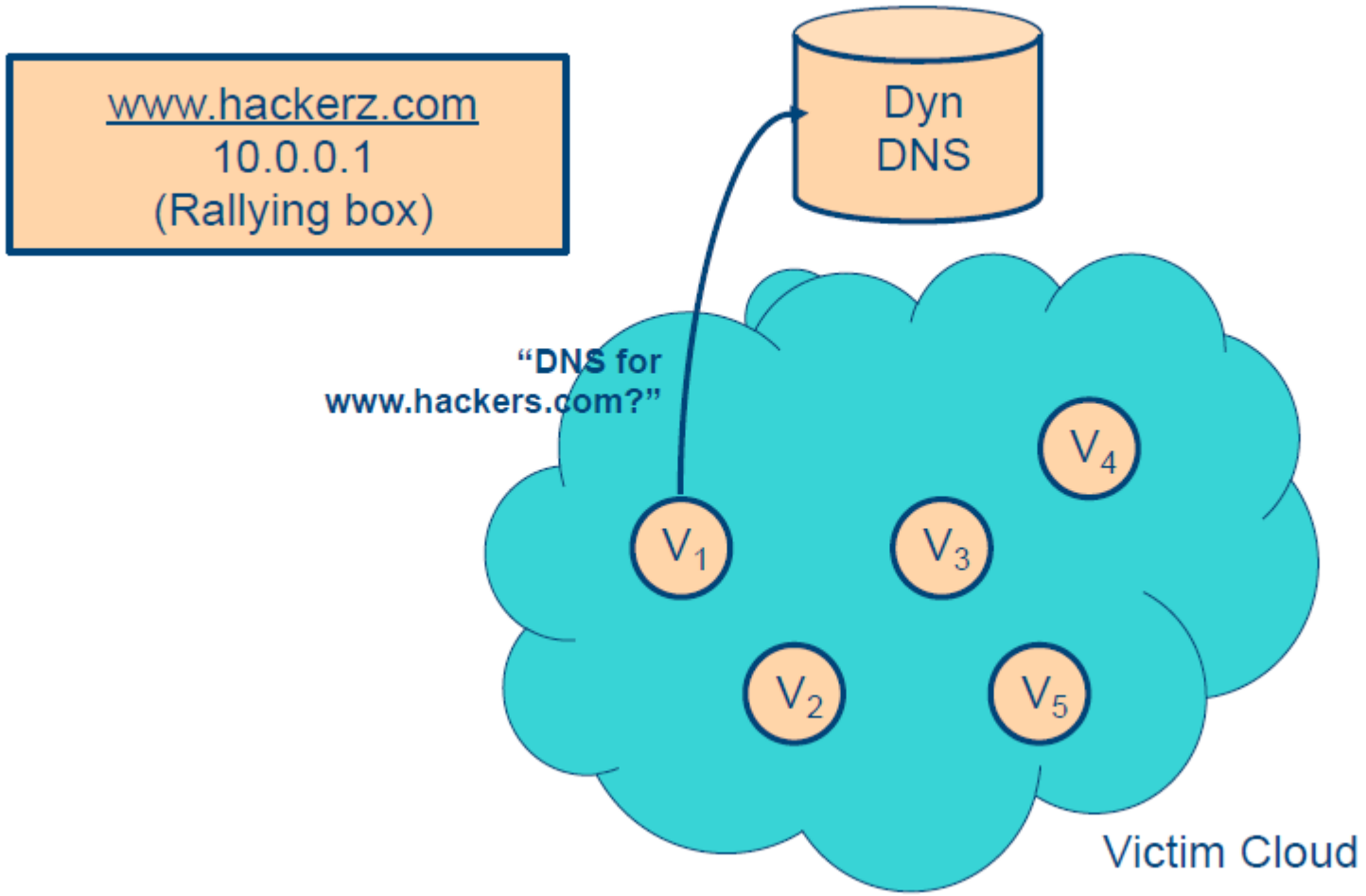
KarstNet: Responding to Botnets

- KarstNet approach
  - Manipulate the DNS for drone armies
  - Almost all malware rallies through use of DynDNS
  - Therefore, have DynDNS provider make a sinkhole Record Response (RR) for the CNAME.

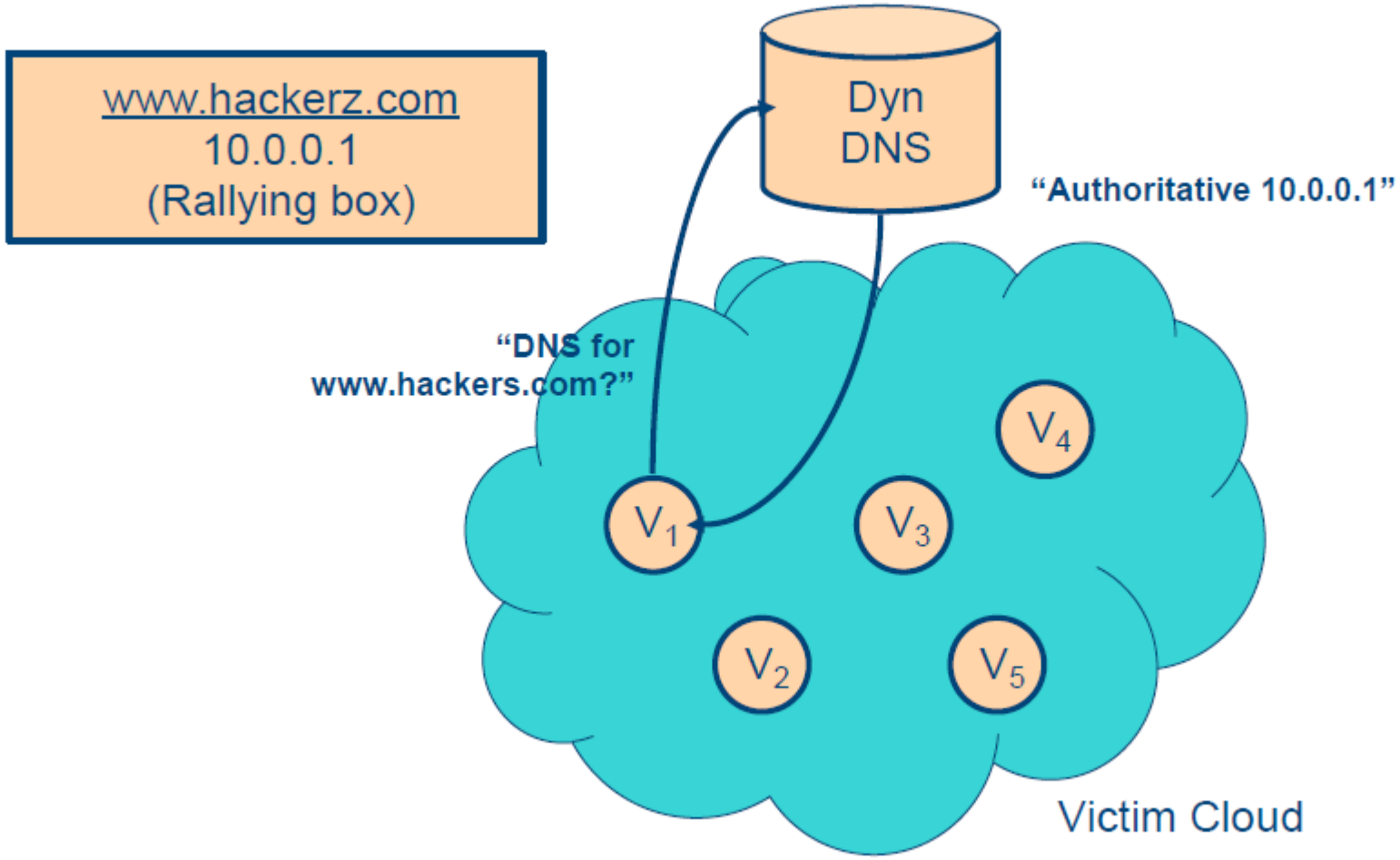
KarstNet: Malware with Strings



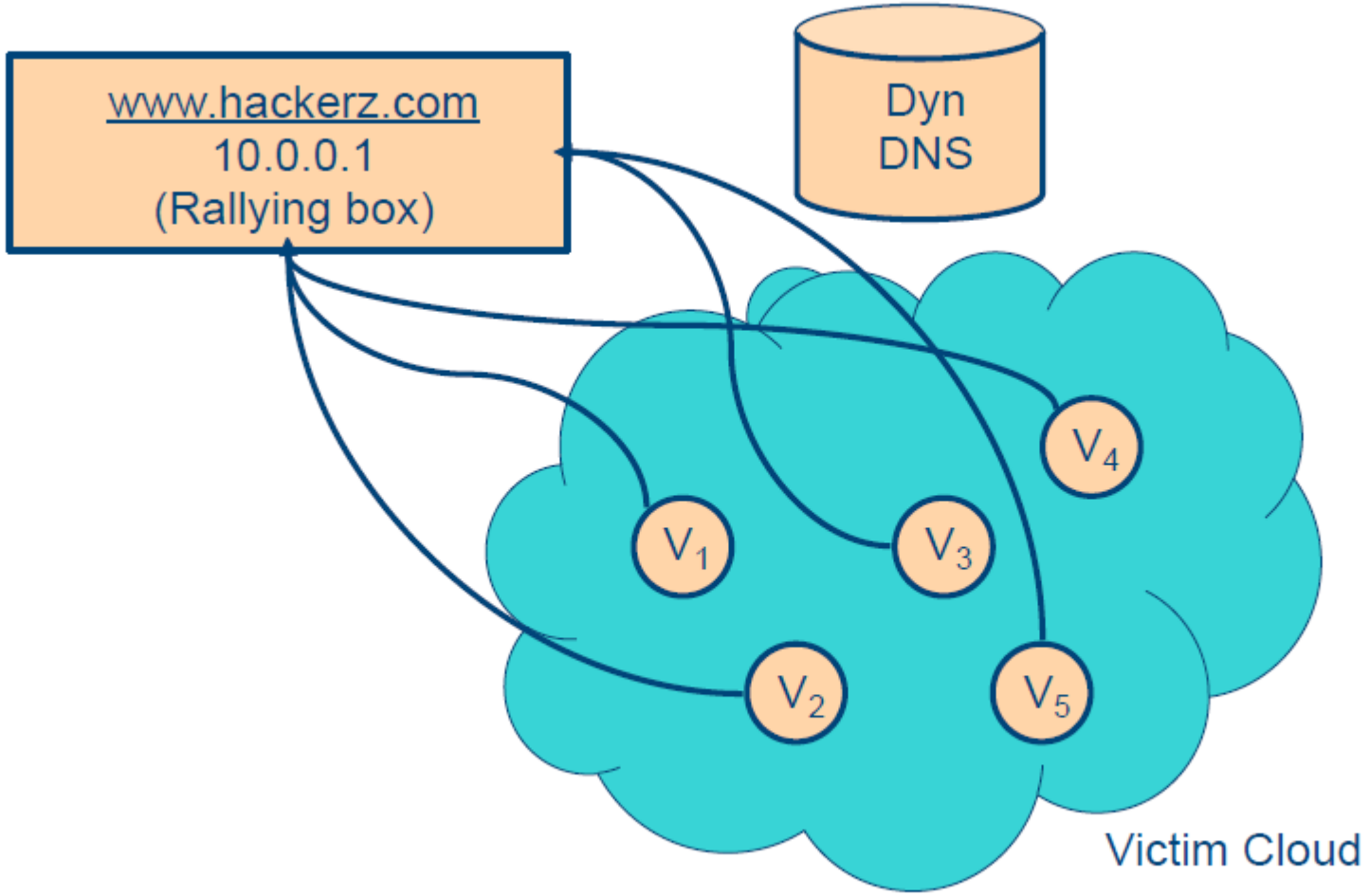
KarstNet: A-record Rallying



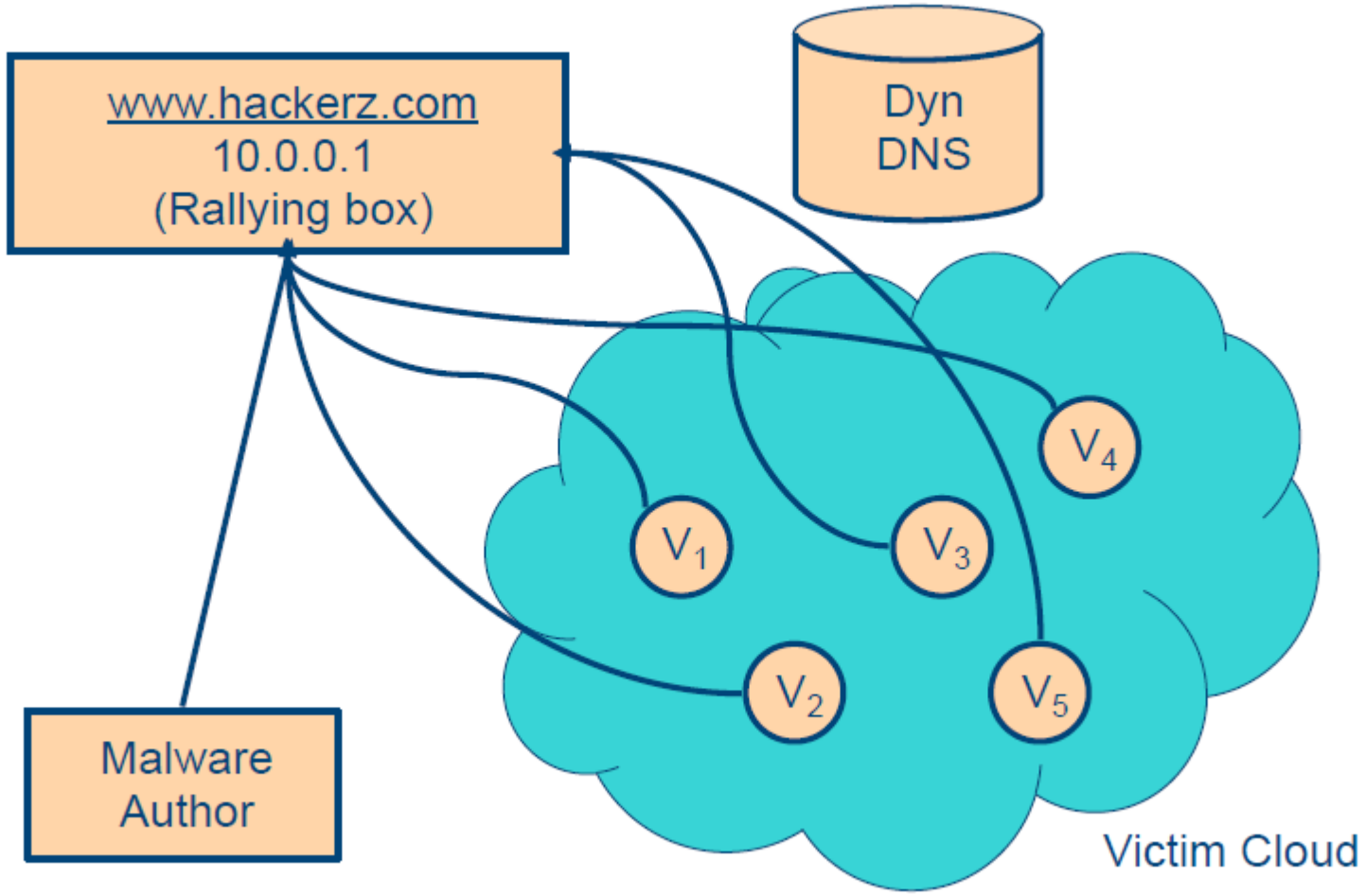
KarstNet: A-record Rallying



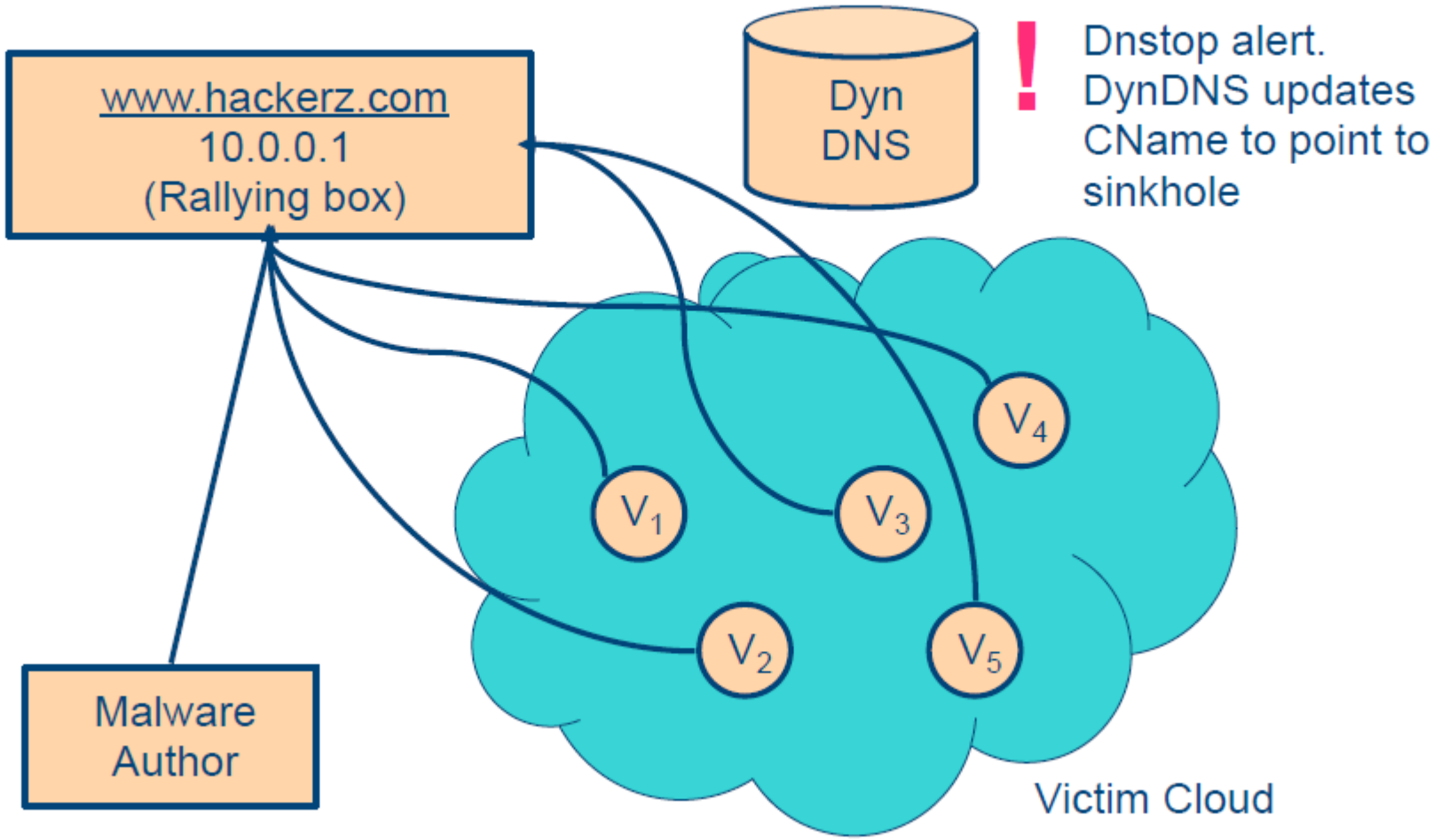
KarstNet: Command and Control



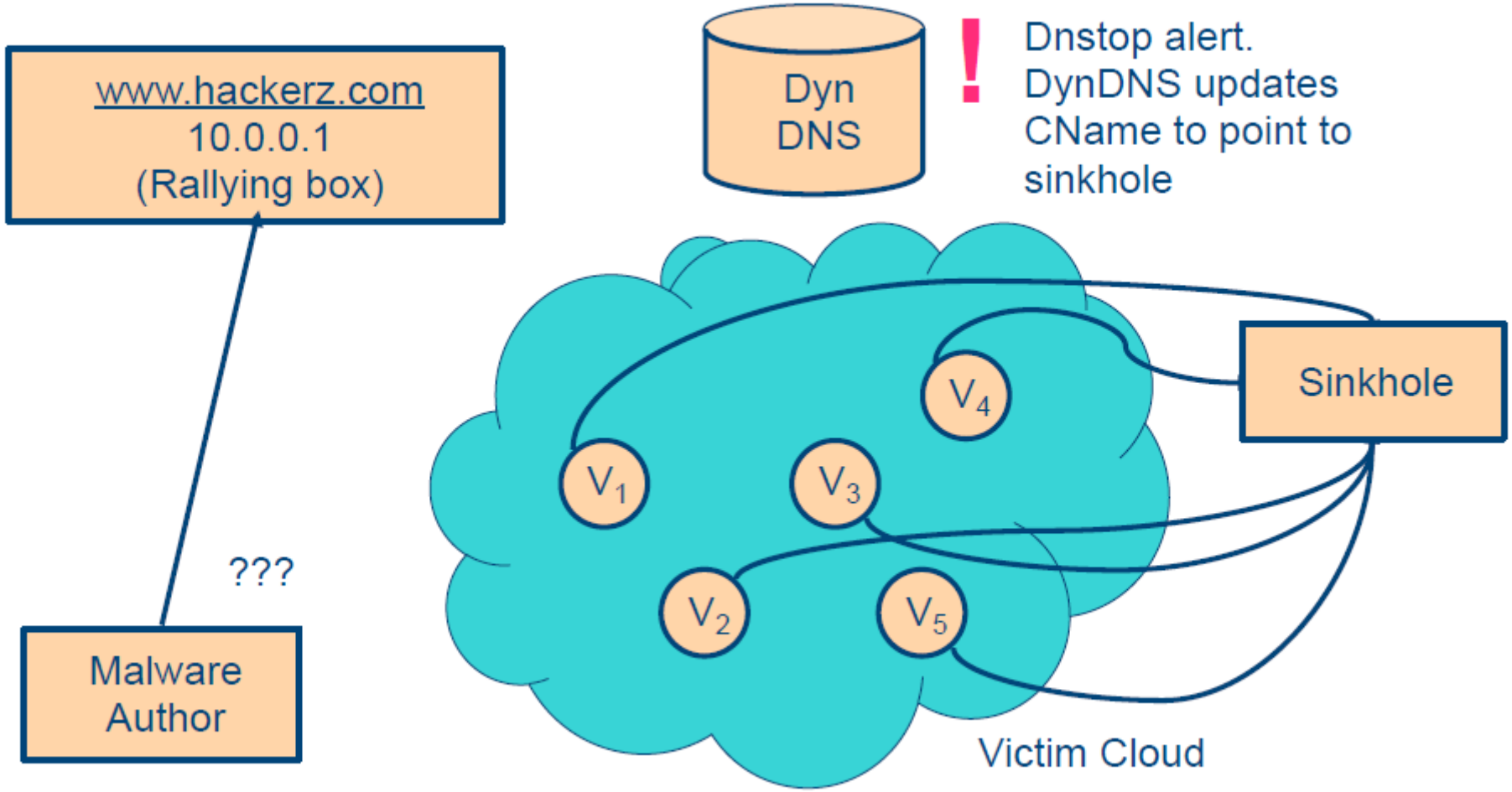
KarstNet: Command and Control



KarstNet: Detection



Drone Army Responses: DNS



Monitoring to Detect C&C Domains and Bots

- Detect abnormal patterns/growth of "popularity" of a domain name
  - Identify botnet C&C domain and bots
- Botnet-related domains usually contain random-looking (sub)strings
  - Many/most sensible domain names have been registered (for legitimate use)
  - In particular, botnet domain name often looks completely random, and the domain name tends to be very long (users can't type but bots don't type!)
  - E.g. wbghid.1dumb.com, 00b24yqc.ac84562.com
- Monitor for "new and suspicious" domain names that enjoy exponential or linear growth of interests/look-ups

Conclusions

- Botnets are the biggest Internet threat of the current generation
  - Source of many attacks
- Detection and containment can be successful only at the network level
  - Detection should be ideally before the attack

Acknowledgments/References

- [Singh] CS 6262 , Kapil Kumar Singh, Georgia Institute of Technology, Fall 2007.
- [Shmatikov] CS 378 - Network Security and Privacy, Vitaly Shmatikov, University of Texas at Austin, Fall 2007.
- [Raftopoulos] HY558, Elias Raftopoulos, Department of Computer Science, University of Crete, August 2008. (http://www.csd.uoc.gr/~hy558/reports/eraftop_zombie_roundup.ppt)
- Research in Botnet Detection and Malware Analysis, Wenke Lee, College of Computing, Georgia Institute of Technology