CE 817 - Advanced Network Security
Botnets

Lecture 11

Mehdi Kharrazi
Department of Computer Engineering
Sharif University of Technology

Acknowledgments: Some of the slides are fully or partially obtained from other sources. Reference is noted on the bottom of each slide, when the content is fully obtained from another source. Otherwise a full list of references is provided on the last slide.

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)

[Shmatikov]

Botnets as a Root cause

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

- … many others

Botnets – Money matters !

- CPM
- 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) Redux

- 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
  - Ping of death: ICMP packets with payloads greater than 64K crash older versions of Windows
  - SYN flood: "open TCP connection" request from a spoofed address
  - UDP flood: exhaust bandwidth by sending thousands of bogus UDP packets

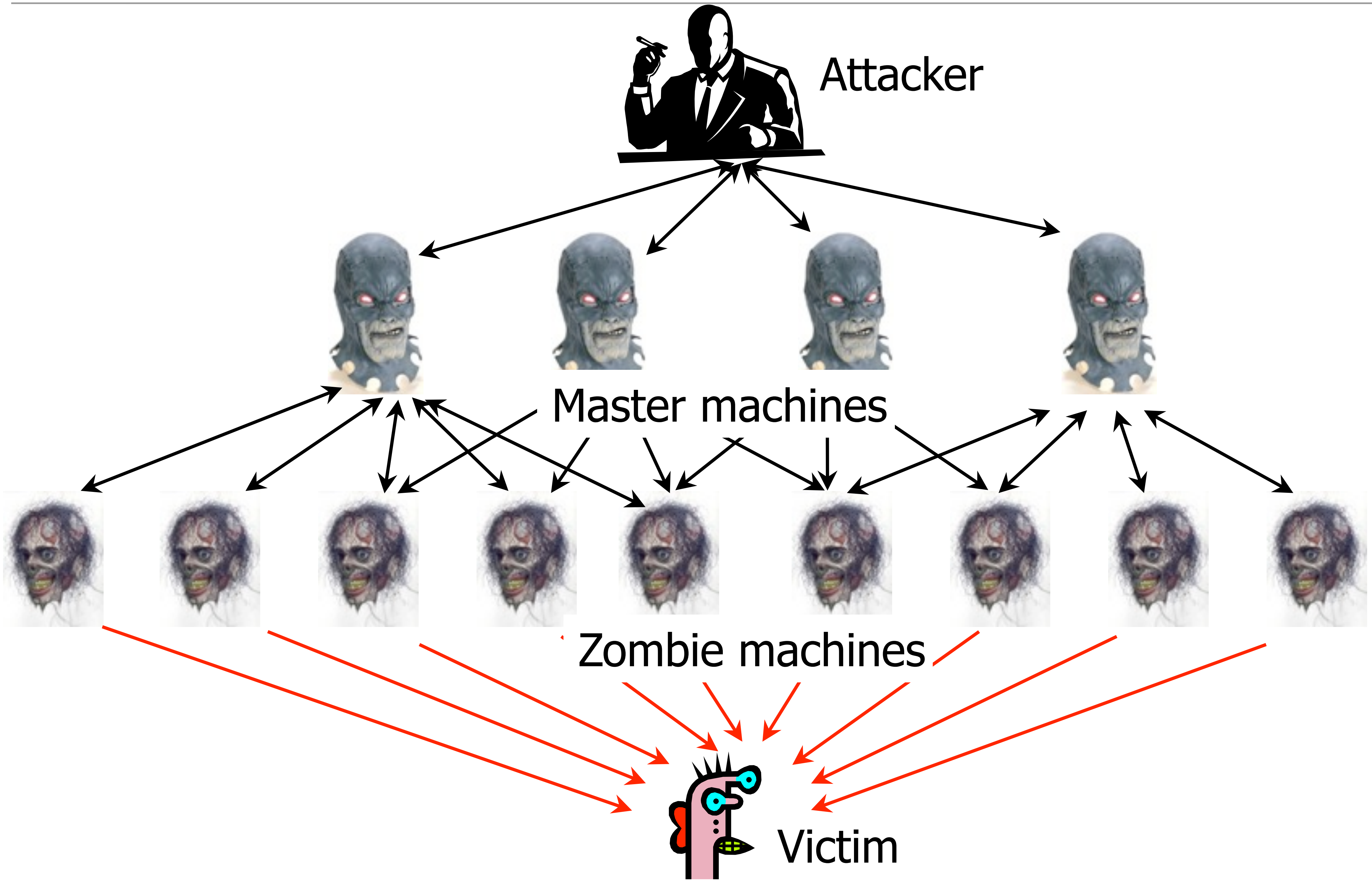
[Shmatikov]

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

[Shmatikov]

DDoS Architecture



Spring 1391

[Shmatikov]

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

[Shmatikov]

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

[Shmatikov]

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

[Shmatikov]

Agobot

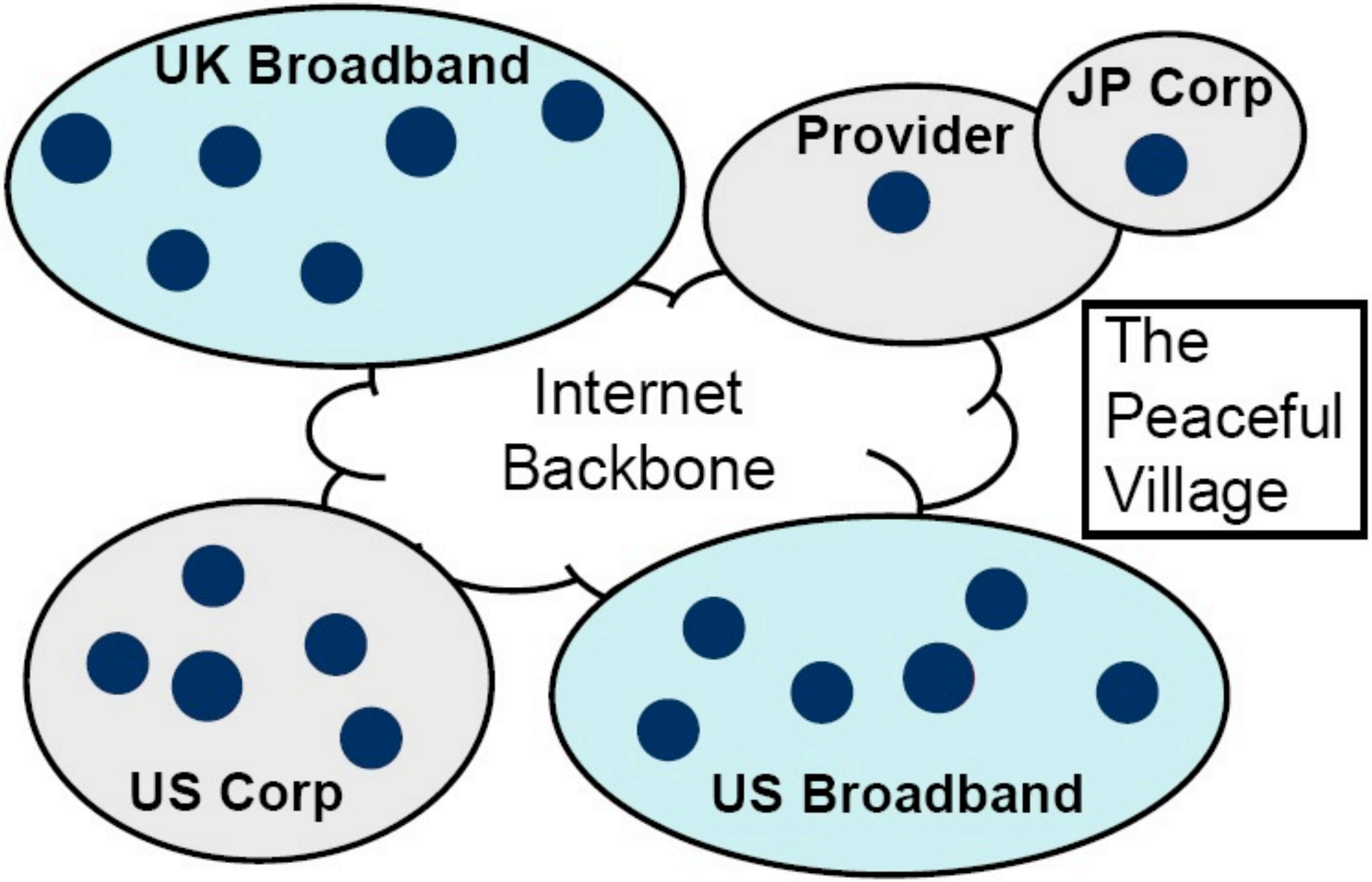
- 20,000 lines of C/C++ code
- IRC-based command and control
- Scanning tools, many propagation vectors
- Capable of many DoS flooding types
- Code obfuscation to avoid detection
- Installs sniffer, terminates anti-virus processes, points DNS for anti-virus to localhost

Other Modern Bots

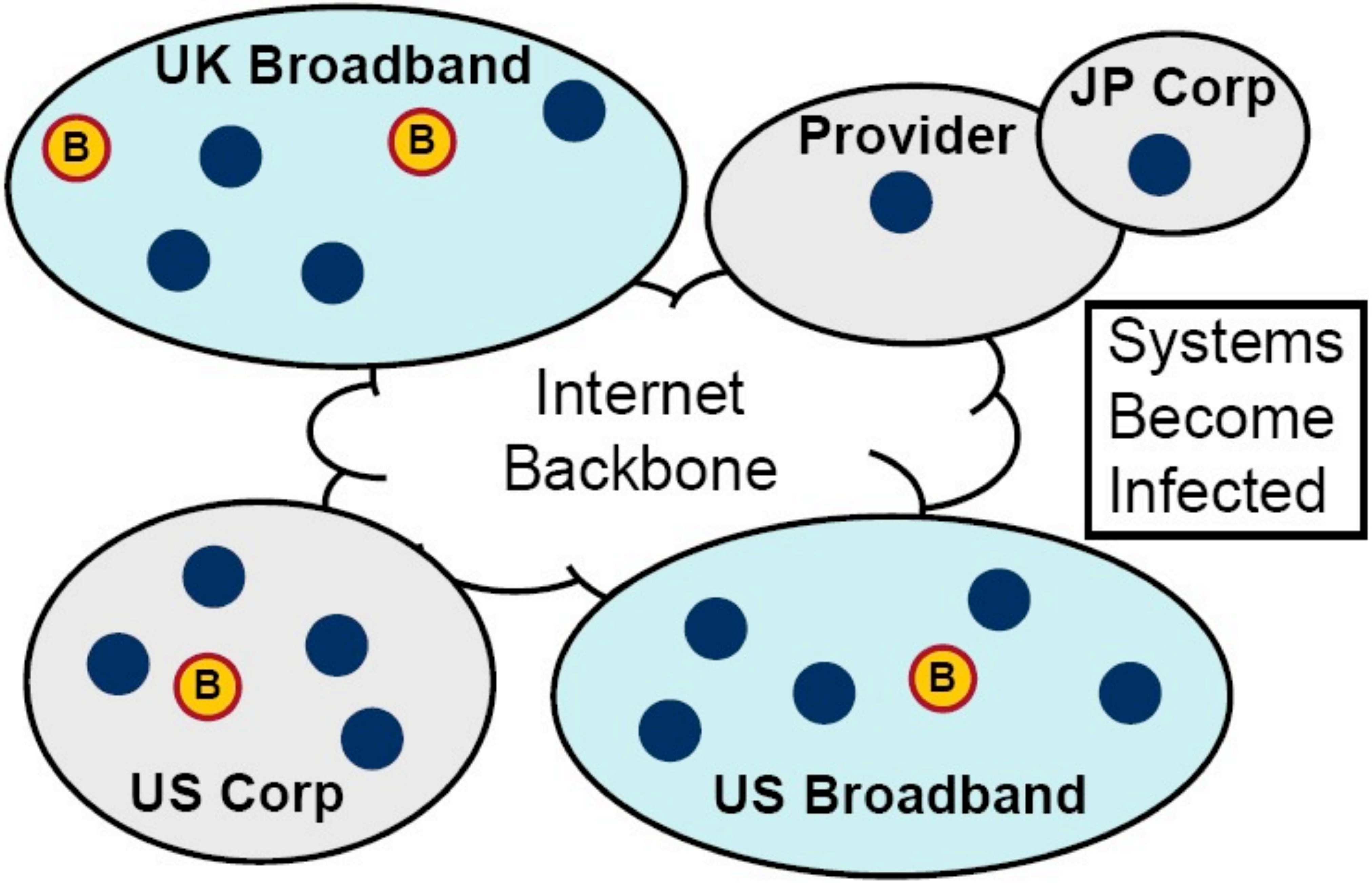
- SDBot / SpyBot
  - Non-malicious, but can be extended for scanning, sniffing, DoS attacks
- GT-Bot
  - Renamed mIRC
  - Scanning, DoS, RPC and NetBIOS exploits
  - Simpler than Agobot
    - 2-3,000 lines of C code
  - Extensible and customizable codebase
- Trend: hybrids of bots, trojans, worms

[Shmatikov]

Botnet creation (1/5)



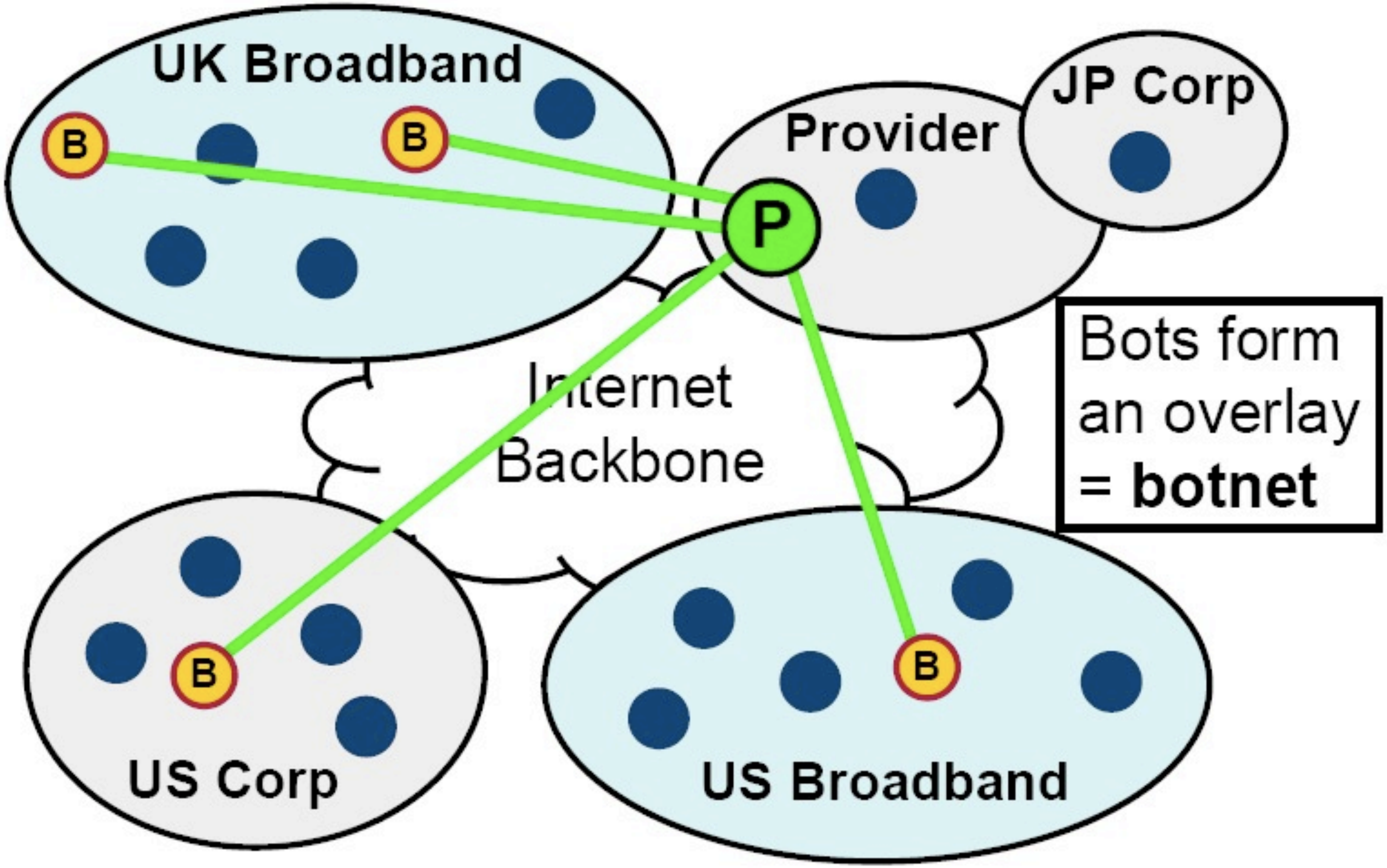
Botnet creation (2/5)



Systems Become Infected

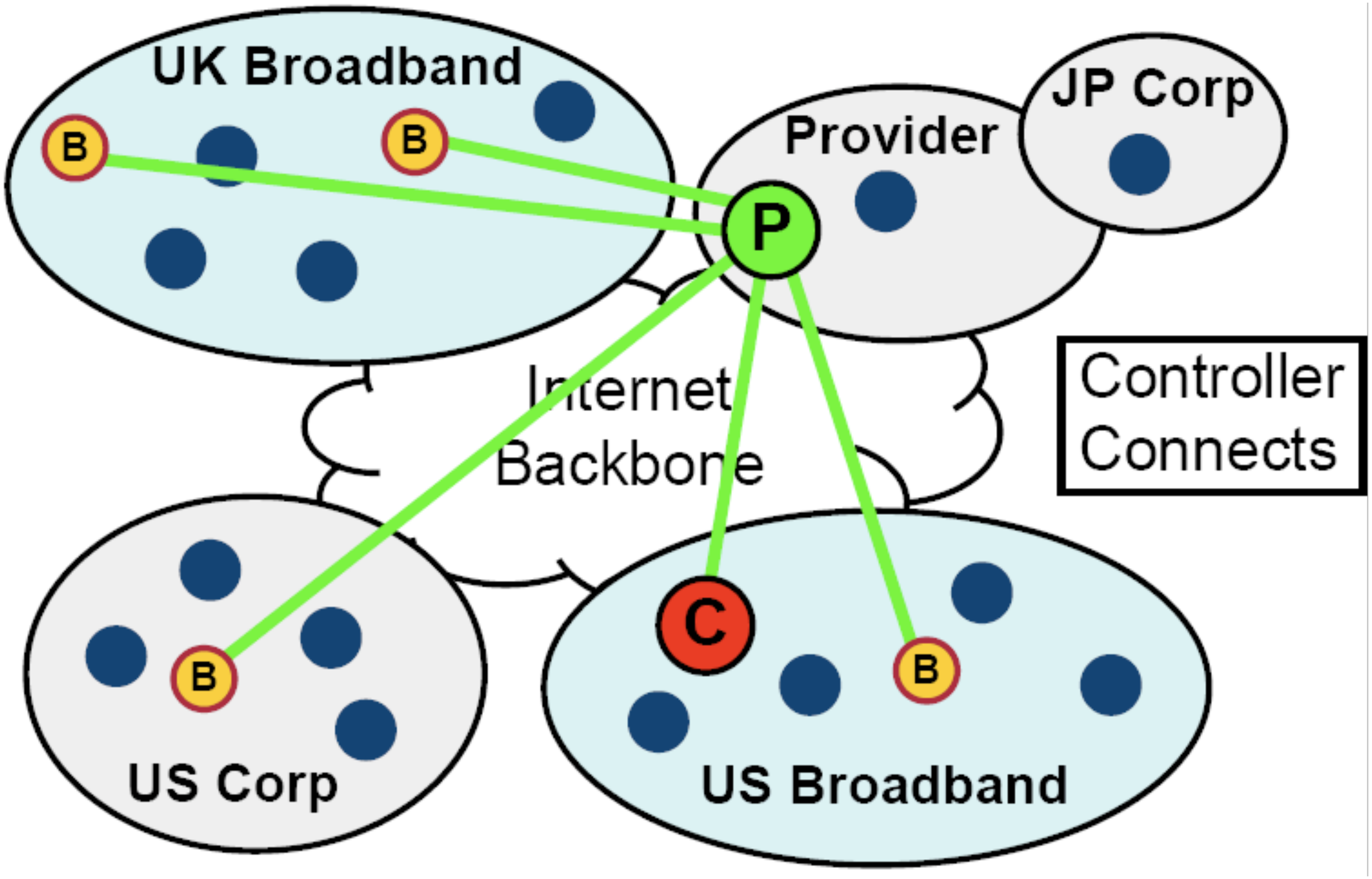
[Raftopoulos]

Botnet creation (3/5)



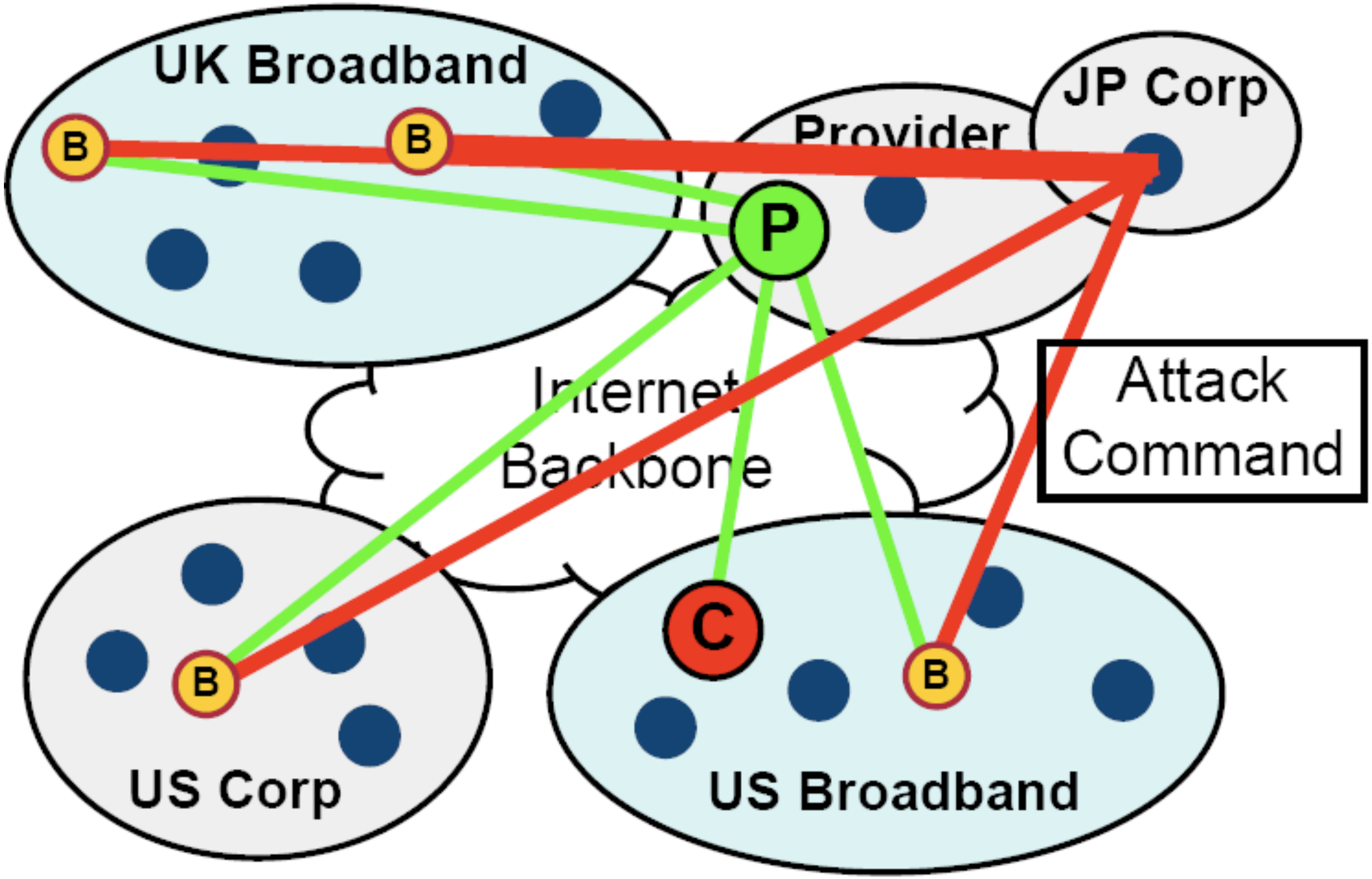
Bots form an overlay = **botnet**

Botnet creation (4/5)



Controller
Connects

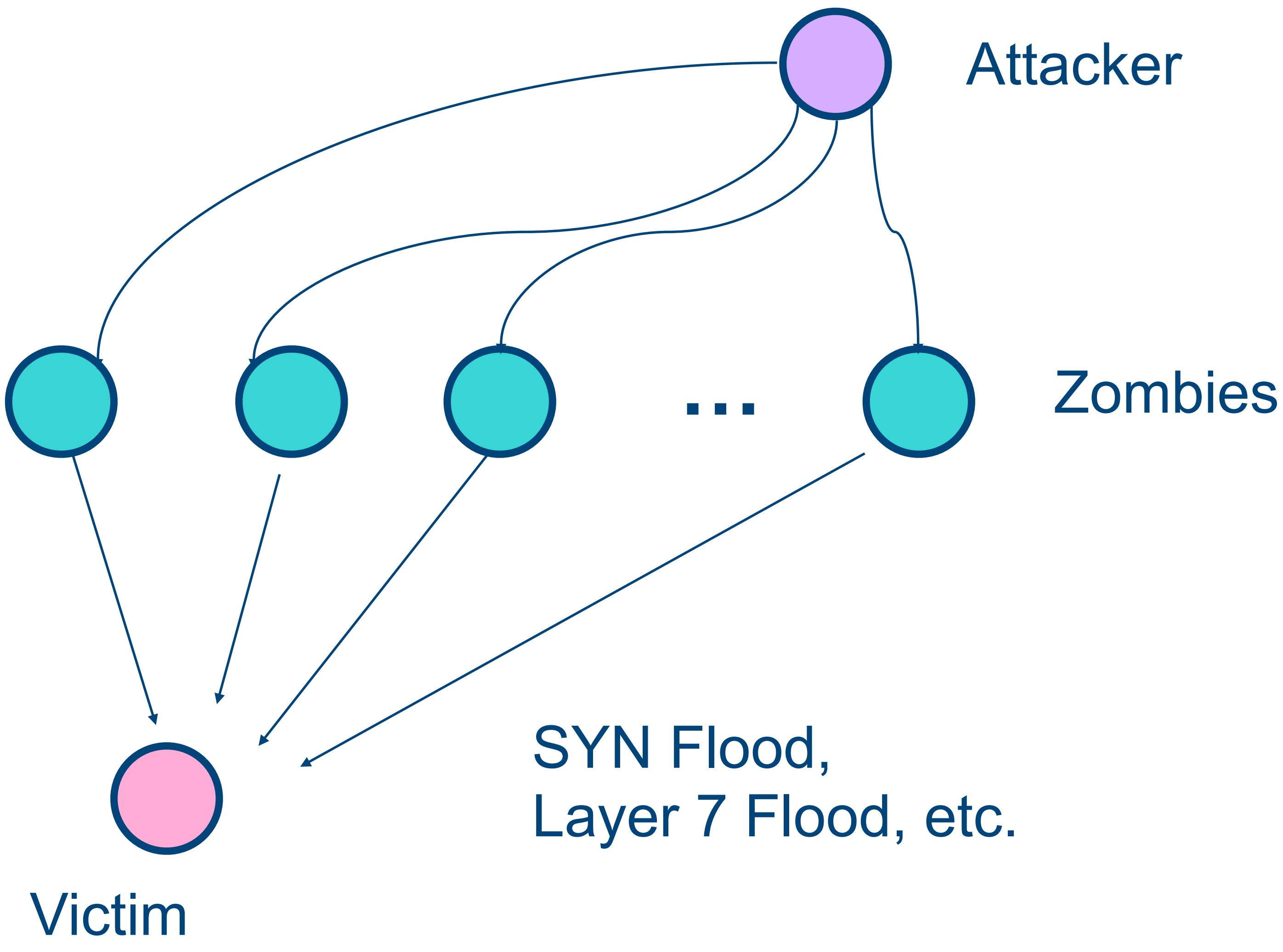
Botnet creation (5/5)



[Raftopoulos]

Attack Update

- Botnets of course are used for DDoS

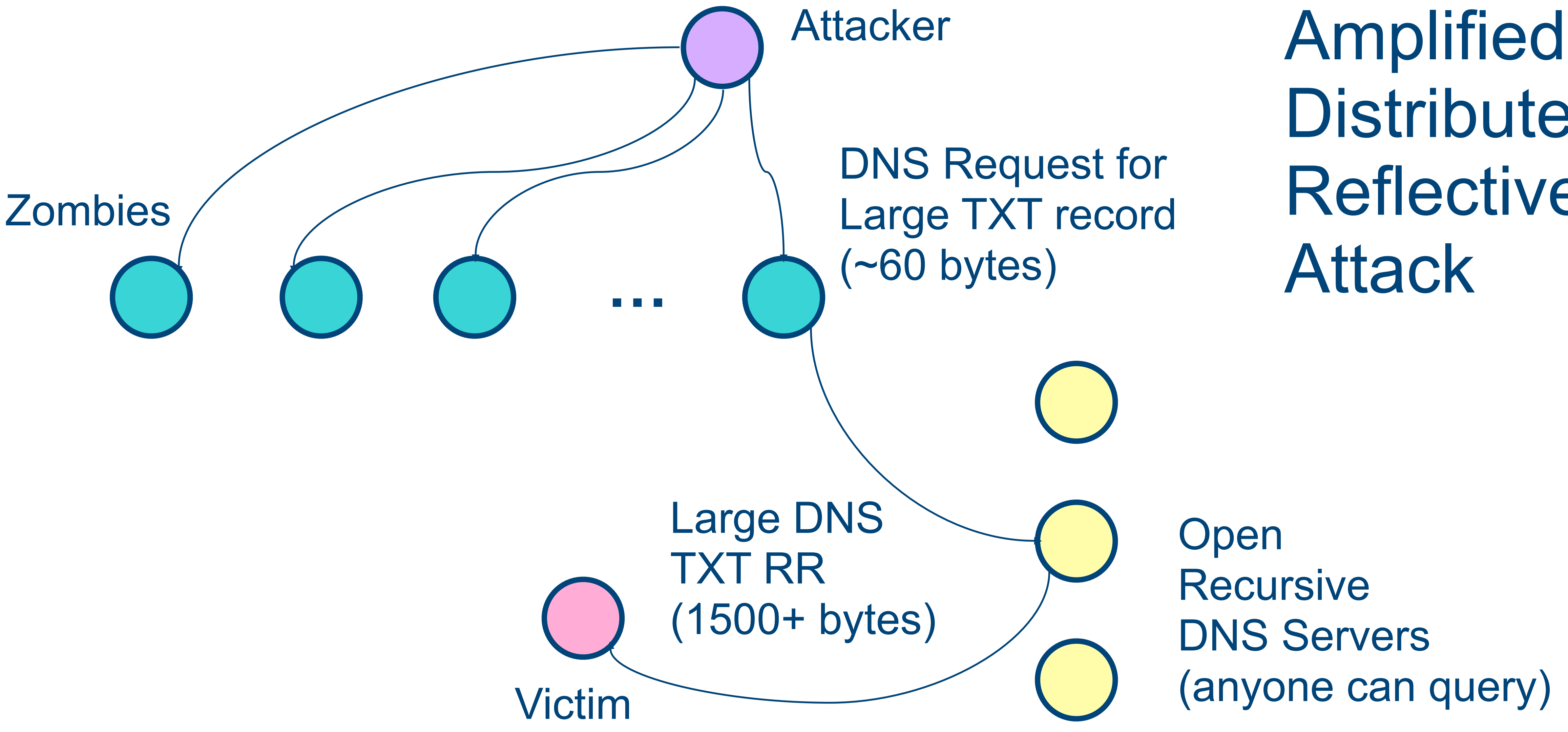


SYN Flood,
Layer 7 Flood, etc.

Typical Distributed Denial of Service (DDoS)

Attack Update

- Botnets increasingly used for amplified distributed reflective attacks



Amplified Distributed Reflective Attack

[Singh]

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

[Singh]

Botnet Propagation – Hiring of new bots

- “seed” botnets
  - Botnets create botnets.
  - Used for upgrades.

- More than 80% of the bots are unpatched windows machines!

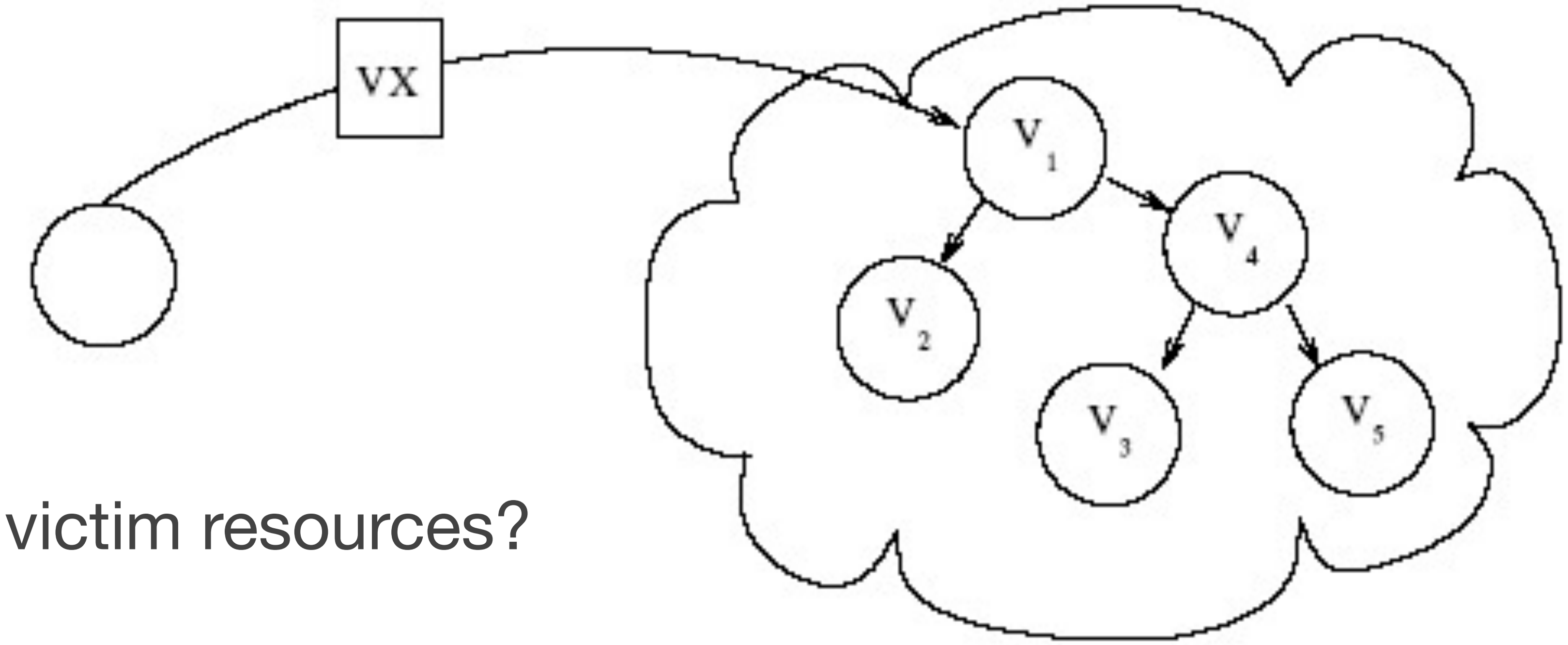
[Singh]

Attacker Challenges

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

The Rallying Problem

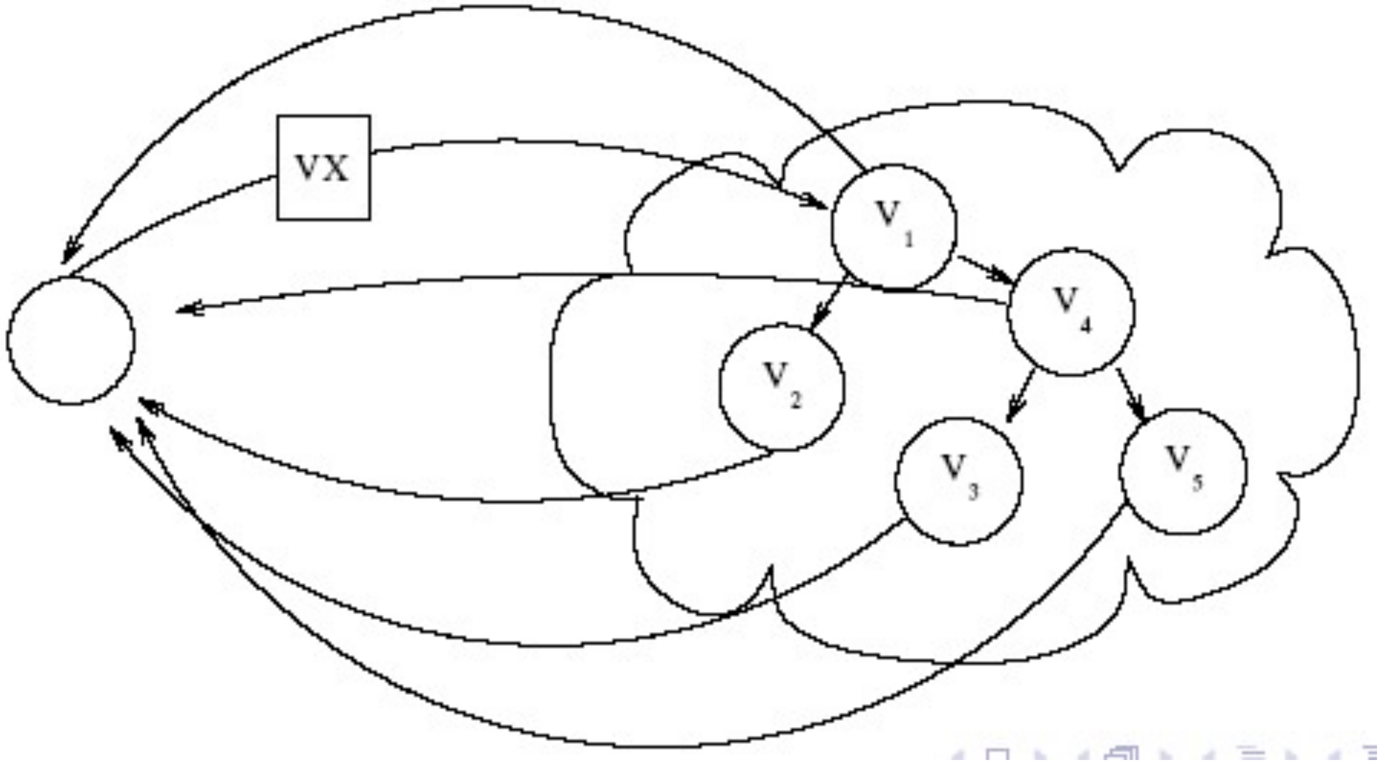
- Suppose we create virus
  - Download vx code; fiddle; compile
  - Uses email propagation/social engr.
- We mail it...



- What if we want to use victim resources?

Rallying - I

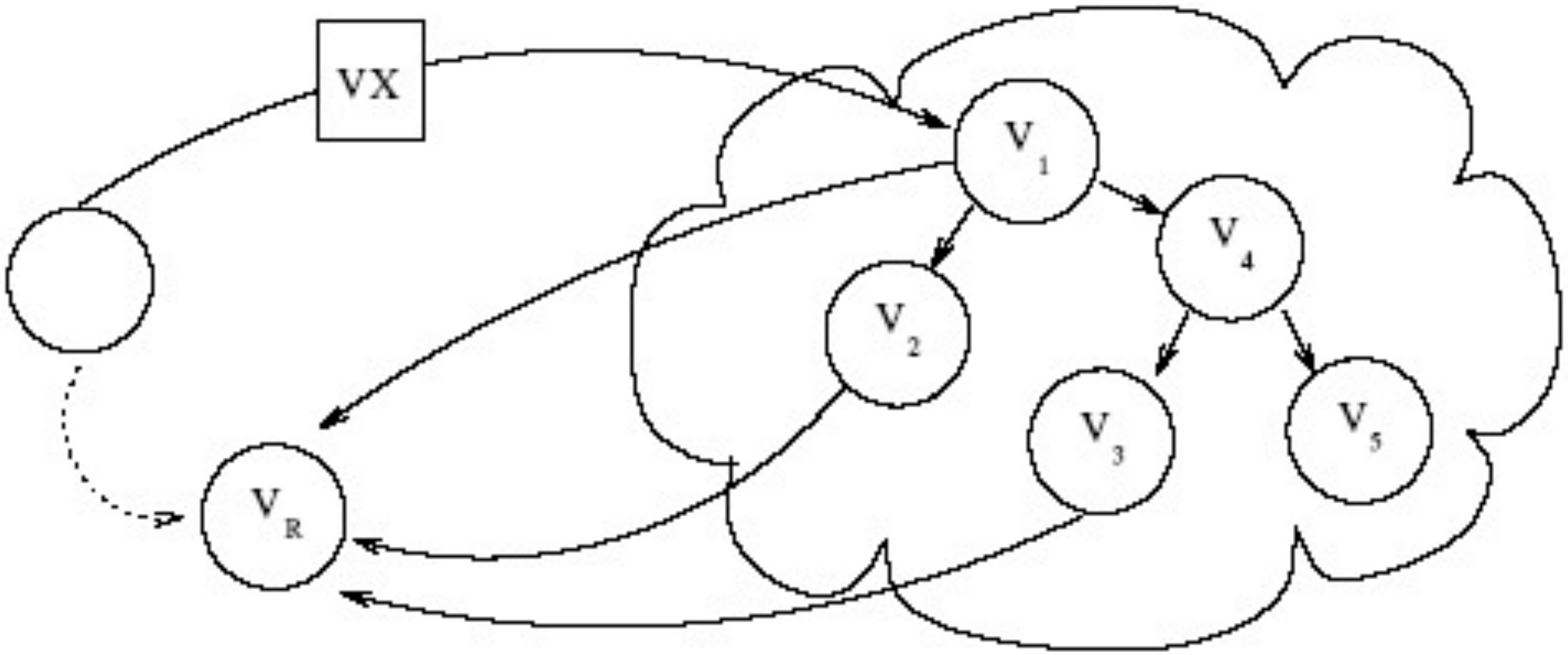
- Naively, we could have victims contact us...



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

Rallying - II

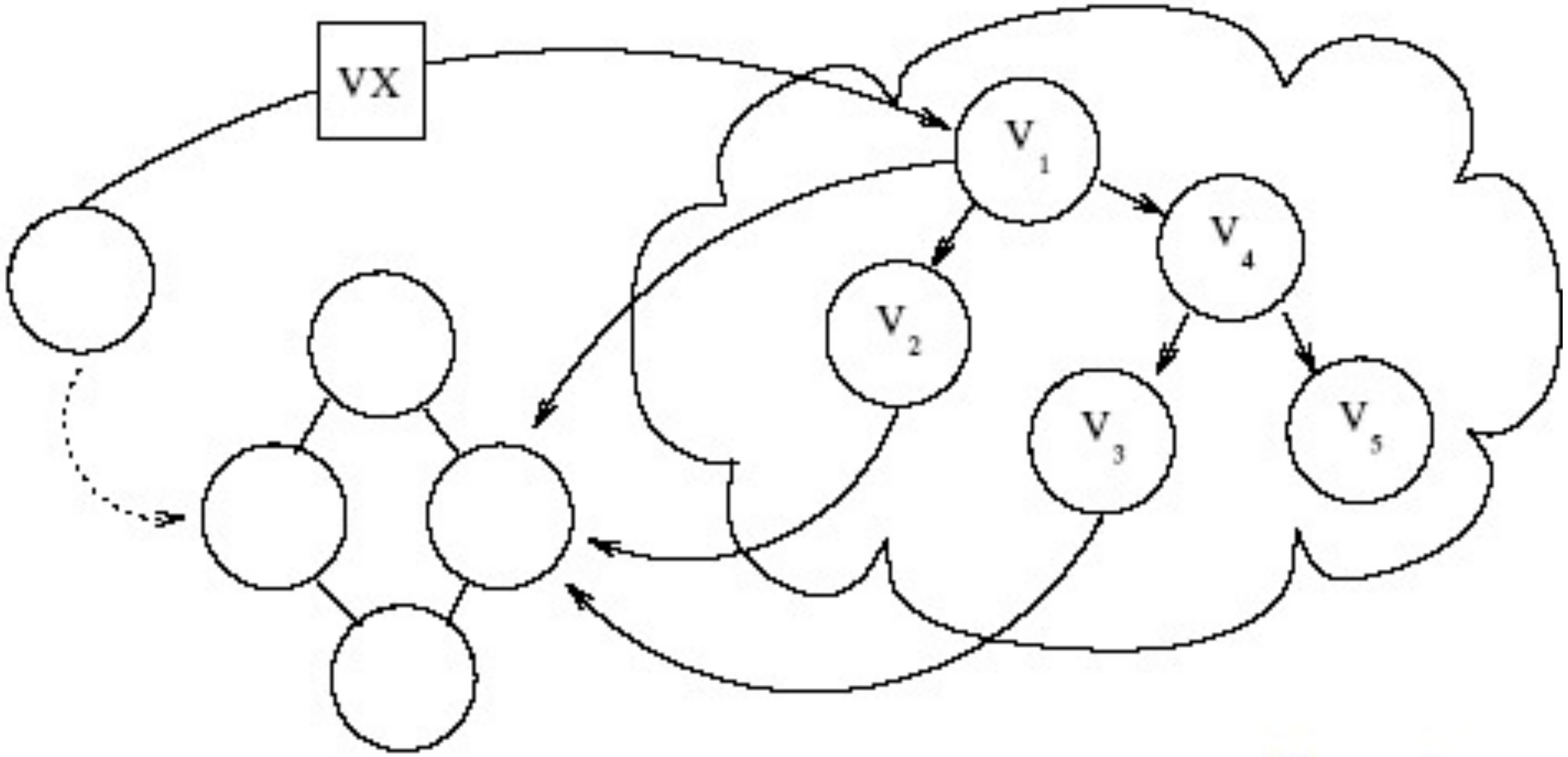
- The victims could contact a 3rd party, e.g., post to Usenet
  - Some connections dropped, single point of failure (not robust)
  - Rival VXers and AVers obtain list (not stealthy)
  - Public, lasting record of victims (not stealthy)



[Singh]

Rallying - III

- The victims could contact a robust service, e.g., IRCd
  - 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.
  - Portability of IRCd DNS (is mobile)



[Singh]

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

[Singh]

Bot/Botnet Measurements - Operators

- Very little hard data on botnets!
- Network operators (Tier-1 & Tier-2) actively fighting the problem:
- # of Botnets – increasing
- Bots per Botnet - decreasing
  - Used to be 80k-140k, now 1000s (evasion/economics?)
- More firepower:
  - Broadband (1Mbps Up) x 100s == OC3!!!

[Raftopoulos]

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

[Raftopoulos]

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

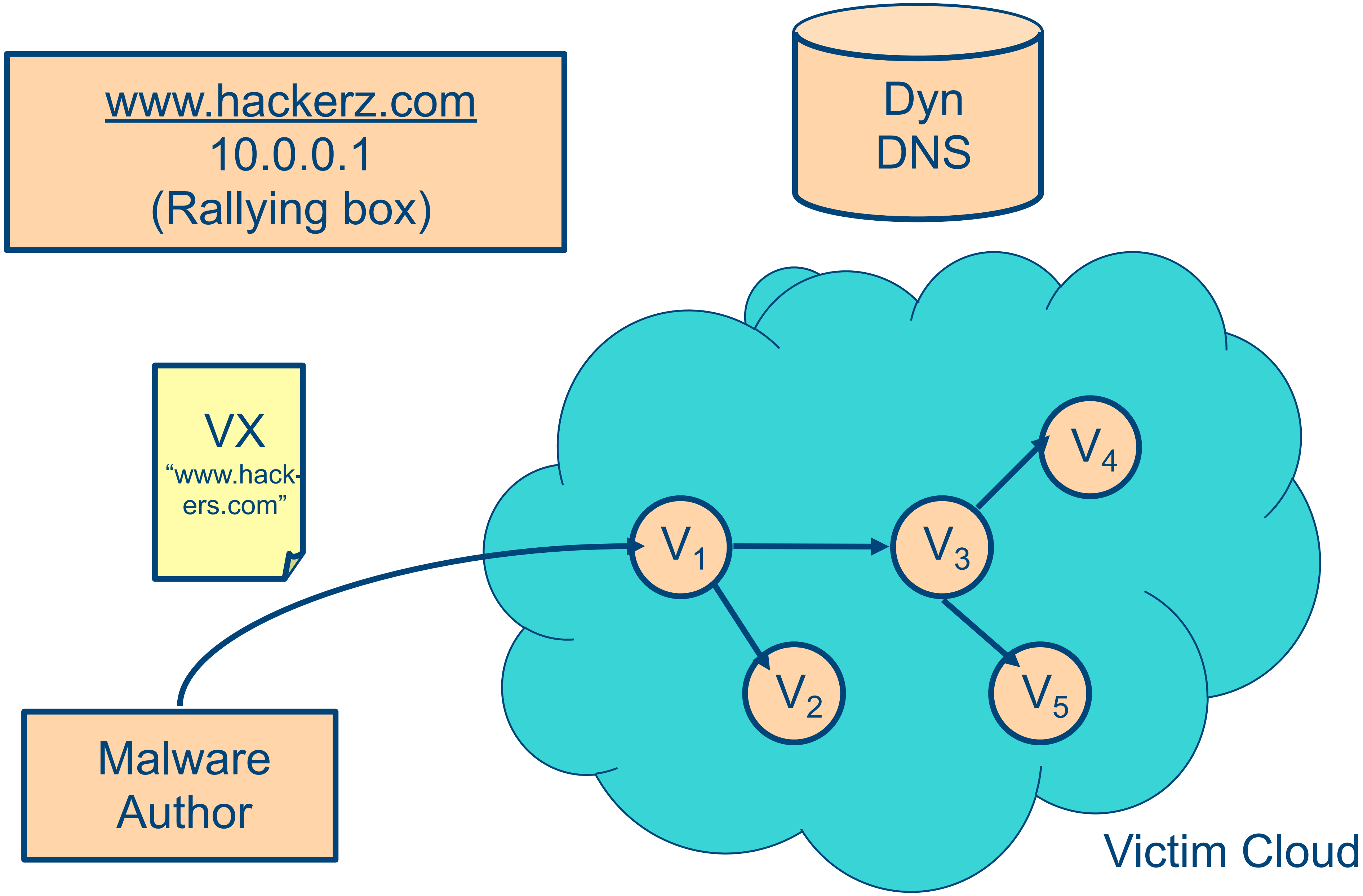
Removal Example

- 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. (Your proxies will get burned.)
- “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.

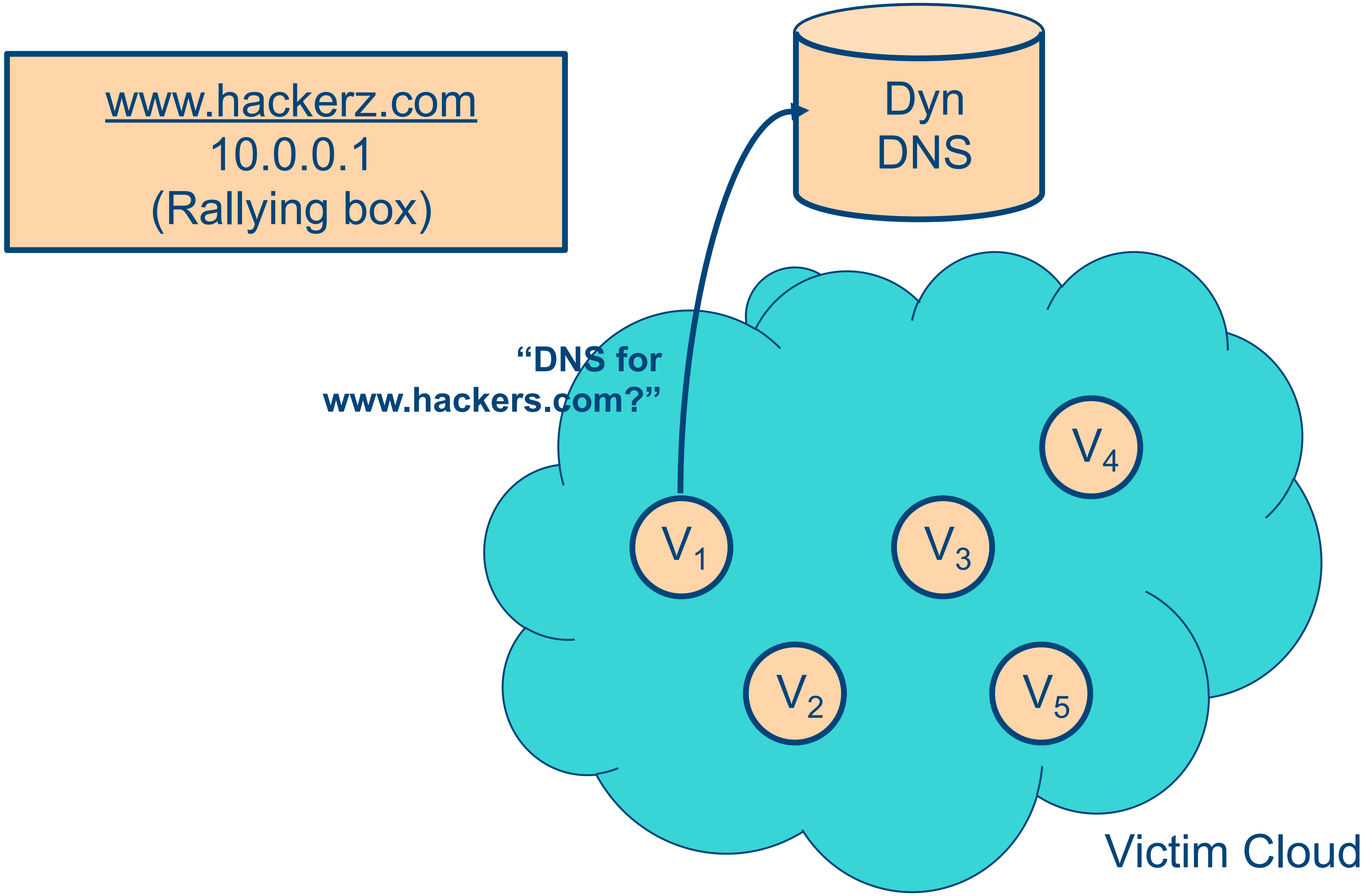
KarstNet: Responding to Botnets

- 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.

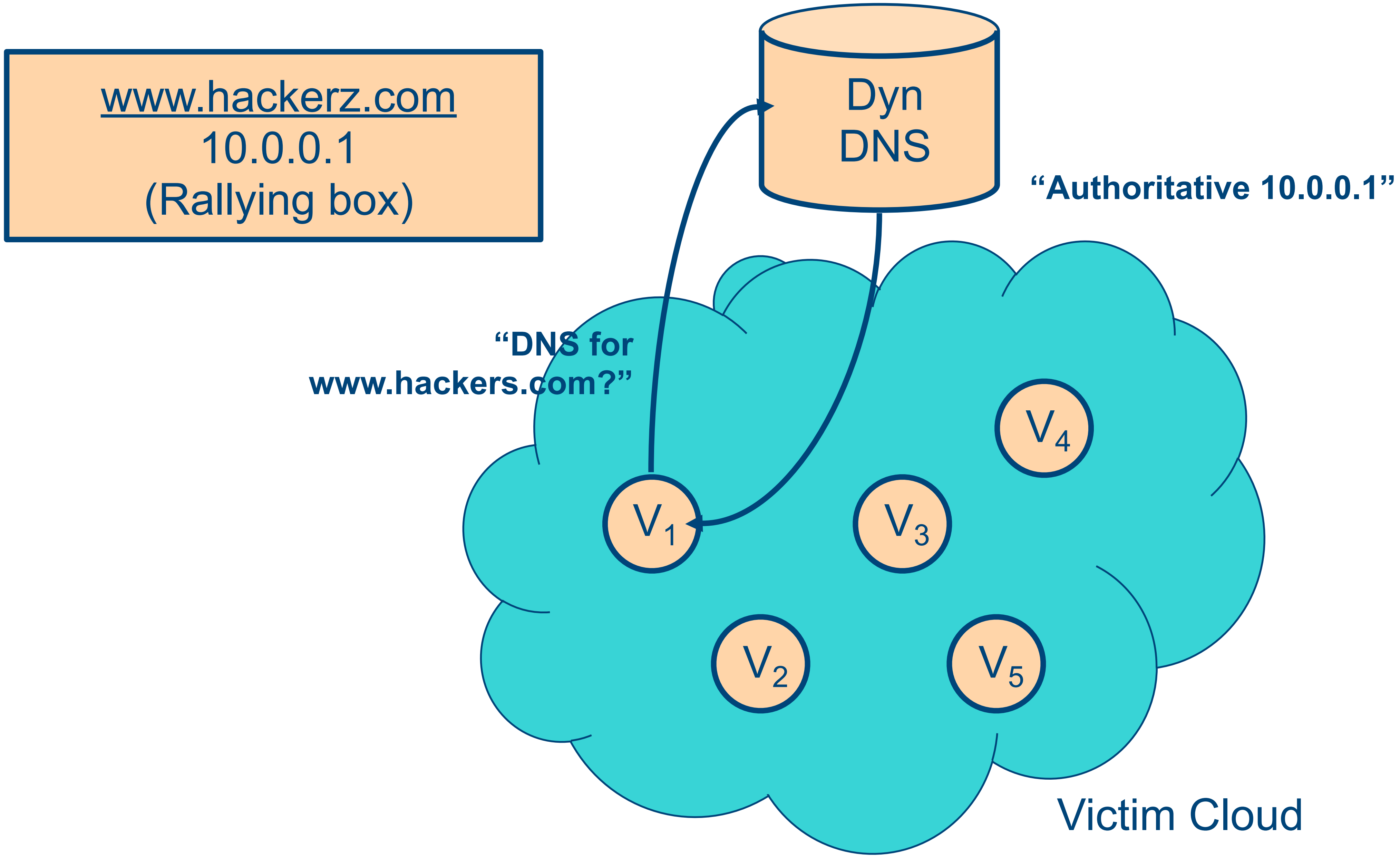
KarstNet: Malware with Strings



KarstNet: A-record Rallying



KarstNet: A-record Rallying



www.hackerz.com
10.0.0.1
(Rallying box)

Dyn DNS

"Authoritative 10.0.0.1"

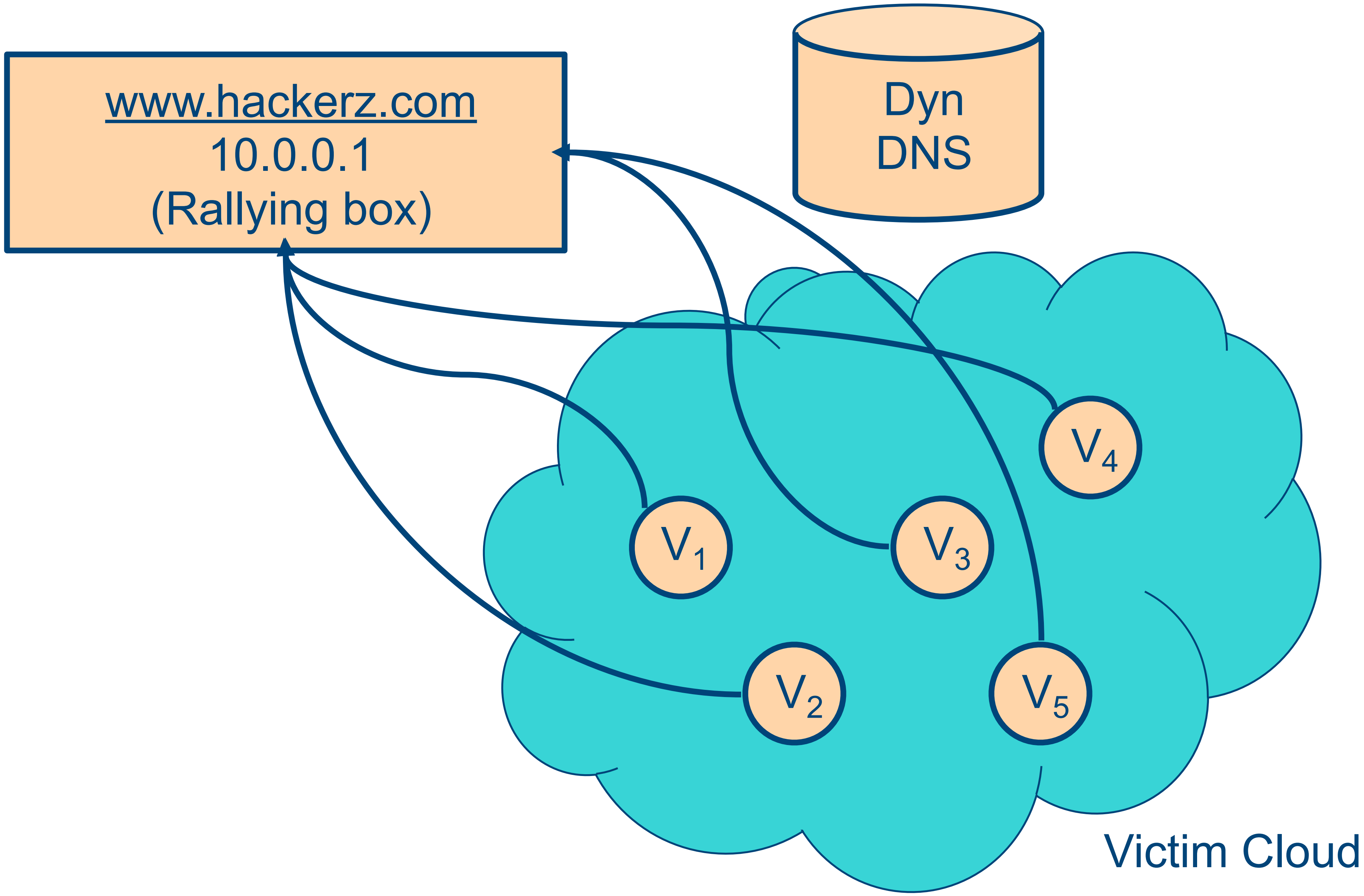
"DNS for www.hackers.com?"

V_1 V_2 V_3 V_4 V_5

Victim Cloud

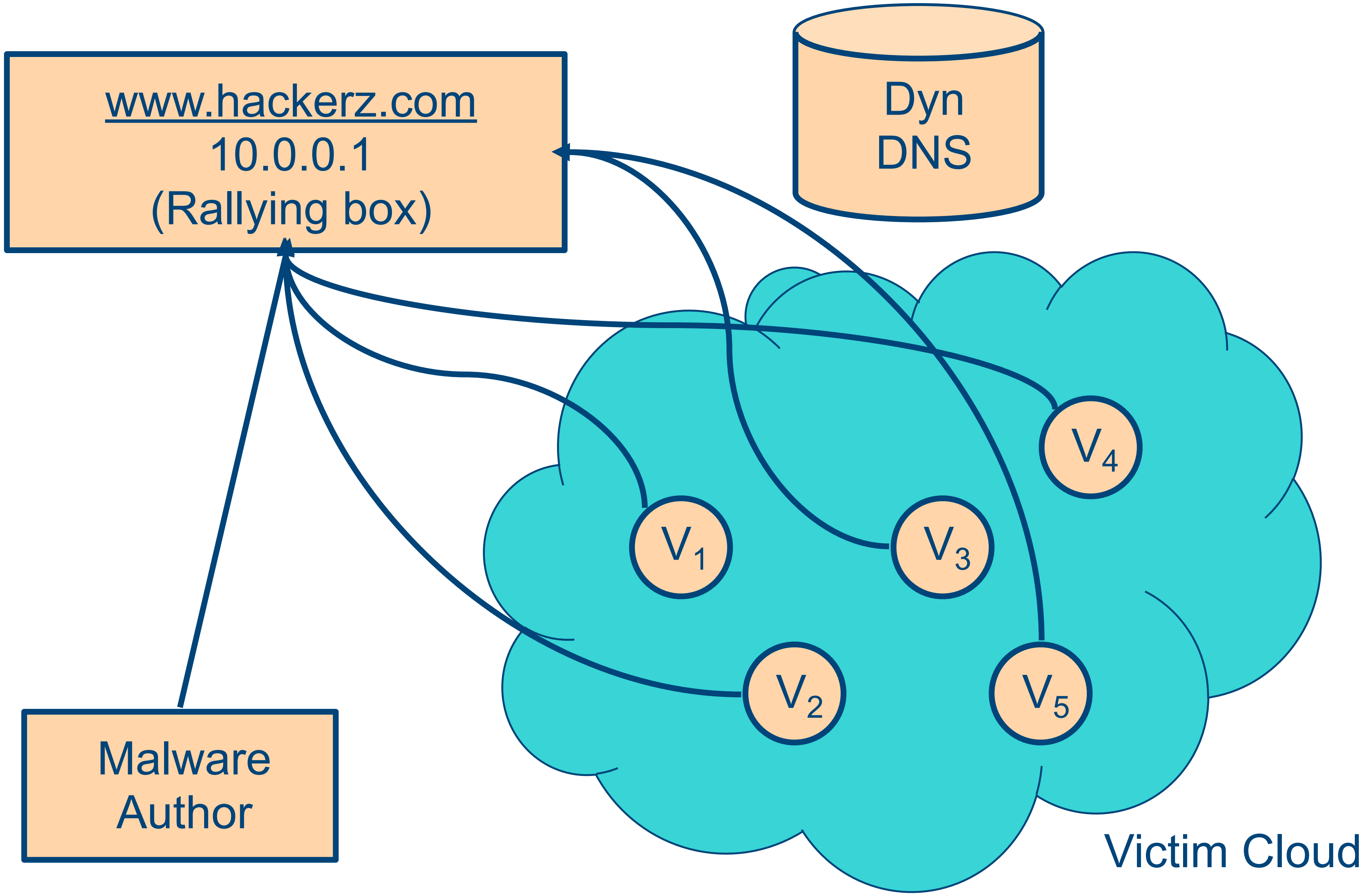
[Singh]

KarstNet: Command and Control

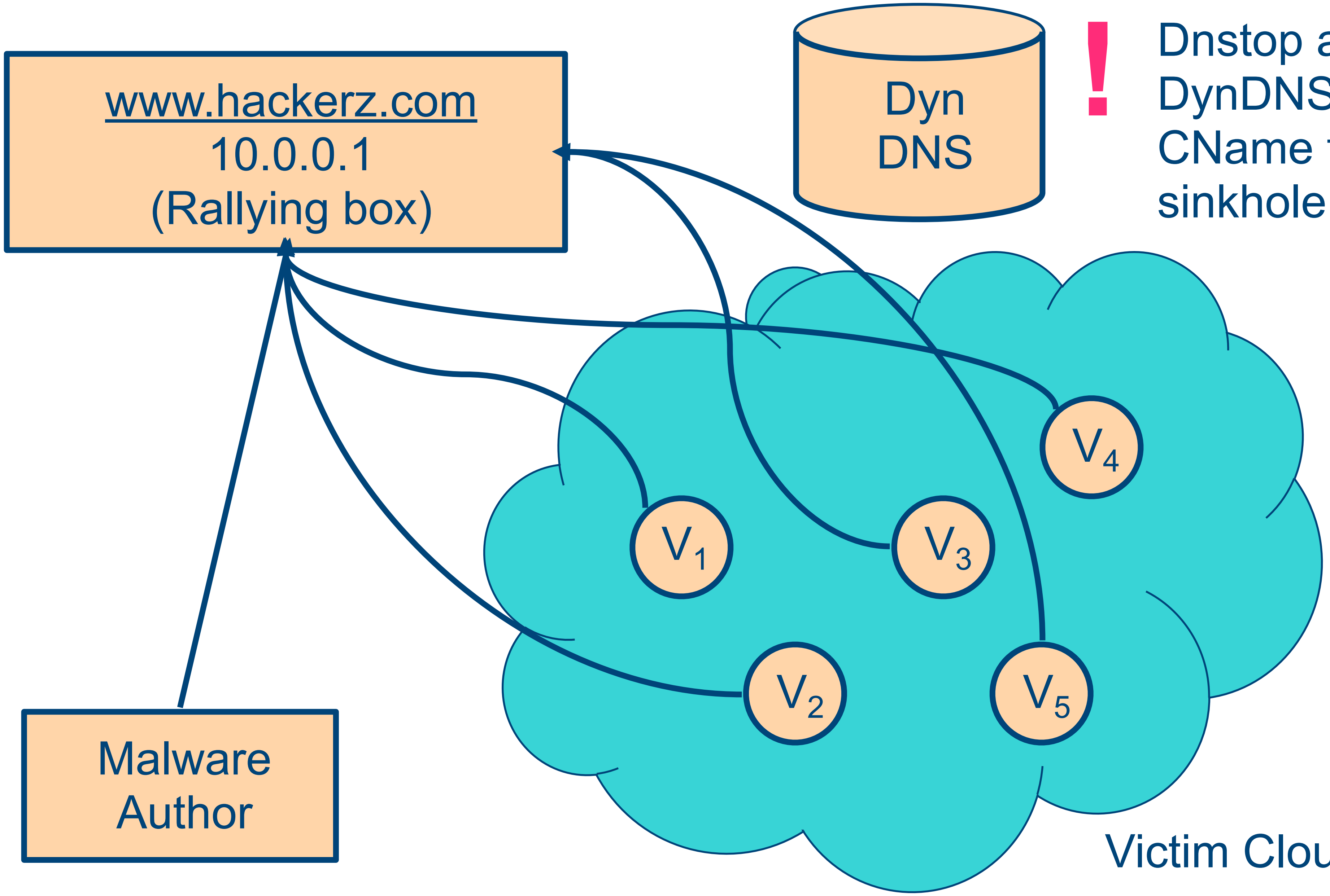


[Singh]

KarstNet: Command and Control

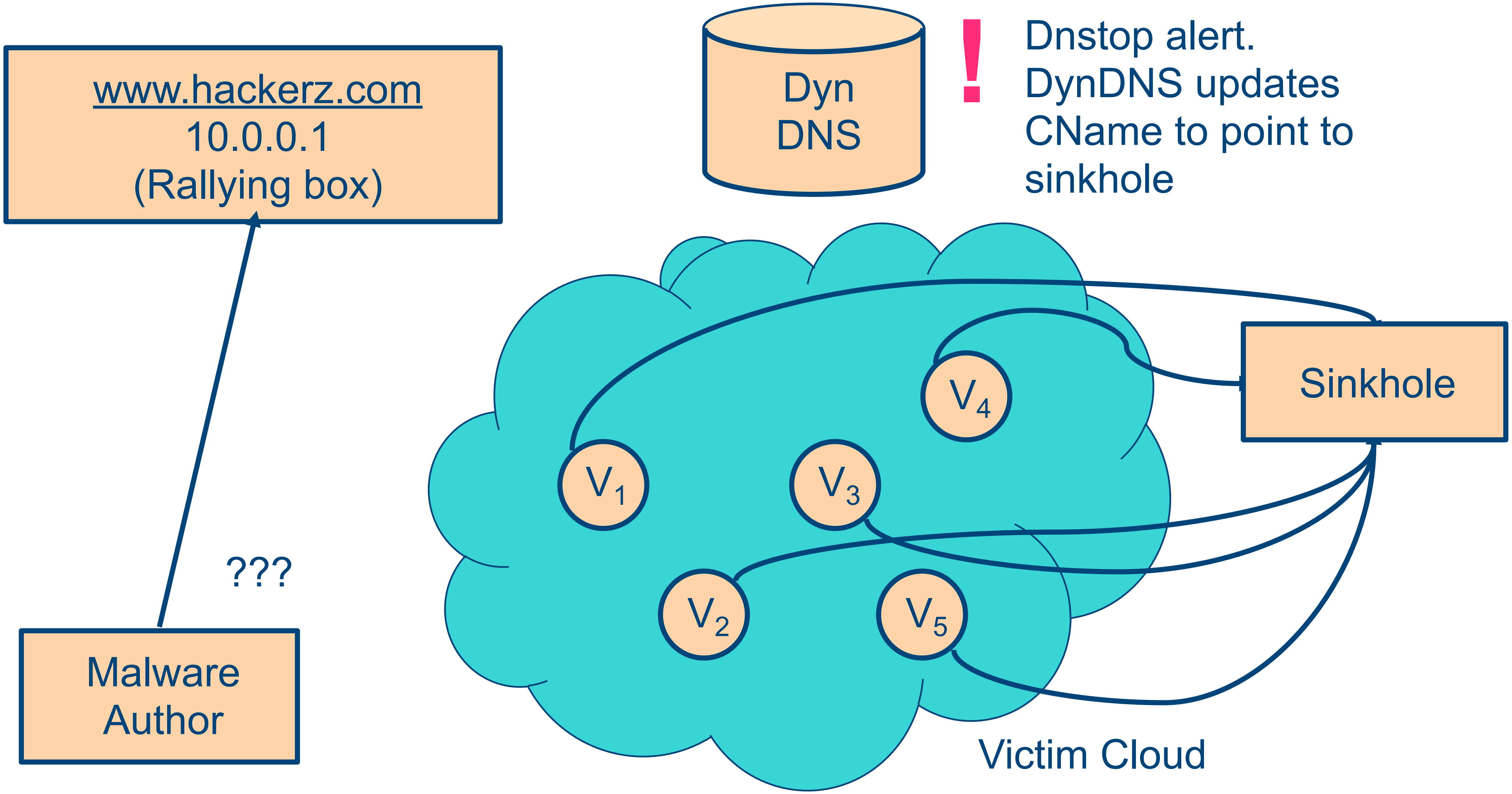


KarstNet: Detection



Dnstop alert. DynDNS updates CName to point to sinkhole

Drone Army Responses: DNS



Dnstop alert.
DynDNS updates
CName to point to
sinkhole

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)