A half-day tutorial @ ICIMP 2010, May 9, 2010, Barcelona, Spain

Understanding the Threat of Botnets

Presented By:
Dr. Basheer Al-Duwairi
Department of Network Engineering & Security
Jordan University of Science and Technology

basheer@just.edu.jo
http://www.just.edu.jo/~basheer

Table of Contents

Table of Contents (contd.)

Module I: Introduction

Talk Outline – Module I

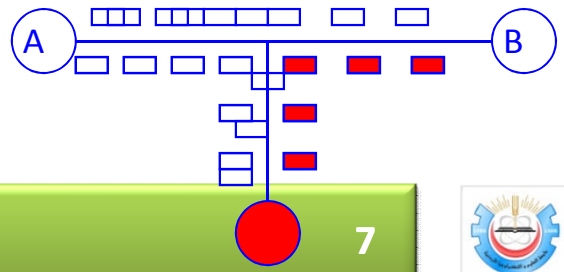
- **Basic security concepts**
- Information Security vs. Infrastructure Security
- Emerging Security Threats
- An Overview of Botnets

Characteristics of a Secure Network

- **Confidentiality:** message content should be accessed by authorized users only (achieved by encryption/decryption)
- **Authentication:** sender, receiver want to confirm identity of each other (achieved using digital signature)
- **Message Integrity:** Making sure that message was not altered in transit, or afterwards without detection (achieved by hashing)
- **Non-Repudiation:** The actual sender cannot claim that he did not send the message (achieved using digital certificates)
- **Availability:** services must be accessible and available to authorized users; i.e., preventing unauthorized withholding of messages
- Public Key Infrastructure (PKI) aims to achieve these characteristics

Security Threats

- **Interruption:** preventing messages from reaching authorized users
- **Interception:** getting access to the message content
- **Modification:** altering the message content
- **Fabrication:** creating a new message that appears to be coming from authorized user
- **Replication:** sending previously sent message at a later time



Security threats/characteristics mapping

Security Threat	Characteristics affected
Interruption	Availability
Interception	Confidentiality
Modification	Integrity Confidentiality Availability
Fabrication	Authentication Availability
Replication	Authentication Availability

Securing the Internet is difficult

- **Open and interoperable protocols:** while desirable, tend to work against security
- **Security/performance tradeoff:** performance is traditionally preferred
- **Security is expensive:** special resources are needed to support it
- **People do not like security:** security often complicates usage
- **Attackers enjoy breaking into a system:** some people see circumventing security as a challenge and enjoy doing it
- **Internet Infrastructure is vulnerable:** most systems and networks were not designed with security concerns in mind

Talk Outline – Module I

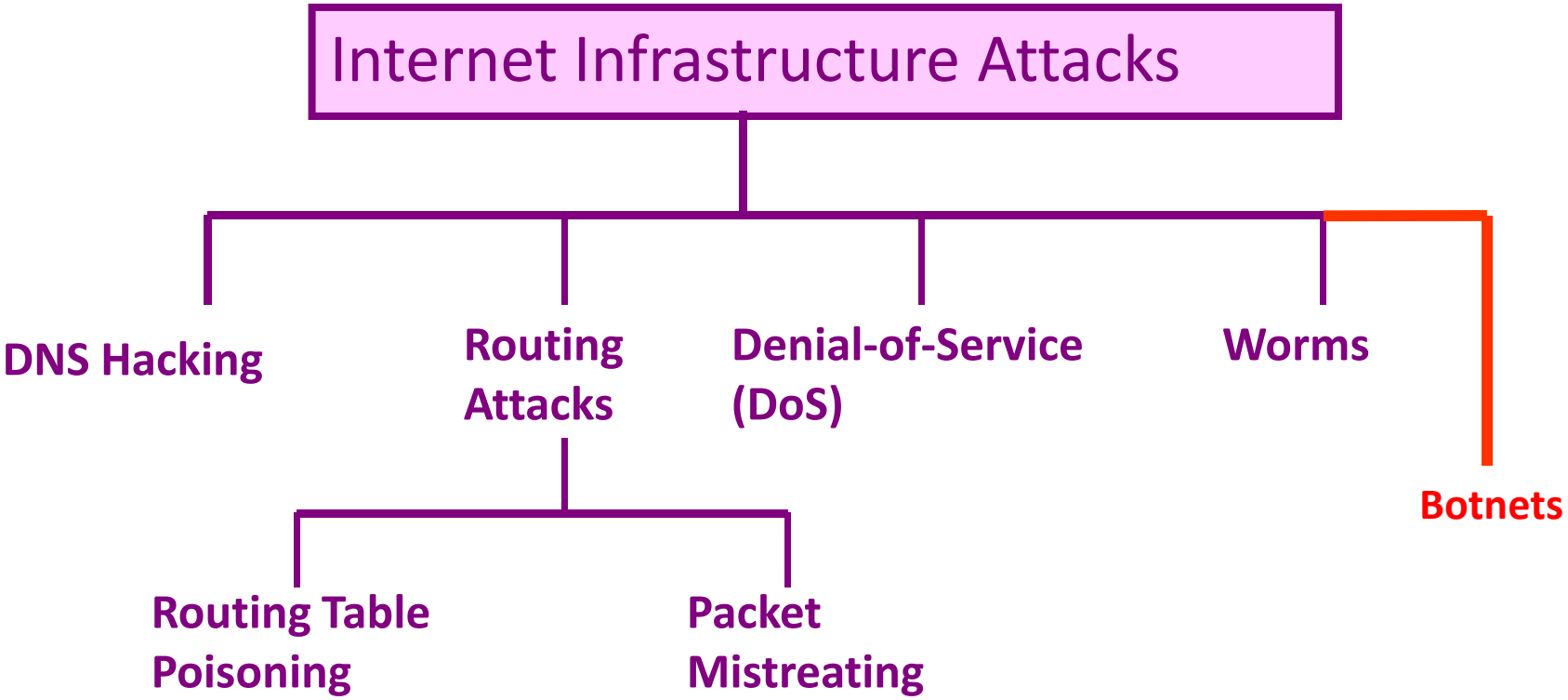
- Basic security concepts
- **Information Security vs. Infrastructure Security**
- Emerging Security Threats
- An Overview of Botnets

Information Security vs. Infrastructure security

	Information Security	Infrastructure Security
Scope	▫ Information Protection ▪ Message confidentiality ▪ Message Integrity ▪ Message Authenticity ▪ Non-repudiation	▫ Infrastructure Protection ▪ Routers ▪ DNS Servers ▪ Communication Links ▪ Internet Protocols ▫ Service Availability
Approach	▫ Encryption/Decryption ▫ Digital Signatures ▫ Message Authentication Code ▫ PKI	▫ Traffic Monitoring & Firewalls ▫ Intrusion Detection ▫ DoS Prevention, Mitigation, and Traceback ▫ Secure Internet protocols ▫ Wireless Infrastructure Security

[Chakrabarti et. al., IEEE Network, Nov/Dec. 2002]

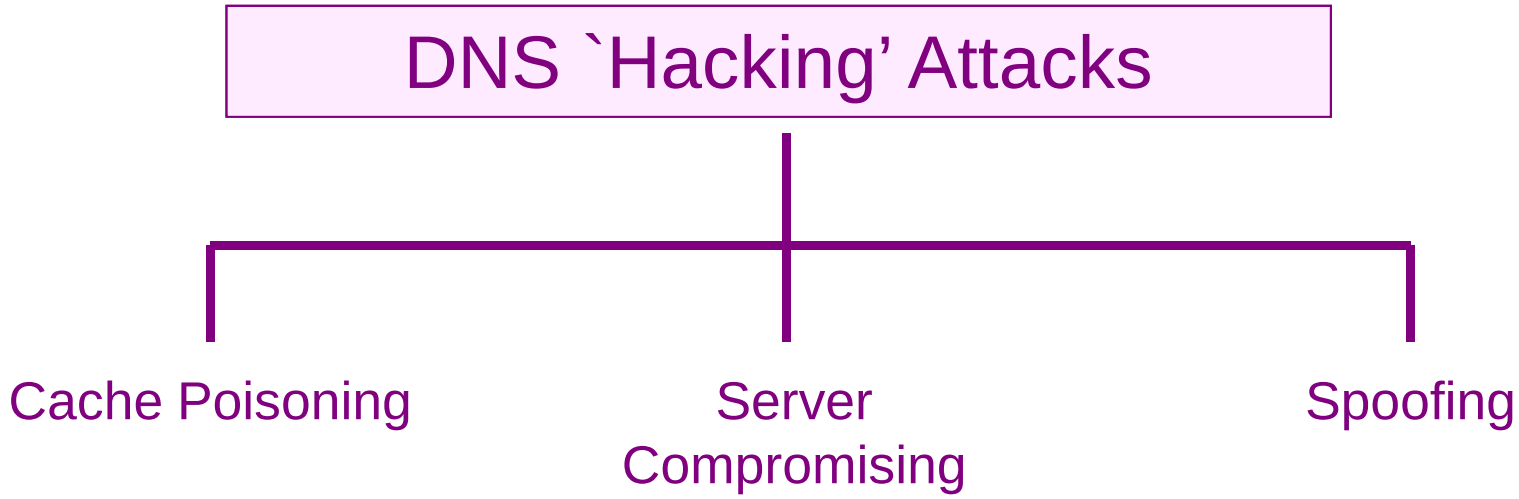
Attack Taxonomy



[Householder et. al., Security & Privacy 2002]

[Householder et. al., Security & Privacy 2002]

DNS `Hacking'

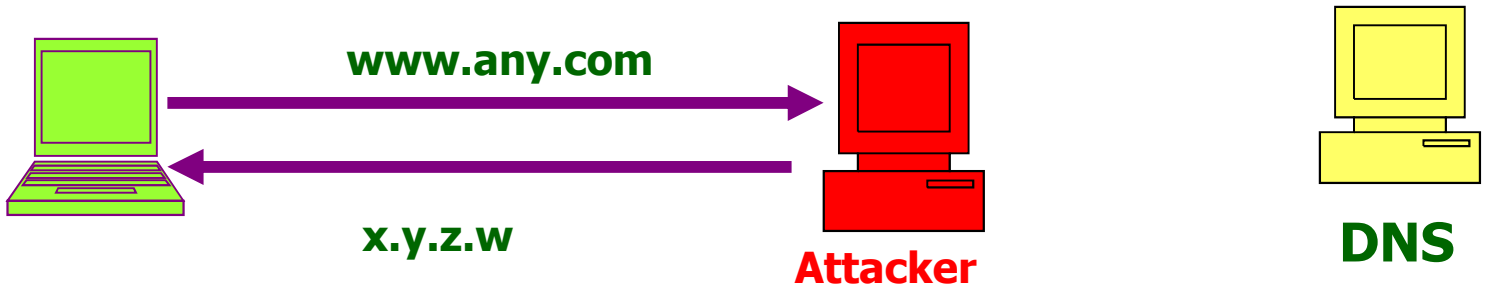


- Consequences:
  - Denial-of-Service
  - Domain Hijacking

DNS Attacks- Server Compromising

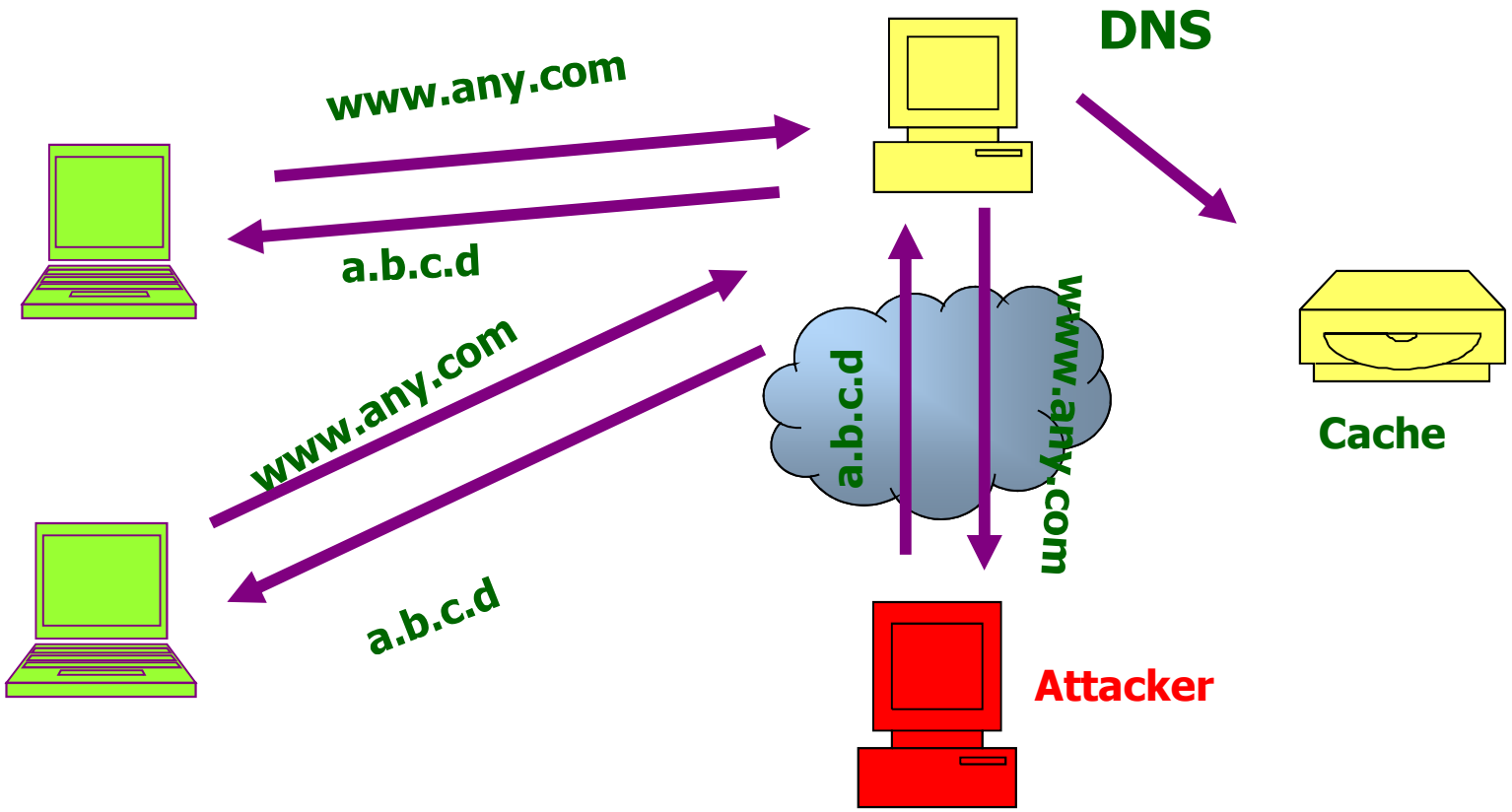
- Attackers can compromise a DNS server, thus giving them the ability to modify the data served to the users
- These compromised servers can be used for cache “poisoning” or DoS attacks on some other server

DNS Attacks- Spoofing



- The attacker masquerades as a DNS server and feeds the client wrong and/or potentially malicious information
- This type of attack can also redirect the traffic to a site under attacker's control and also launch a DoS attack on the unsuspecting client

DNS Attacks- Cache Poisoning



[Eastlake, IETF RFC 2535, Mar. 1999]

DNSSEC

- Designed to provide end-to-end integrity and authenticity of DNS data

- Public Key cryptography helps to answer these questions
  - One can use signatures to check integrity and authenticity of data
  - One can verify the authenticity of the signatures

- Key Distribution
  - A resource record format (KEY) is defined to associate keys with DNS names
  - Can be used to distribute keys associated with other applications and protocols (e.g., IPsec)

- Data Origin Authentication and Integrity
  - A resolver could learn a public key of a zone either by reading it from the DNS or by having it statically configured
  - A resource record format (SIG) is defined to cryptographically bind the RRset being signed to the signer and a validity interval

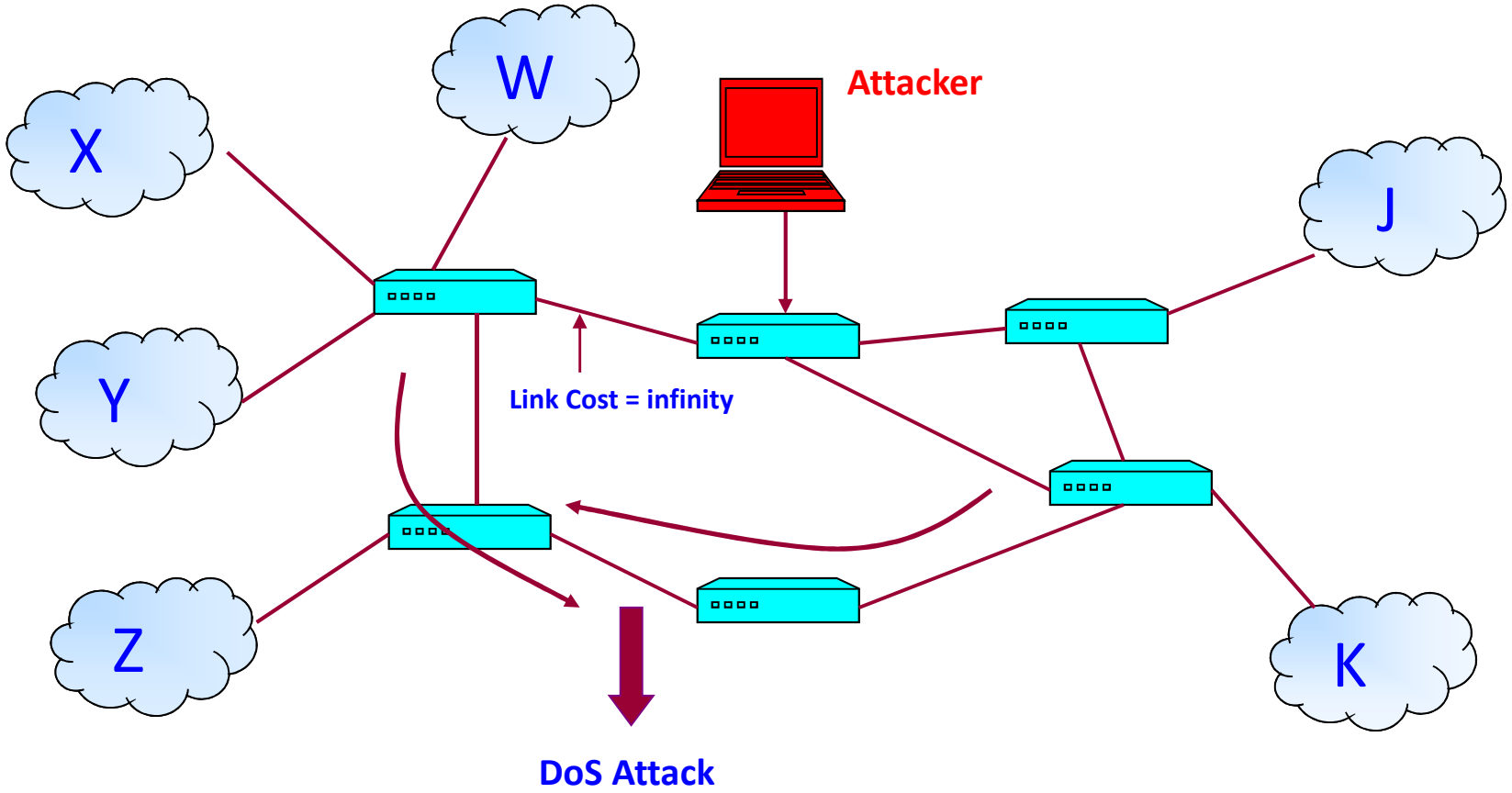
- DNS Transaction and Request Authentication
  - A resolver can be sure it is at least getting messages from the server it thinks it queried and that the response is for the query it sent
  - Requests can also be authenticated by including a special SIG RR at the end of the request

Routing Tables

- Used by each node to route packets
- Created by Routing Protocols
  - Intra-domain routing
    - OSPF, ISIS, RIP
  - Inter-domain routing
    - EGP, BGP
  - Link state routing protocols
    - OSPF
  - Distance vector routing protocols
    - RIP
  - Path vector routing protocols
    - BGP

[Chakrabarti et. al., IEEE Network, Nov/Dec. 2002]

An Attack Scenario- Routing Table Poisoning

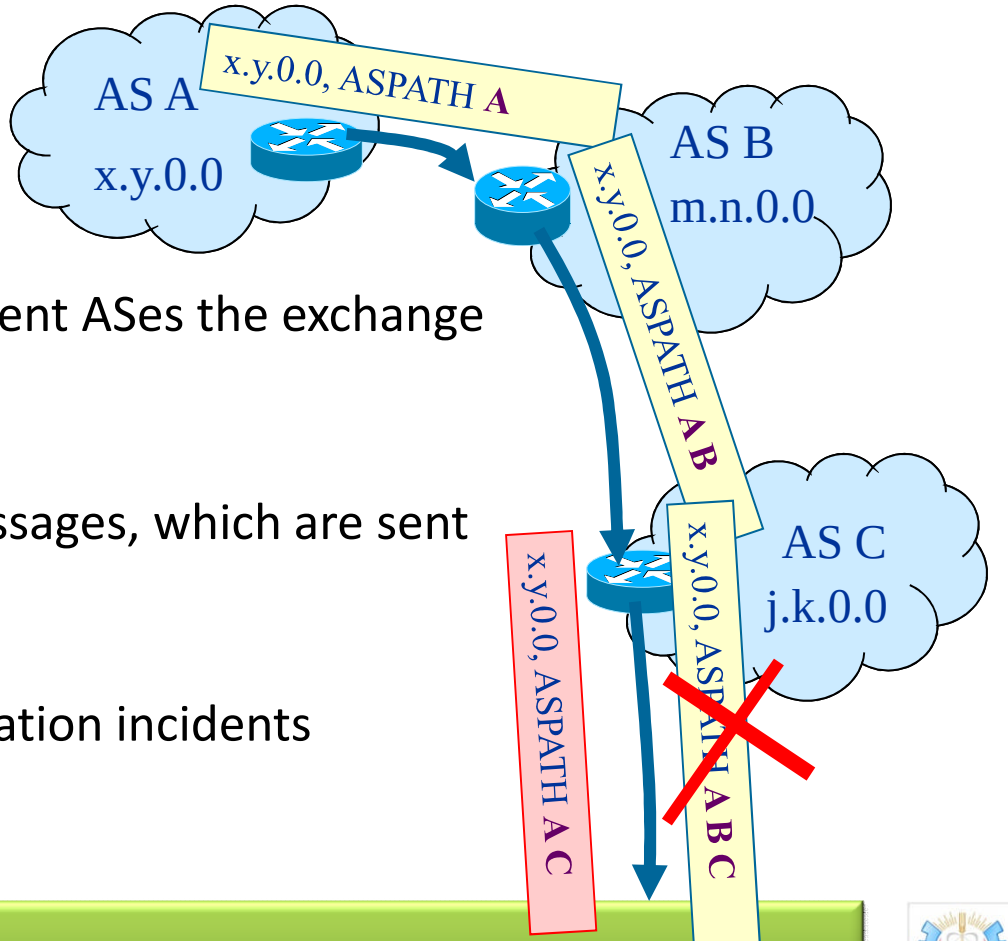


Impact of ‘Routing Table Poisoning’

- Sub-optimal routes, routing loops
- Congestion
- Network Partition
- Blackhole
- Denial of Service
- Overwhelmed hosts
- Traffic subversion

BGP Security Threats

- BGP is central for Internet packet routing
- BGP allows gateways in different ASes the exchange of routing information
- BGP operates in terms of messages, which are sent over TCP connections
- Many attack and misconfiguration incidents



BGP Security Threats (Contd.)

- Falsification attacks
  - A bogus BGP protocol message that differs from a message that a correctly configured router would send
  - Falsify what?
    - NLRI : originate a route to a prefix with which it is not affiliated, advertise longer prefix for a given route
    - Path attributes: truncation attack, modification attack
    - Withdrawn routers: send withdrawals for a working route
- Denial of service attacks
  - Exhaust router's computation resources
  - Exhaust the bandwidth
  - Lower layer protocol attacks

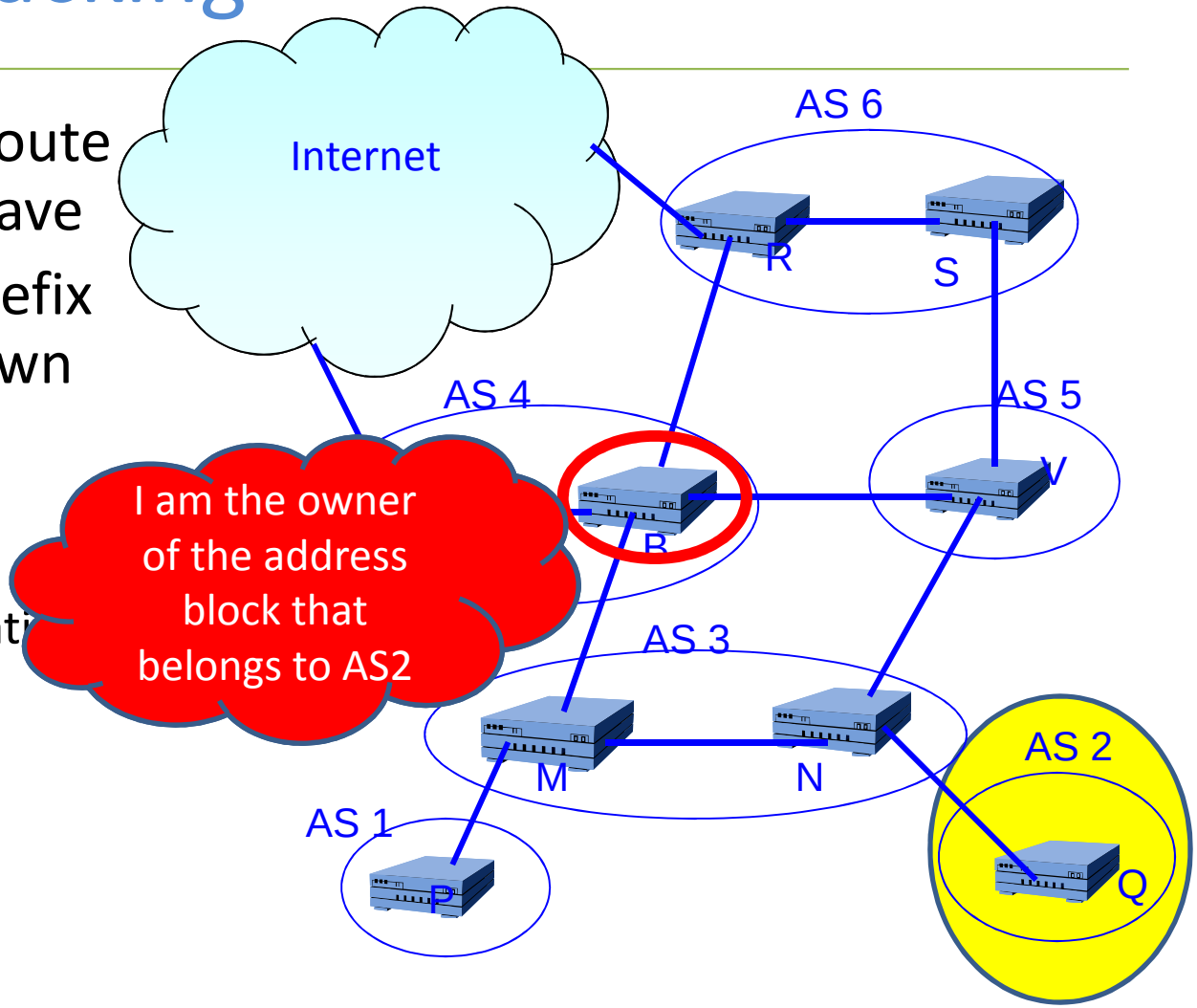
BGP Attack Mechanisms

- A compromised router can modify, drop, or introduce fake BGP updates → other routers have incorrect view of the network

- The effectiveness of some attacks depends on
  - The AS topology
  - The location of the compromised router relative to the victim network

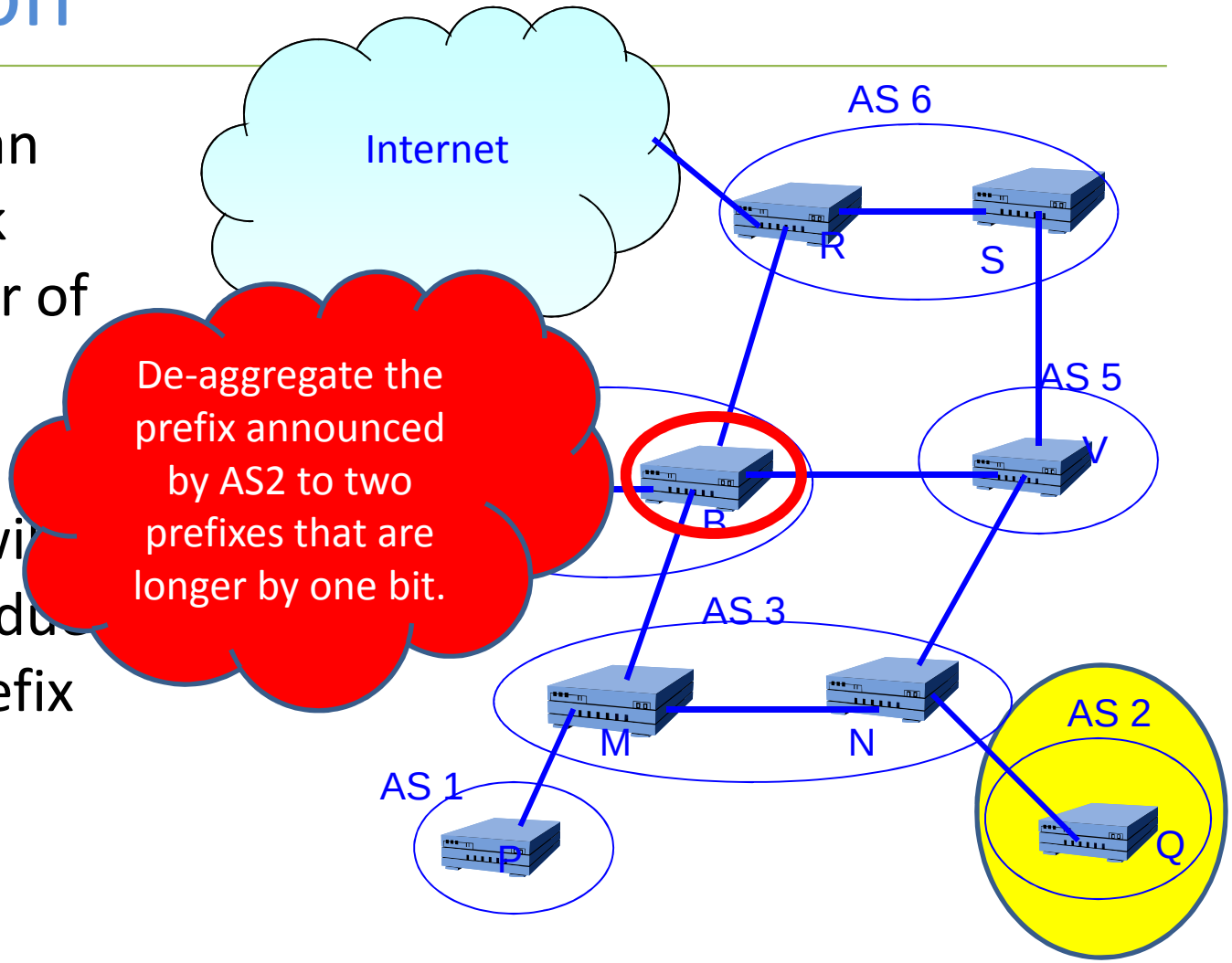
BGP Attack Mechanisms –False Updates and Prefix hijacking

- AS announces a route that it does not have
- AS originates a prefix that it does not own
  - Blackholing
  - Multiple Origin AS (MOAS) conflicts
    - Due to configurati errors
    - Causes partial connectivity



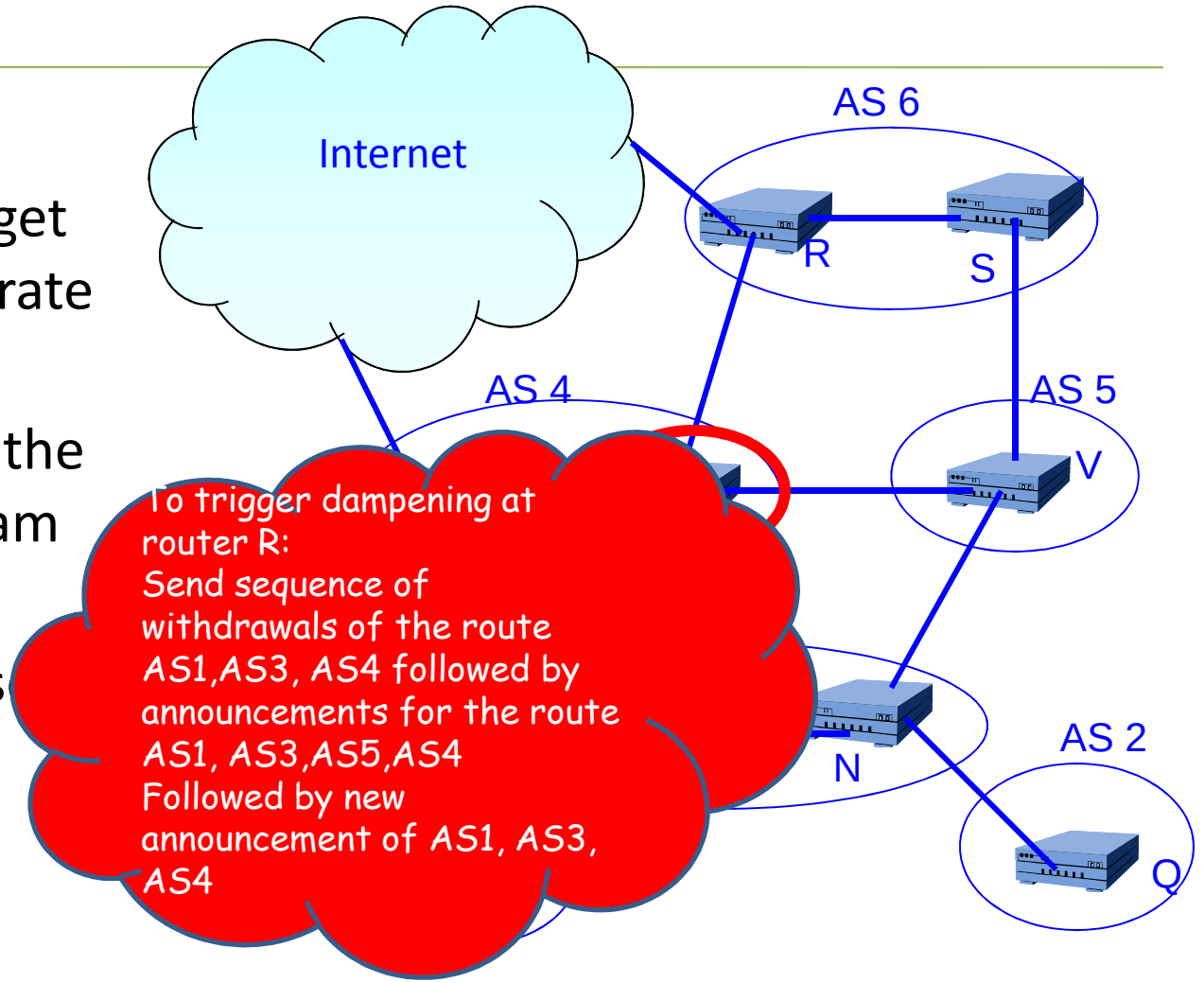
BGP Attack Mechanisms-Prefix De-aggregation

- Breaking up an address block into a number of more specific prefixes
- Fake routes wi… be preferred du… to longest prefix matching
- Blackholing



BGP Attack Mechanisms -Advertent link flapping

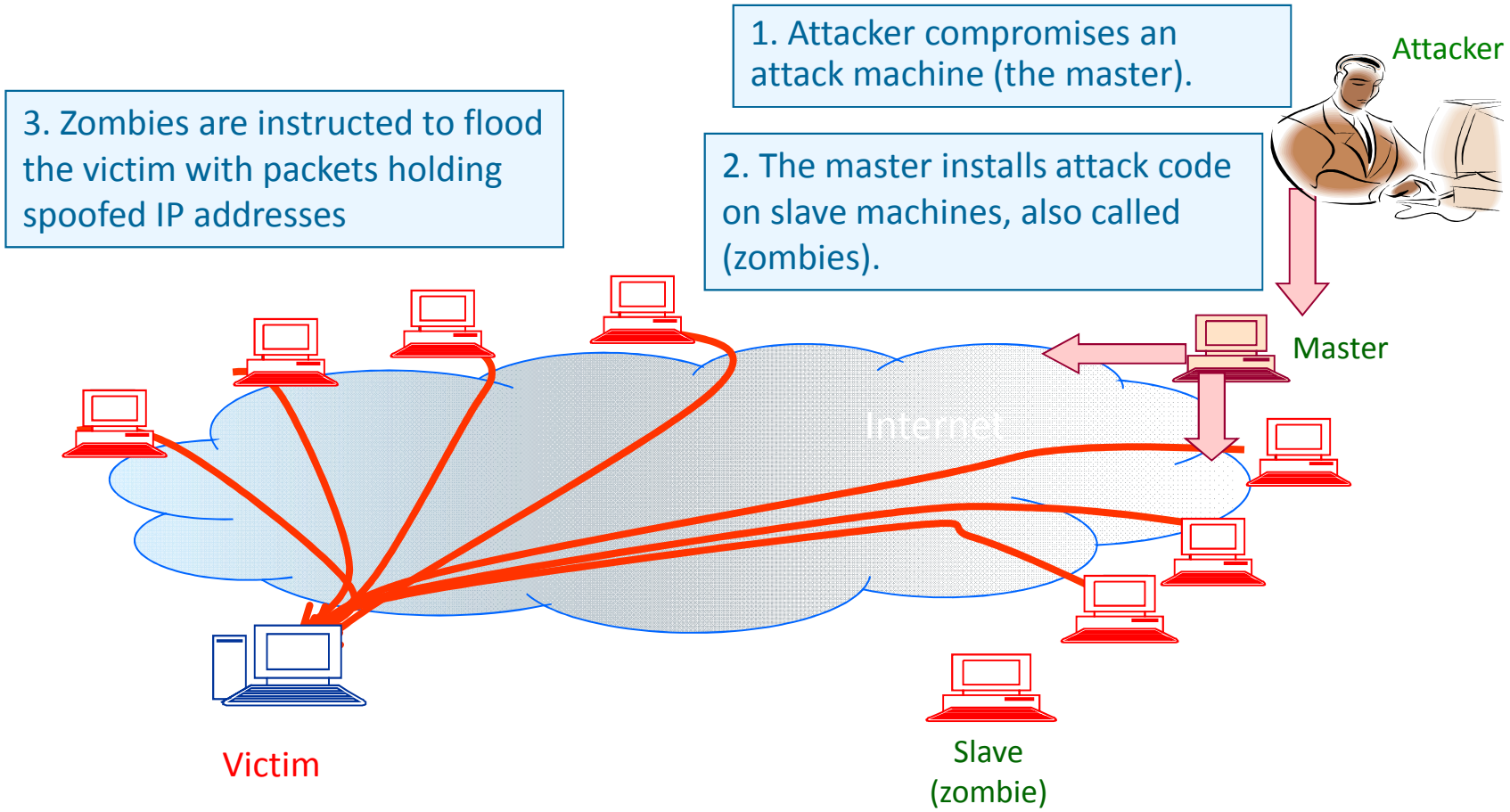
- Announcing and withdrawing target routes at a high rate
- Trigger "route dampening" for the victim at upstream router
- Dampening caus redirection, unreachability



DoS Attacks- An overview

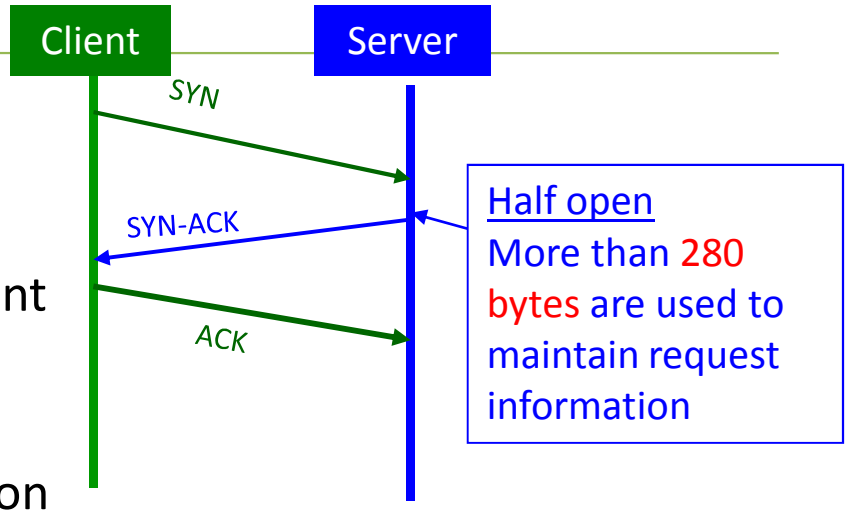
- **Denial of Service (DoS) attacks**
  - malicious means of denying Internet services
- Survey over 3-weeks period [Moore et.al., USENIX Security 2001]
  - 12,000 attacks against 5000 targets
  - Intensity as high as 600,000 packets/sec
- Easy to conduct yet difficult to defeat due to many factors
  - Destination oriented routing
  - Lack of authenticity over the Internet
  - Deterministic nature of Internet protocols

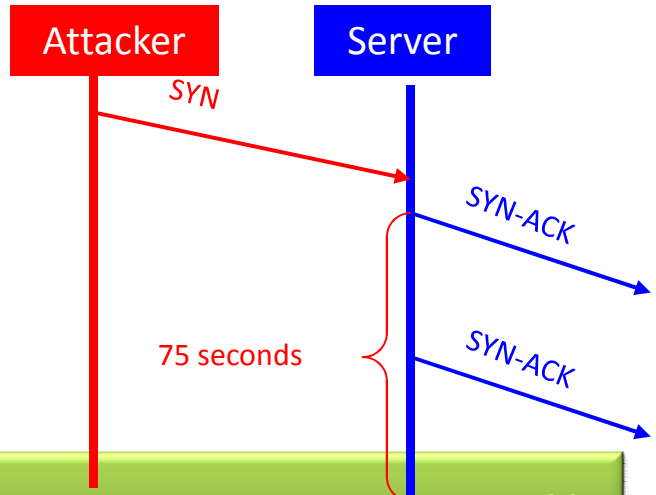
DoS Attack Scenario



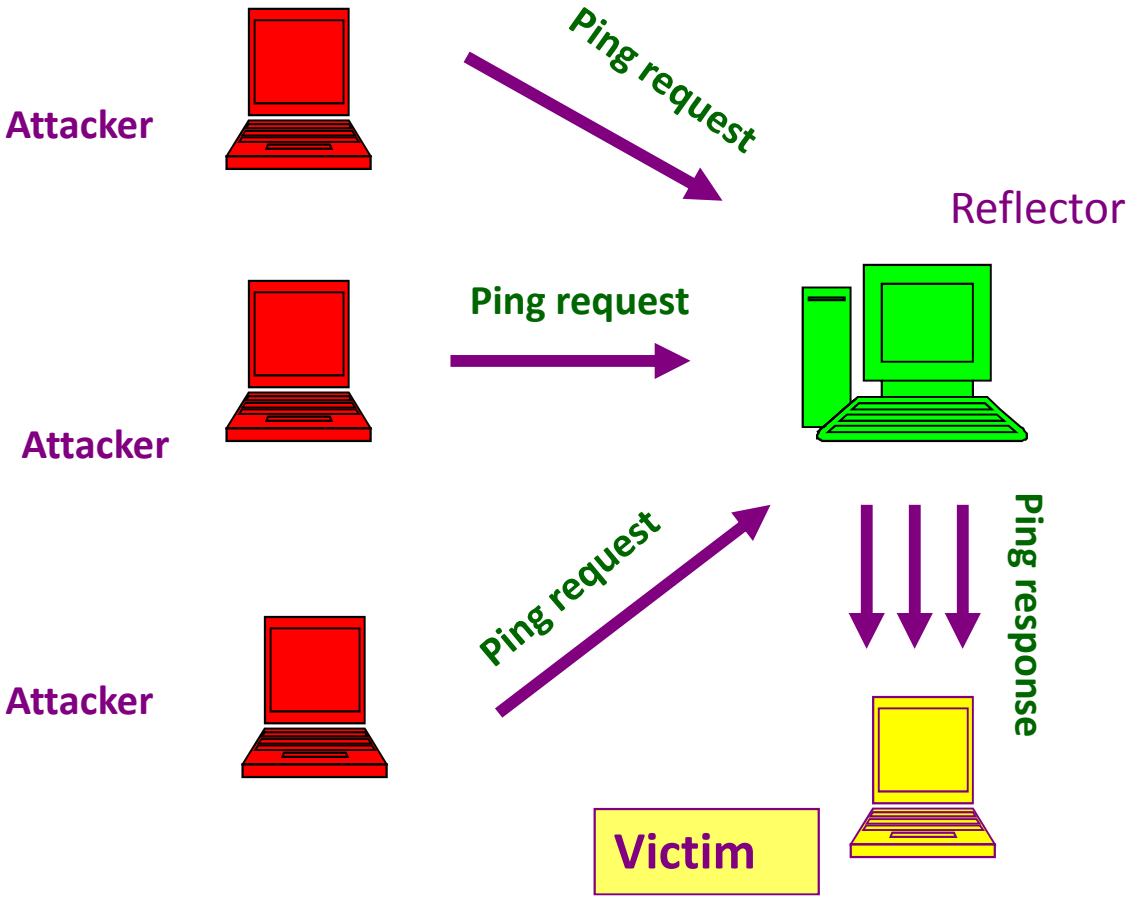
SYN Flooding Attacks

- **The attack**
  - Exploits the TCP connection establishment procedure
  - Floods the victim with spoofed connection establishment requests that will never complete
- **The impact**
  - Victim's & network's resources are consumed





Smurfing

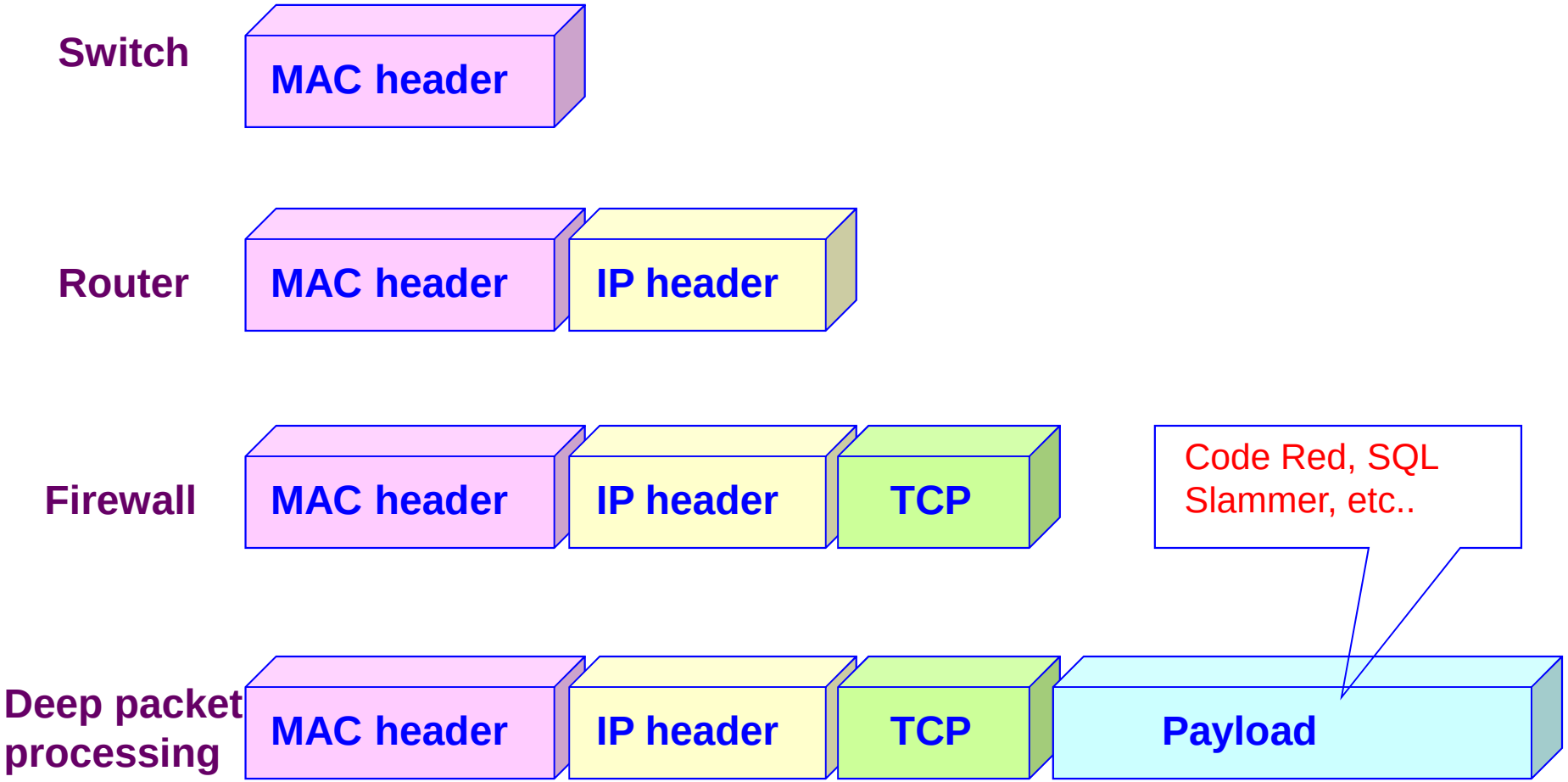

Attacker
Ping request
Reflector
Attacker
Ping request
Attacker
Ping request
Ping response
Victim

Worms

- Worm is a self-propagating malicious code
- Produces copies of itself and may also activate malicious code each time it activates
- Searches for systems to infect (exploits flaws in OS)
- Establishes a connection with the remote system
- Copies itself to the remote system, a new copy of worm is then run on the remote system
- Code Red worm infected more than 250K systems in just 9 hours on July 18, 2001 [Householder et. al., 2002]
- Counter-measures: Access control, Intrusion detection, Firewalls

[P. Jungck et. al., IEEE Computer, July 2004]

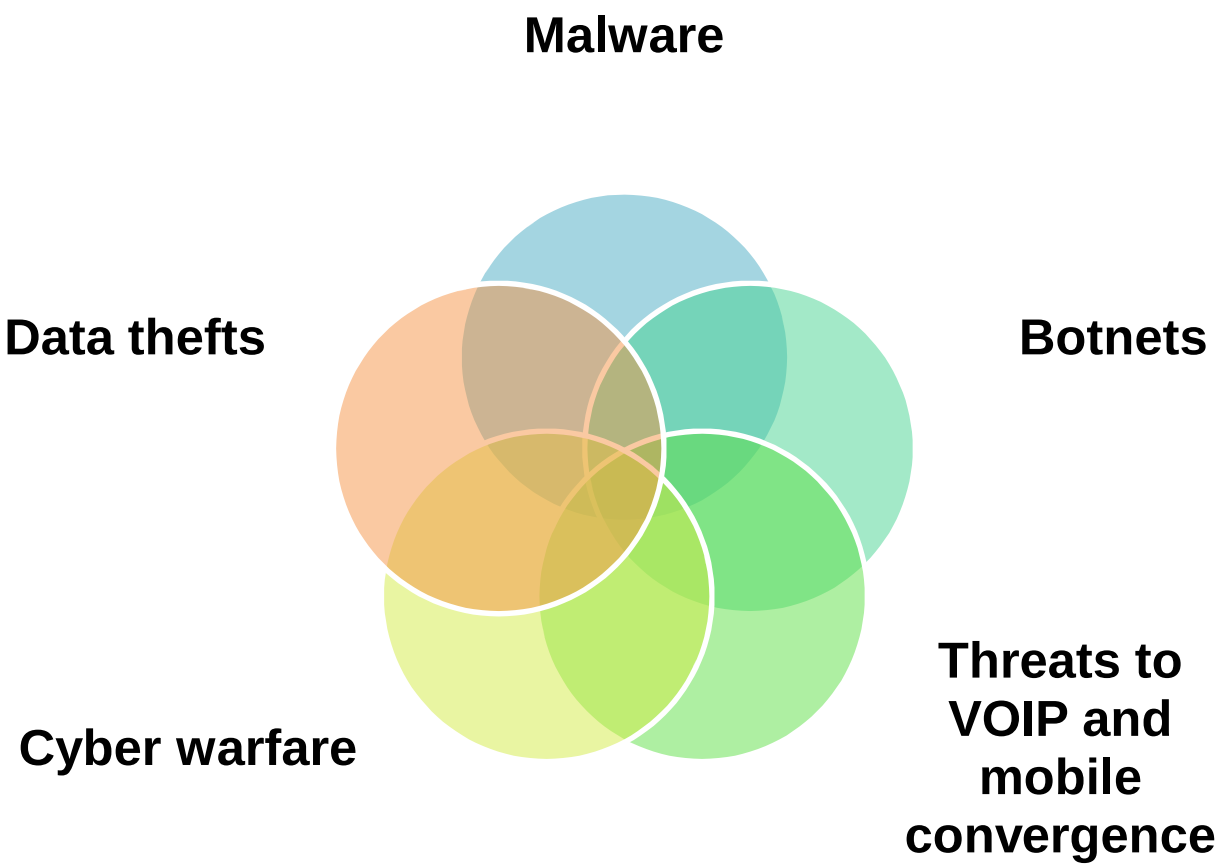
Packet Inspection



Talk Outline – Module I

- Basic security concepts
- Information Security vs. Infrastructure Security
- **Emerging Security Threats**
- An Overview of Botnets

Emerging Cyber Threats
Report of Georgia Tech Information Security Center (GTISC) - 2009



Malware (Malicious Software)

- Can be loosely defined as “Malicious computer executable”
- Running a code without user’s consent
- Reasons for increase
  - Growing number and connectivity of computers
  - Growing system complexity
  - Systems are easily extensible

A total of 28940 different malicious and potentially unwanted programs were detected on users’ computers in August. That is an increase of more than 8,000 on July's figures and points to a significant increase in the number of in-the-wild threats.
http://www.kaspersky.com/news?id=207575678

Cyber Warfare

- Security experts believe cyber warfare will accompany traditional military interaction more often

- Attacks that occurred between Russia and Georgia in 2009 as a model for military cyber engagements
- Increasing cyber warfare activity are due to:
  - The low cost to launch cyber attacks compared with physical attacks
  - The lack of cyber defenses
  - The "plausible deniability" the Internet affords
  - The lack of "cyber rules of engagement" in conflicts between nation states

"The future threat goes beyond what we think of as cyber-espionage and intellectual property theft, although that certainly remains a factor," said Heron. "I think we're going to see more technologically savvy, state-sponsored attacks to the IT systems that support foundational services here in the U.S."
George Heron - Founder, BlueFin Security

Threats to VoIP and Mobile Convergence

- VoIP infrastructure has been vulnerable to the same types of attacks that plague other networked computing architectures

"At this point, mobile device capability is far ahead of security," said Traynor. "We'll start to see the botnet problem infiltrate the mobile world in 2009."
Patrick Traynor - Assistant Professor, School of Computer Science at Georgia Tech, and member of the Georgia Tech Information Security Center

- Financial motivation and increased adoption will increase attacks to smartphones in the years to come. As more payment infrastructure gets placed on these devices, they will become a more attractive target

Data Theft and Cyber Crimes

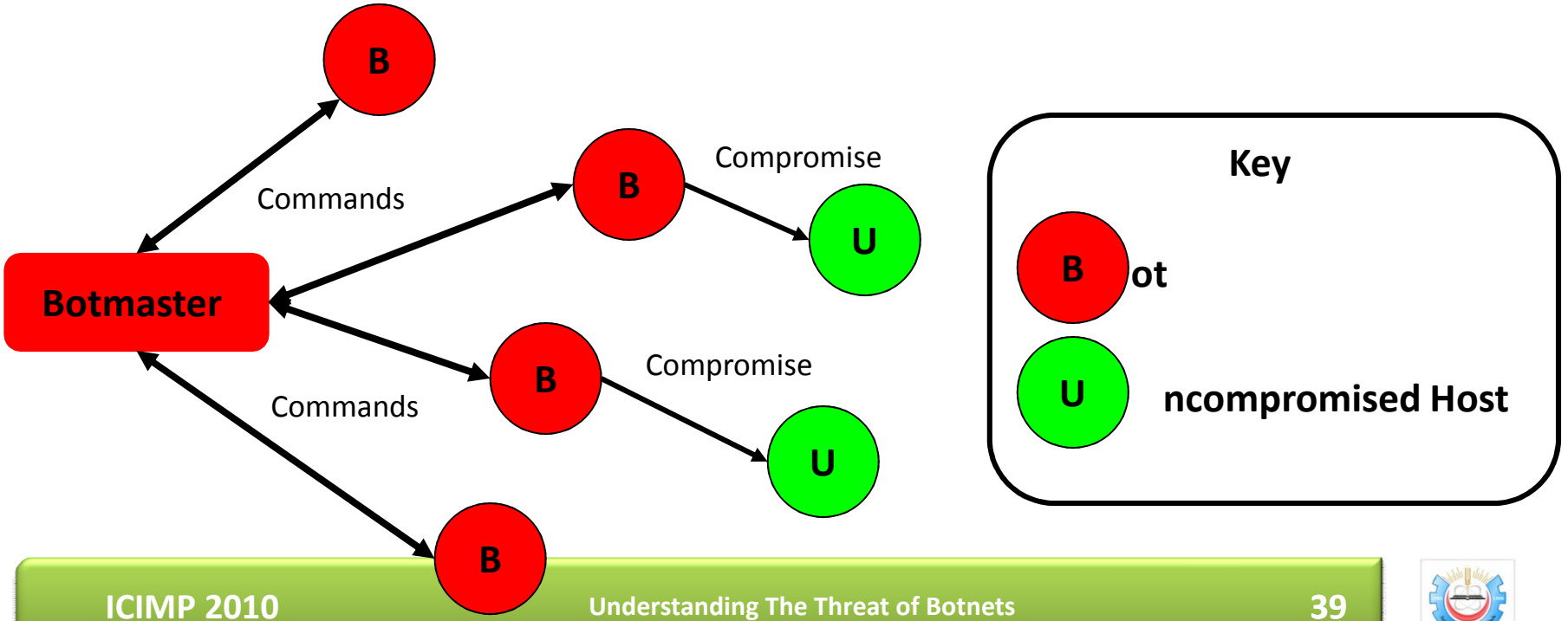
- Sources of cyber crime will become increasingly organized and profit-driven in the years ahead

- cyber criminal industry into three tiers:
  - Low-level criminals who use kits to create the specific malware required for their targeted crimes
  - Skilled developers and collectives of technical experts creating new components to embed within their commercial malware creation kits
  - Top-tier managed service providers that wrap new services around malware kits to increase propagation and enable organized fraud on a global scale, feeding gains back into existing money laundering chains

"The Web-based attack platforms come in a variety of packages and are available for lease, purchase or any payment model in between," said Ollmann.
Gunter Ollmann - Chief Security Strategist, IBM Internet Security Systems

Botnets

- A Botnet is coordinated group of malware instances that are controlled by a botmaster via some C&C channel.



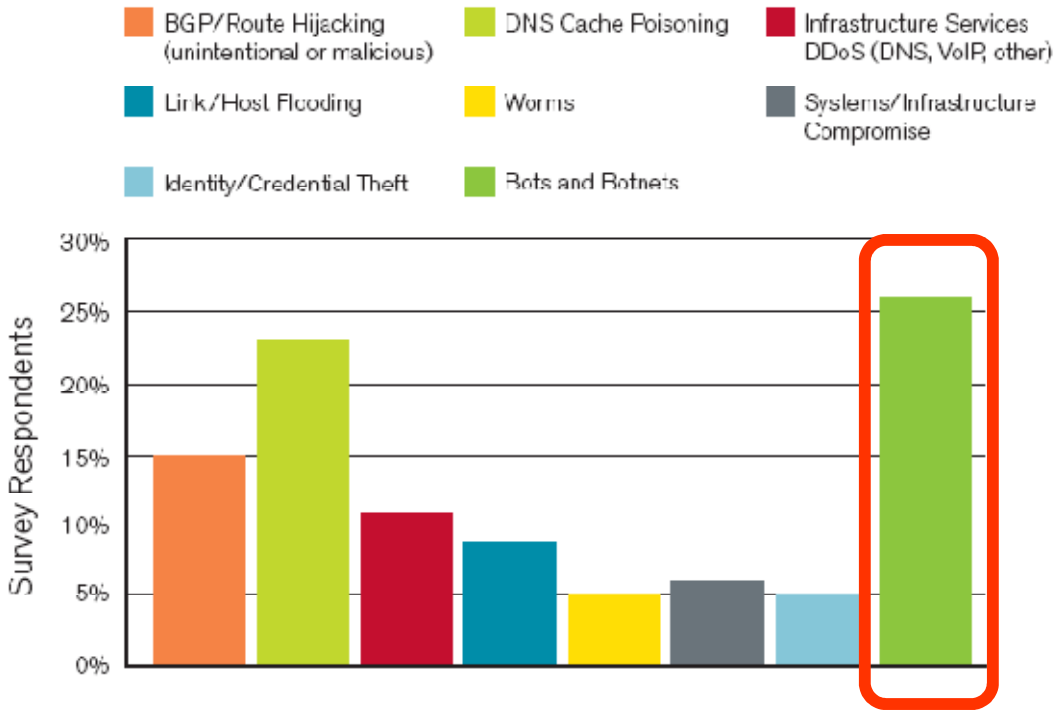
Botnets (Contd.)

- Three unavoidable factors that are spurring botnet growth:
  - Infection can occur even through legitimate Web sites
  - Bot exploits/malware delivery mechanisms are gaining sophistication and better obfuscation techniques
  - Users do not have to do anything to become infected; simply rendering a Web page can launch a botnet exploit

in 2Q 2008, 10 million bot computers were used to distribute spam and malware across the Internet each day
[http://www.darkreading.com/document.asp?doc_id=161524]

Botnets- A Significant Threat

- Most significant threats to network operators



- Source: Worldwide Infrastructure Security Report, Arbor Networks, Sep. 2008

Botnet scams are exploding

Updated 3/16/2008 10:13 PM | Comments 97 | Recommend 40 E-mail | Save | Print | Reprints & Permissions | RSS

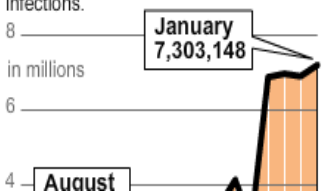
CYBERCRIME PAYS

The escalating number of botnets have helped feed a surge in various forms of online fraud.

Botnet deluge · Virus rate · E-mail spam · Phishing attacks

Botnet deluge

The average daily number of unique botnet communiqués to accept instructions from a controller, deliver spam, conduct phishing campaigns, click on ads to earn ad revenue, carry out denial-of-service attacks, steal data, scan for vulnerable computers, and spread infections.



By Byron Acohido and Jon Swartz, USA TODAY

SEATTLE — Two days after actor Heath Ledger died, e-mails began moving across the Internet purportedly carrying a link to a detailed police report divulging "the real reason" behind the actor's death. Ledger had been summarily drafted into the service of a botnet.

Bots are compromised computers controlled by profit-minded crooks. Those e-mails were spread by a network of thousands of bots, called a botnet. Anyone who clicked on the link got instantly absorbed into the fast-spreading Mega-D botnet, says security firm Marshal. Mega-D enriches its operators, mainly by distributing spam for male-enhancement pills.

BACKGROUND: Botnets can be used to blackmail targeted sites

Largely unnoticed by the public, botnets have come to inundate


CNN INTERNATIONAL .com
U.S. Edition
Member Center: Sign In | Register
MAKE CNN.com YOUR HOME PAGE
Every week on CNN International
Watch online now
world's untold stories
SEARCH
THE WEB
CNN.com
SEARCH
Home Page
Asia
Europe
U.S.
World
World Business
Technology
Science & Space
Entertainment
World Sport
Travel
Weather
Special Reports
Video
I-Reports
ONLY ON CNN
CNN Pipeline
What's On
Art of Life
Business Traveller
Future Summit
Inside the Mideast
Principal Voices
Quest
Revealed
Talk Asia
Services
Languages
TECHNOLOGY
CNNACCESS
ARCHIVE »
Expert: Botnets No. 1 emerging Internet threat
Tuesday, January 31, 2006 Posted: 1959 GMT (0359 HKT)
ATLANTA, Georgia (CNN) -- A "botnet" is a network of zombie computers -- thousands surreptitiously are infected with code that allows an unauthorized user to control them via the Internet. The computers can be used to spread spam, launch denial-of-service attacks against Web sites and conduct fraudulent activities.
Merrick Furst, professor of computing and associate dean for undergraduate programs at Georgia Tech's College of Computing, is conducting extensive research into botnets. He talked recently with CNN technology correspondent Daniel Sieberg.
SIEBERG: How do we understand what a botnet is exactly?
FURST: A botmaster is a criminal who wants to use your computer as a resource
story.furst.jpg
Merrick Furst, associate dean for undergraduate programs at Georgia Tech's College of Computing
YOUR E-MAIL ALERTS
Computer Security
Internet
CNN Access
advertisement
Kirloskar
Enriching Lives
KIRLOSKAR
In the business of

Talk Outline – Module I

- Basic security concepts
- Information Security vs. Infrastructure Security
- Emerging Security Threats
- **An Overview of Botnets**

[T. Holz. A short visit to the bot zoo. *IEEE Security & Privacy*, 3(3):76–79, 2005]

Botnets- An Overview

- Bots are used for various forms of illegal activity
- There are many types of bots available in the wild, with a lot of variants for each type
  - Agobot and SDbot are among the most popular
- Bots share similar characteristics in general
  - They take advantage of many of the software vulnerabilities such as software bugs, including those that enable:
    - buffer overflow attacks, hacker installed backdoors, and various memory management problems that allow malicious code to infect a system

Botnets- An Overview (Contd.)

- Publicizing bot code is one of the main reasons for the appearance of many bot variants within short period of time.
- Making bot's source code available for hackers enables them to modify it to obtain customized versions that serve their bad intents.
- Bots usually start their operation by estimating the infected system's bandwidth
  - This is typically done by accessing several servers and sending data to them
  - This measurement is of particular importance for the attacker especially when performing DDoS attack
- Overall, there are a lot of differences between bots which are due to the variation in the level of sophistication and features presented in the bot code
- The common thing about bots is that attackers are eager to integrate new software vulnerabilities in their bot code very quickly. This means that bots will continue to evolve in an unpredictable manner

Botnets- An Overview (Contd.)

Table 1. New bot variants by month.

MONTH	AGOBOT	SDBOT
May 2004	543	332
June 2004	249	654
July 2004	339	1018
August 2004	133	977
September 2004	123	818
October 2004	158	1111
November 2004	113	1156
December 2004	196	1637
January 2005	227	1539
February 2005	97	2010
March 2005	200	1689

Source: [T. Holz. A short visit to the bot zoo. *IEEE Security & Privacy*, 3(3):76–79, 2005]

References

- VOGT, R., AYCOCK, J., and JACOBSON, M., “Army of botnets,” in Proceedings of the 14th Network and Distributed System Security Symposium (NDSS’07), 2007
- ZOU, C. C. and CUNNINGHAM, R., “Honeypot-aware advanced botnet construction and maintenance,” in International Conference on Dependable Systems and Networks (DSN’06), 2006
- R. Clarke, “Looking of Vulnerability Issues in Cyber-Security,” Business Session of the President’s National Security Telecommunications Advisory Committee (NSTAC), Mar. 2002.
- A. Chakrabarti and G. Manimaran, “Internet Infrastructure Security: A Taxonomy,” IEEE Network, vol.16, no.6, pp.13-21, Nov/Dec. 2002.
- A. Householder, K. Houle, and C. Dougherty, “Computer attack trends: Challenge Internet security,” Security and Privacy – 2002, supplement to IEEE Computer, Jan. 2002.

References (Contd.)

- D. Eastlake, “Domain name system security extensions,” RFC 2535, Mar. 1999.
- P. Jungck and S. S.Y. Shim, “Issues in high-speed Internet security,” IEEE Computer, Jul 2004.
- P. Papadimitratos and Z.J Haas, “Securing the Internet routing infrastructure,” IEEE Communications, vol. 40, no. 10, pp. 60-80, Oct. 2002.
- J. F. Kurose and K. W. Ross, “Computer Networking: A Top-Down Approach Featuring the Internet,” Pearson Addison-Wesley, 2002.
- A. S. Tanenbaum, “Computer Networks,” Prentice Hall, 4th edition, 2002.

Module II:
Botnet Formation

Talk Outline – Module II

- **Botnet-life time**
- IRC-Based Botnets
- P2P- Botnets
- New Trends in Botnet Design

[P. Wang et.al., A systematic Study on Peer to Peer Botnets, IEEE ICCCN 2009]

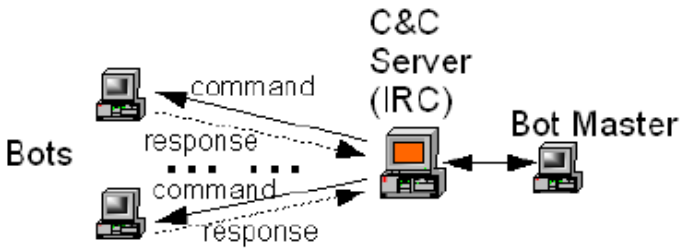
Botnet Lifetime

- Stage one: recruiting members, a botmaster needs to compromise many computers in the Internet, so that he/she can control them remotely
- Stage two: forming the botnet, bots need to find a way to connect to each other and form a botnet
  - **The C&C plane** where bots receive commands from the botmaster
- Stage three: standing by for instructions, after the botnet is built up, all bots are ready to communicate with their botmaster for further instructions, such as launching an attack or performing an update
  - **The activity plane** where bots execute these commands to launch different types of attacks that include DDoS, spam, click fraud, etc

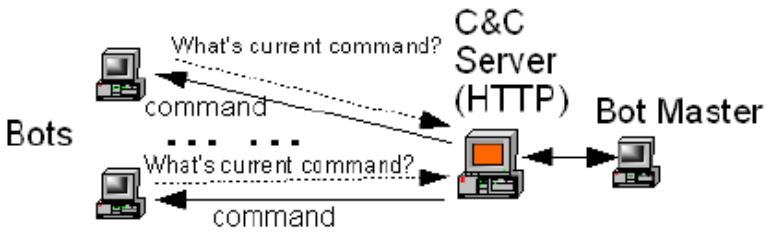
Botnets- C&C

- Push style: Bots passively wait for commands to come and will forward received commands to others

- Pull style: refers to the manner that bots retrieve commands actively from a place where botmasters publish commands



(I) C&C: Push style



(II) C&C: Pull style

[Source: G. Gu. et. al., NDSS 2008]

Botnets C&C

- The structure of a botnet is basically determined by its C&C plane topology which in turn specifies the way botmaster delivers commands to botnet members.
- C&C is usually implemented using one of the following protocols:
  - IRC (Centralized)
  - HTTP (Centralized)
  - Email (Centralized)
  - P2P (Distributed)

Selective well known Botnets

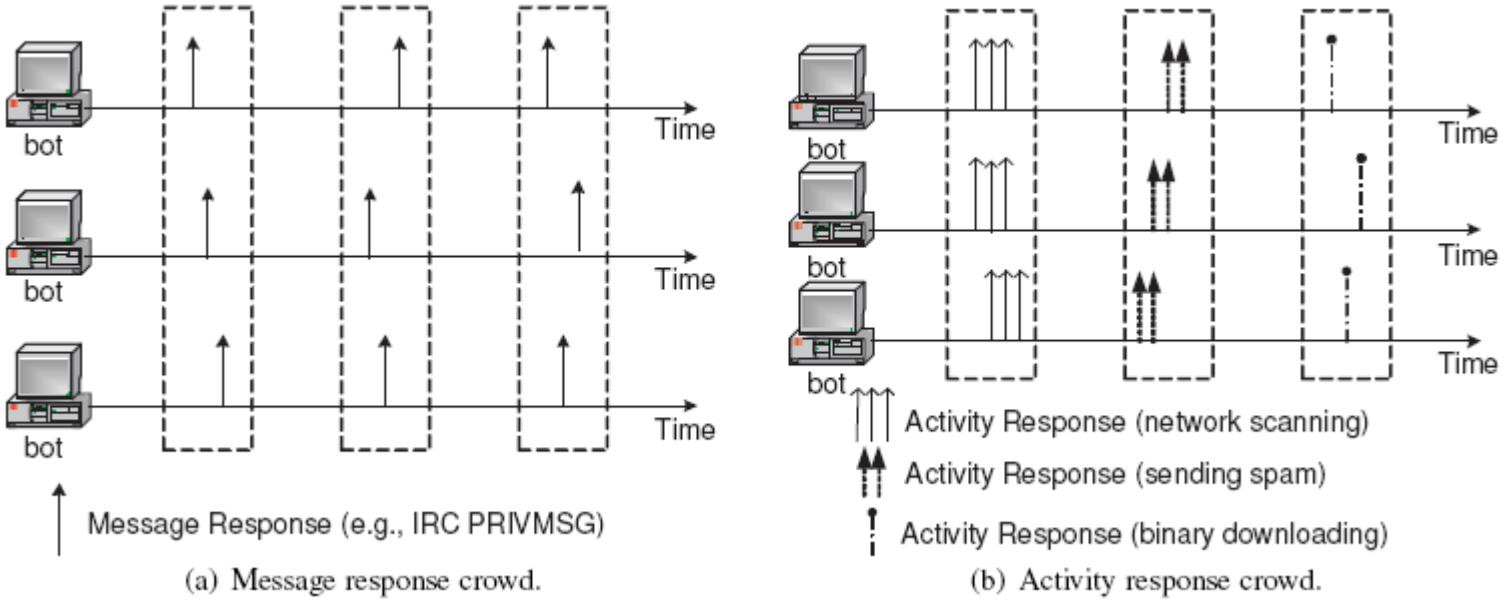
Date	Name	C&C Protocol	Structure	Distinguishing Description
04/1998	GTbot	IRC	Centralized	First widely spreading IRC bot using mIRC executables and scripts
04/2002	SDbot	IRC	Centralized	First stand-alone and open-source IRC bot
10/2002	Agobot	IRC	Centralized	Very robust, flexible, and modular design
04/2003	Spybot	IRC	Centralized	Extensive feature set based on Agobot
2004	Rbot/rxbot	IRC	Centralized	SDbot descendant, code base wildly distributed
03/2004	Phatbot	WASTE	P2P	Experimental P2P bot using WASTE protocol
05/2004	Bobax	HTTP	Centralized	First well-known spambot using HTTP as C&C
04/2006	Nugache	Self-defined	P2P	First "practical" P2P bot connecting to predefined peers
01/2007	Storm	Kademlia	P2P	Famous large-scale P2P botnet mainly used to send spam
04/2008	Kraken	Self-defined	Centralized	Large botnet penetrating into at least 50 of the Fortune 500 companies

[Source: Goufie Gu, PhD Thesis, 2008]

[G. Gu. et. al., NDSS 2008]

Botnet C&C: Spatial-Temporal Correlation and Similarity

- Bots of a botnet demonstrate spatial-temporal correlation and similarities due to the nature of their pre-programmed response activities to control commands
- Bots need to connect to C&C servers in order to obtain commands
  - They may either keep a long connection or frequently connect back
- Second, bots need to perform certain tasks and respond to the received commands



[Source: G. Gu. et. al., NDSS 2008]

Talk Outline – Module II

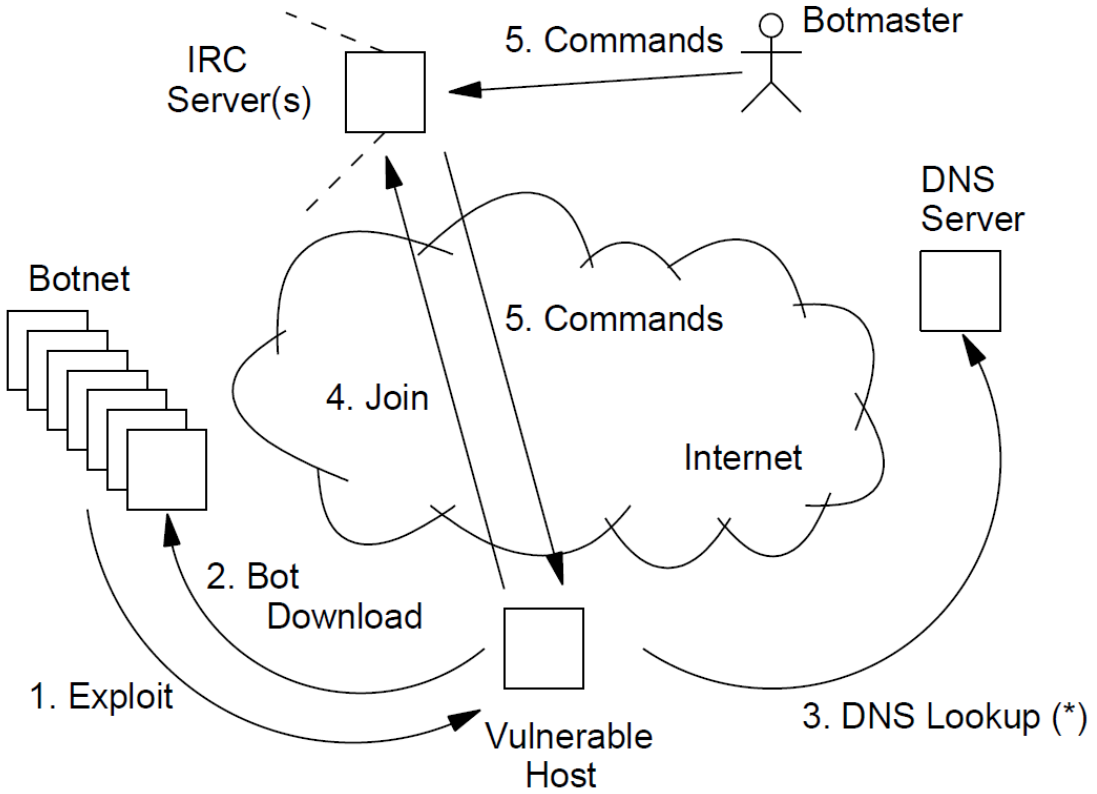
- Botnet-life time
- **IRC-Based Botnets**
- P2P- Botnets
- New Trends in Botnet Design

[M. A. Rajab, et. al , IMC 2006]

IRC-Based Botnets

- The majority of botnets today use the Internet Relay Chat (IRC) protocol
- The IRC protocol was specifically designed to allow for several forms of communication (point-to-point, point to multi-point, etc.) and data dissemination among large number of end-hosts.
- What features make IRC the protocol of choice for botmasters?
  - The inherent flexibility of this protocol
  - The availability of several open-source implementations, enables third parties to extend it in ways that suit their needs
  - It simplifies the botnet implementation and provides a high degree of control over the bots

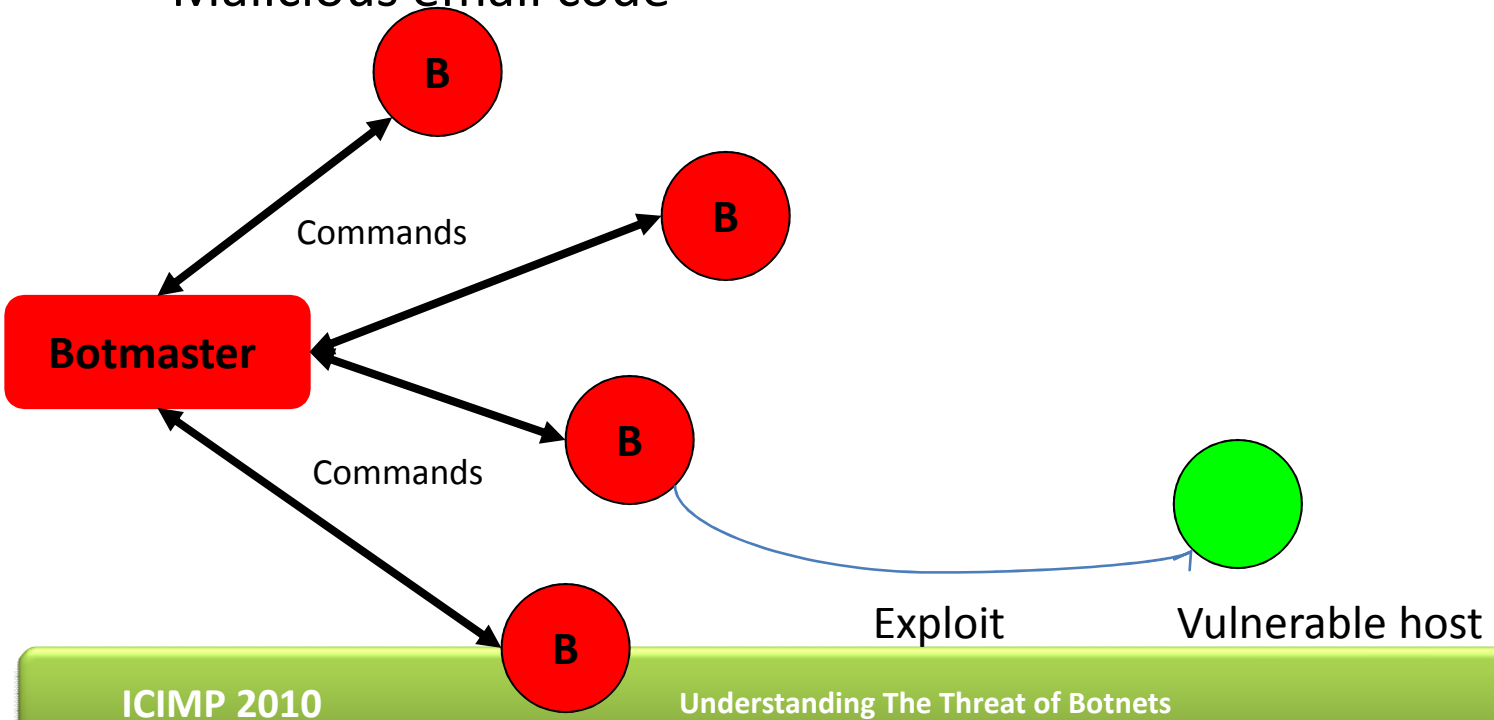
IRC-based Botnet Life Cycle



[Source: M. A. Rajab, et. Al , In IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement. pp. 41–52. 2006]

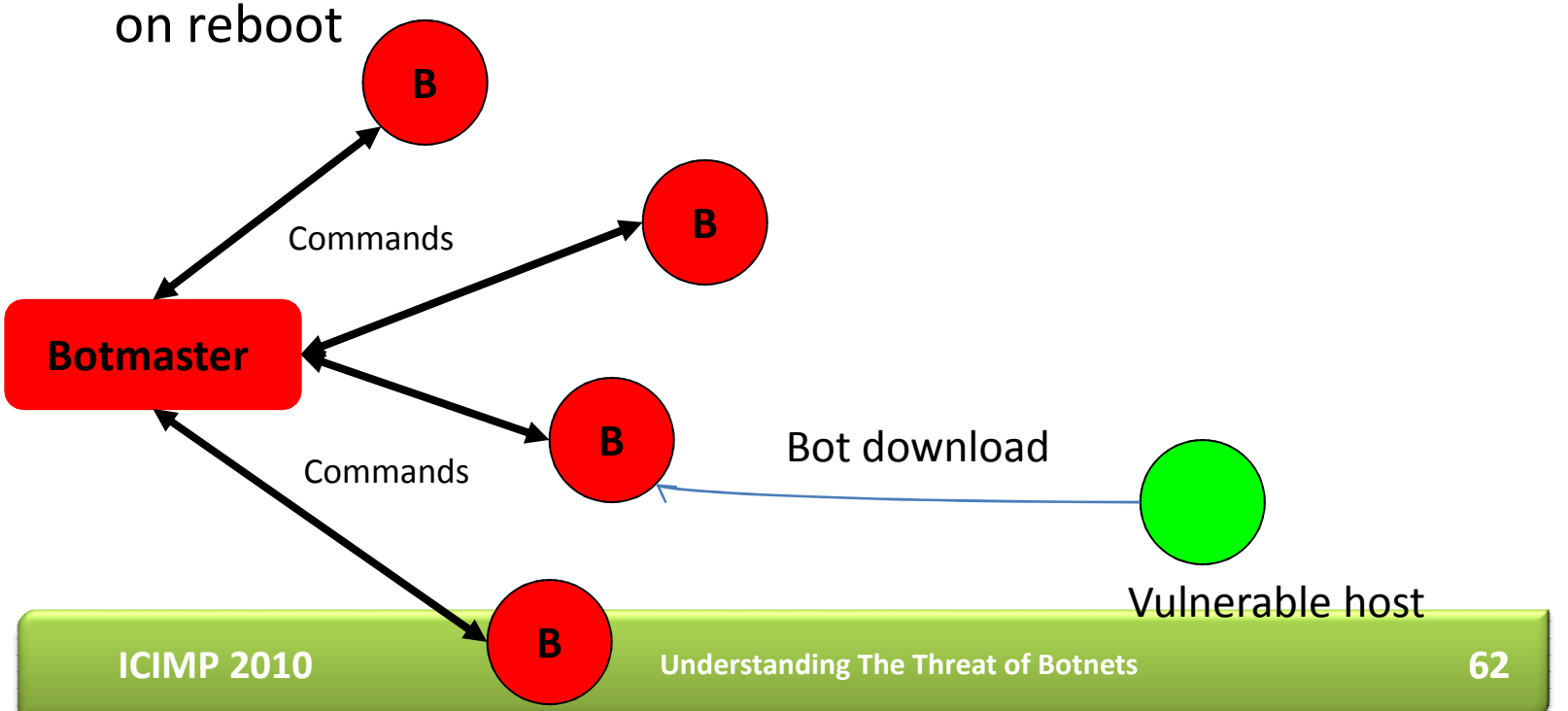
Step 1: Exploit

- Exploit software vulnerability of victim host
- Same infection strategies as other malware
  - Worms
  - Malicious email code



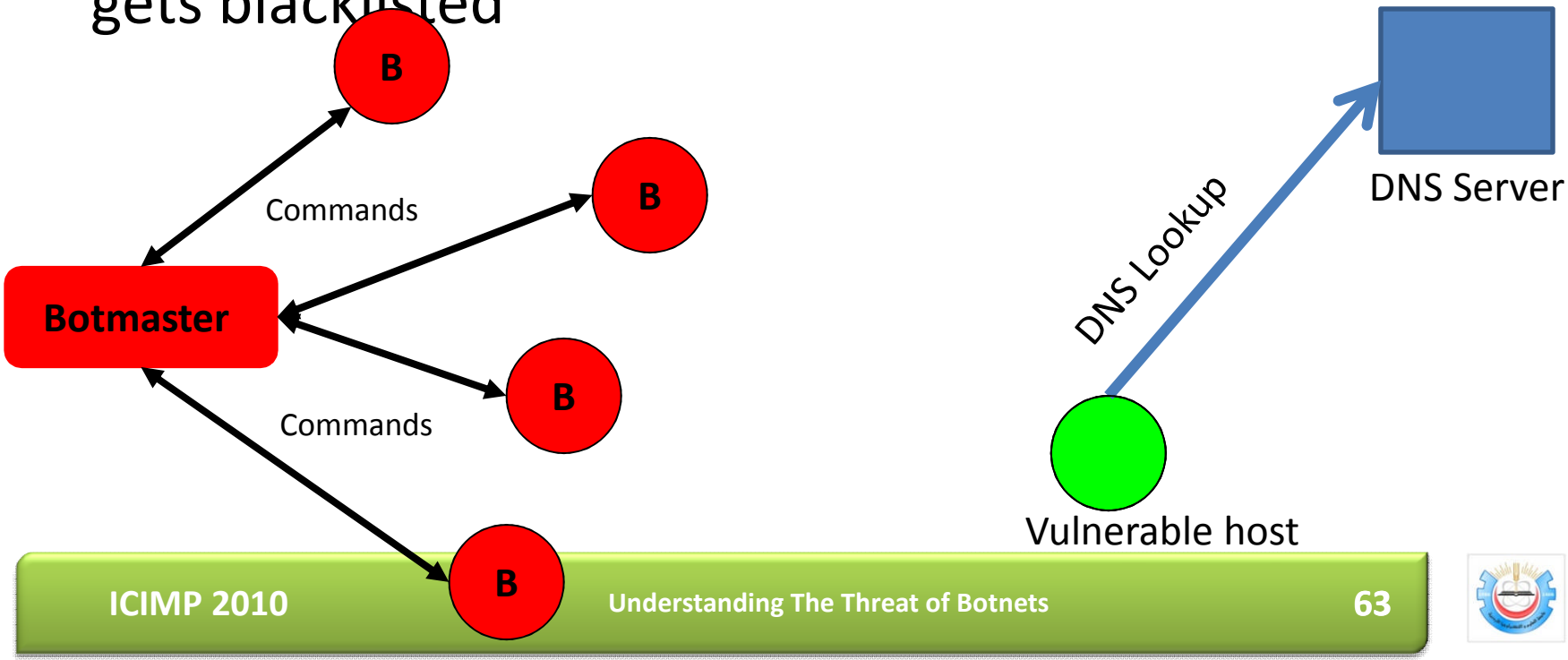
Step 2: Download Bot Binary

- Infected host executes shellcode to fetch bot binary from specified location
  - Usually the same machine that infected it
- After the download, the bot binary installs itself so it can auto start on reboot



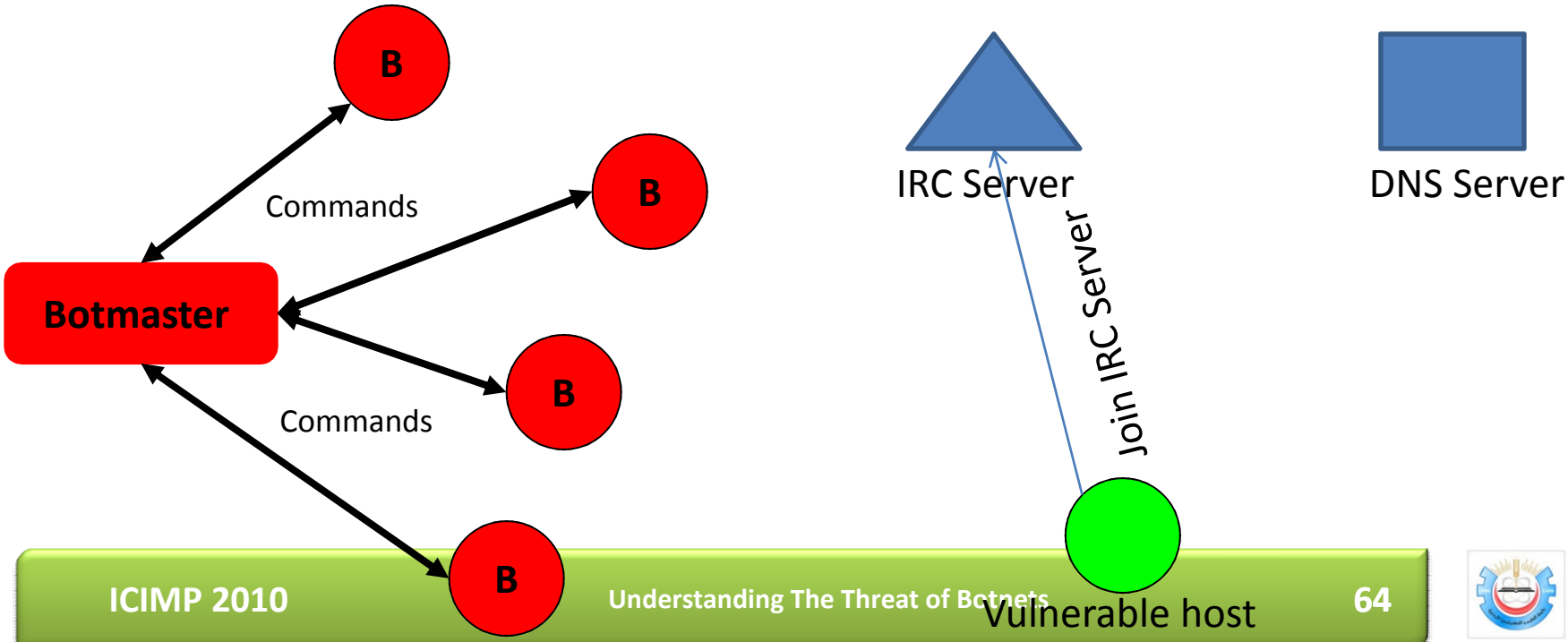
Step 3. DNS lookup

- Bot needs IP address of IRC server
- Perform DNS Lookup
- Better than hard-coding the server IP in case the IP gets blacklisted



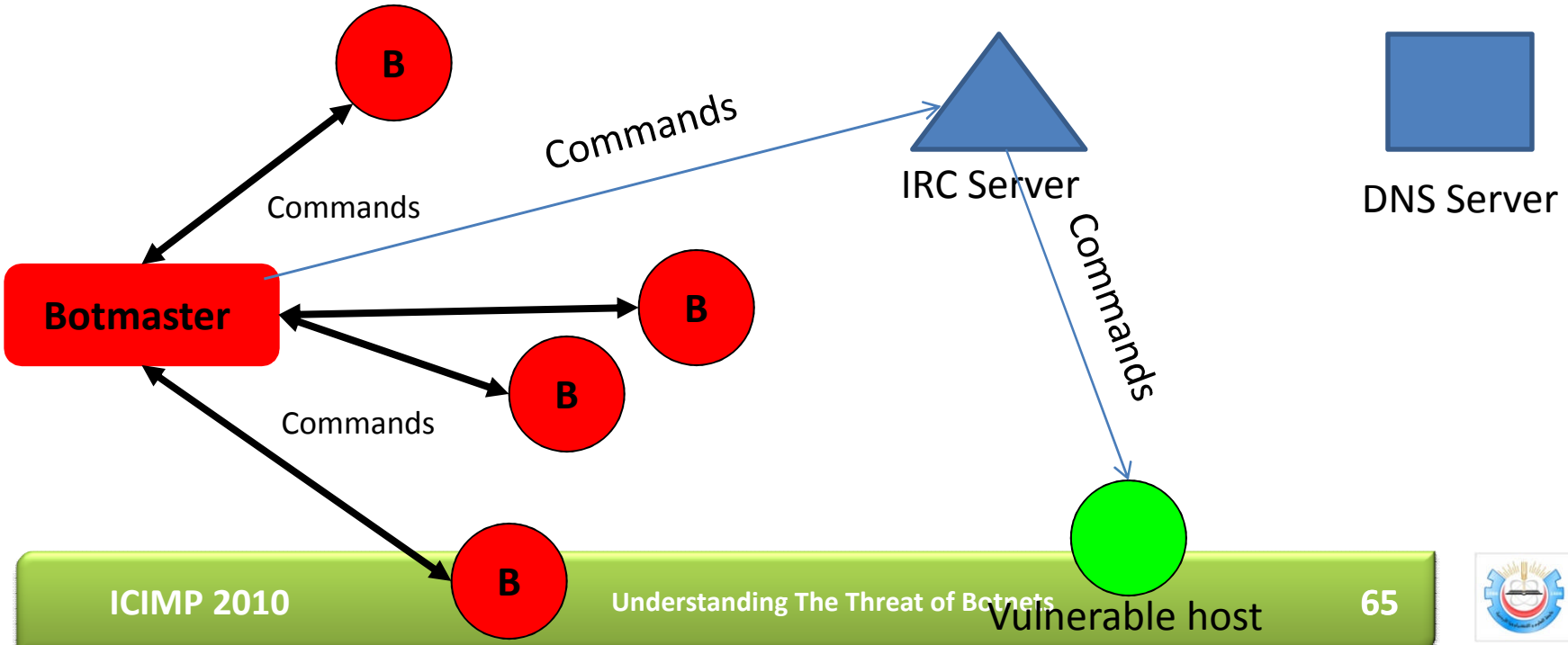
Step 4: Join IRC Server

- Join server and channel specified in bot binary
- May use authentication:
  1) Bot authenticates to join server using password from bot binary
  2) Bot authenticates to join channel using password from bot binary
  3) Botmaster authenticates to bot population to send command

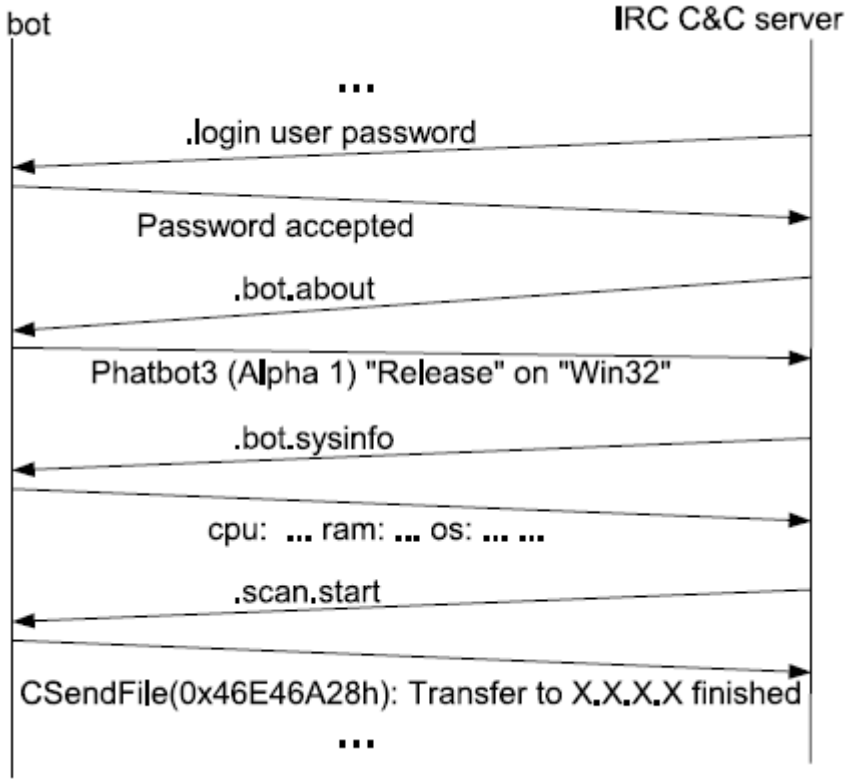


Step 5: Execute Commands

- Bot parses and executes channel topic
- Topic contains default command for all bots to execute



IRC-Based Communication Example



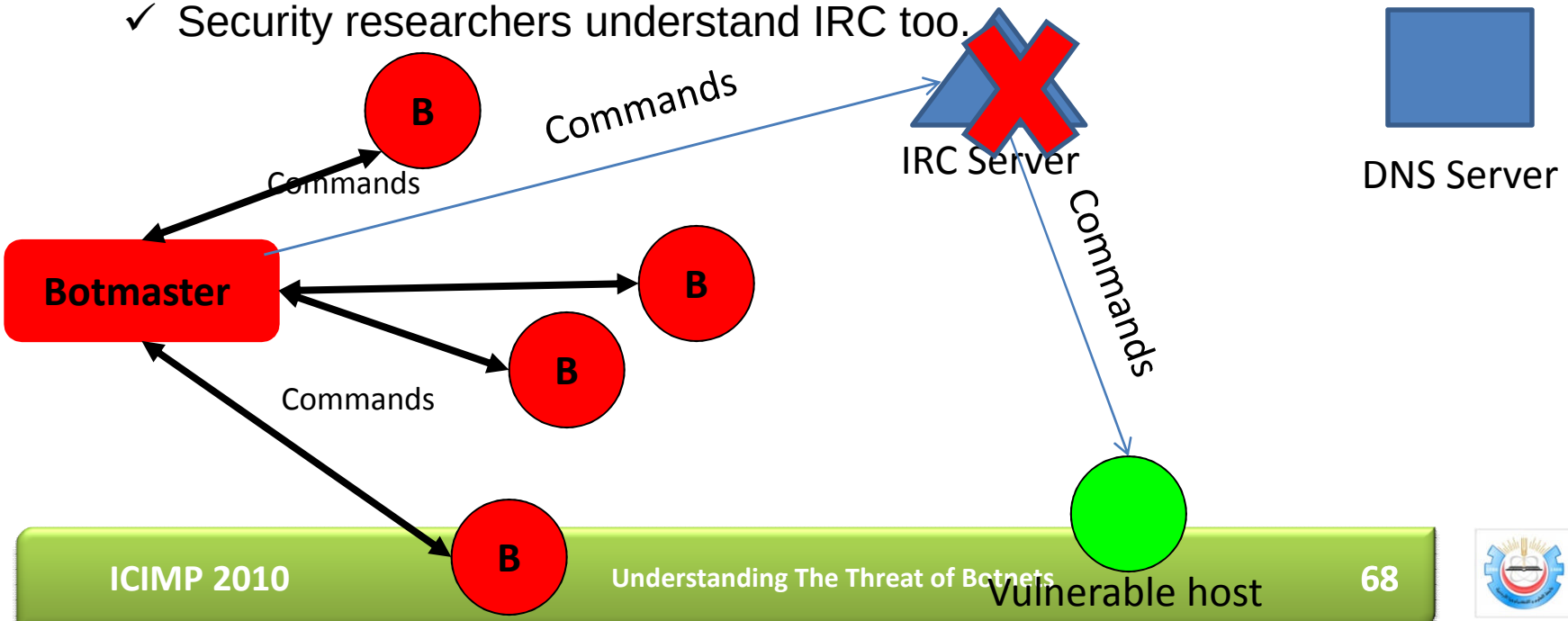
[G. Gu. et. al., NDSS 2008]

Difficulties in Detecting Centeralized Botnets

- Botnet C&C traffic is difficult to detect because:
  - It follows normal protocol usage and is similar to normal traffic
  - The traffic volume is low
  - There may be very few bots in the monitored network
  - It may contain encrypted communication

IRC Botnets (Contd.)

- Botherders are migrating away from IRC botnets because researchers know how to track them.
- Drawbacks:
  - ✓ Centralized server
  - ✓ IRC is not that secure by default
  - ✓ Security researchers understand IRC too.

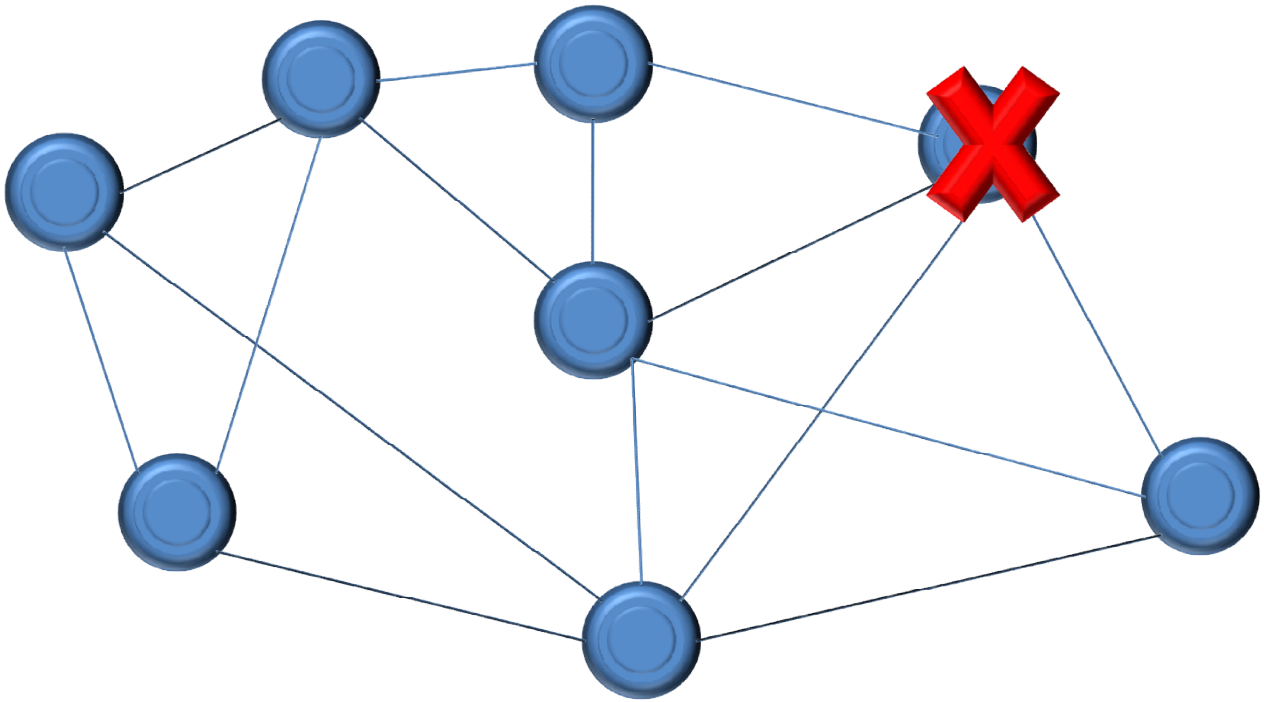


Talk Outline – Module II

- Botnet-life time
- IRC-Based Botnets
- **P2P- Botnets**
- New Trends in Botnet Design

P2P Botnets

- Distributed Control
- Hard to disable



[P. Wang et.al., IEEE ICCCN 2009]

P2P Botnets

- P2P Botnets are classified into:
  - Parasite: All the bots are selected from hosts within an existing P2P network → use this network for C&C
  - Leeching: All the bots join an existing P2P network → it uses this available P2P network for C&C
  - Bot-only: All the members are bots (e.g., Stormnet, Nugache)→ A P2P network has to be formed

Forming a P2P Network

- Current P2P networks provide the following ways for new peers to join the network (bootstrapping)
  - An initial peer list is hard-coded in each P2P client.
  - There is a shared web cache stored somewhere on the Internet and the location of the cache is put in the client code
- These methods can be adopted for P2P botnet construction (eg., Trojan.Peacomm, Stormnet)

P2P-botnets- Standing by for instructions

- Leveraging existing P2P protocols
  - Usually use pull mechanism
  - Eg., Storm botnet utilizes Overnet
- Designing new P2P protocols
  - Can use push/pull mechanisms
  - Eg., Avanced Hybrid P2P botnet [C. C. Zou. et. al., DSN 2006], Super botnet [R. Vogt. et. al., NDSS 2007].

Case Study: Storm Botnet

- P2P network architecture
- Content-based publish/subscribe- style communication
  - An information provider publishes a piece of information i, e.g., a file, using an identifier which is derived solely from i.
  - An information consumer can then subscribe to certain information using a filter on such identifiers
- Unauthenticated communication: Content providers do not authenticate information
  - Authentication is usually implicit: If the information received by a peer matches its subscription, then it is assumed to be correct

[T. Holz, et.al., LEET 2008]

Storm Botnet- Propagation Mechanism

- Propagates using email
- The attackers behind storm change the social engineering quite often
- Storm exploits web browsers with specific User-Agent
- The actual exploit code in the malicious websites is polymorphic
- The binary itself shows signs of polymorphism

Storm Botnet- System Level Behavior

- Storm is sophisticated
  - Uses an advance binary packer
  - Uses a rootkit to hide its presence
  - Uses kernel level components to remain undetected
- During the installation process, the malware also stores a configuration file on the infected system
- Storm synchronizes the system time of the infected machine with the help of the Network Time Protocol (NTP)

Storm Botnet- Network Level Behavior

- The first version of Storm Worm uses OVERNET
  - Kademila-based P2P DHT routing protocol
- Stormnet- New version in October 2007
  - Identical to Overnet except
  - Each message is XOR encrypted with a 40-byte long key
  - Each node has 128-bit ID

Storm Botnet- Network Level Behavior- Routing Lookup

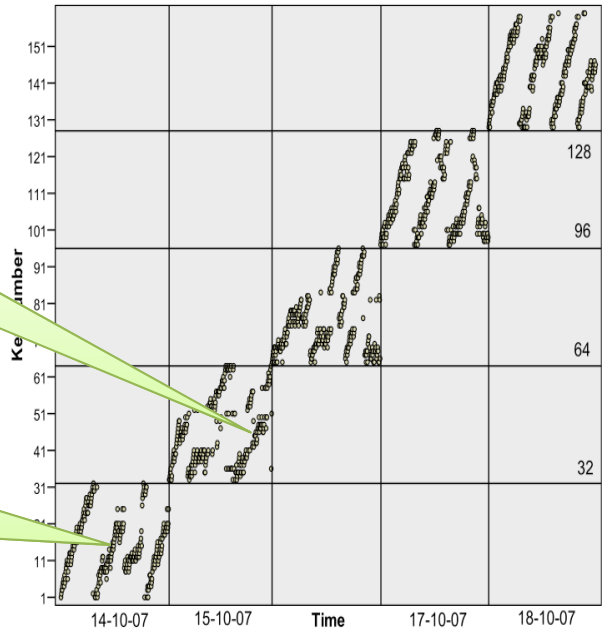
- A node a forwards a query destined to a node *d* to the node in its routing table that has the smallest XOR-distance with *d*
- The XOR-distance d(a, b) between nodes a and b is d(a, b) = a⊕b
- Prefix matching, looks for smallest XOR distance between destination and contacts it has
- Contacts: ID, IP, UDP port
- Iterative lookups. Queries closest node for ID and repeats until returned ID is further away than ID queried

Storm Botnet- Network Level Behavior- Publishing and Searching

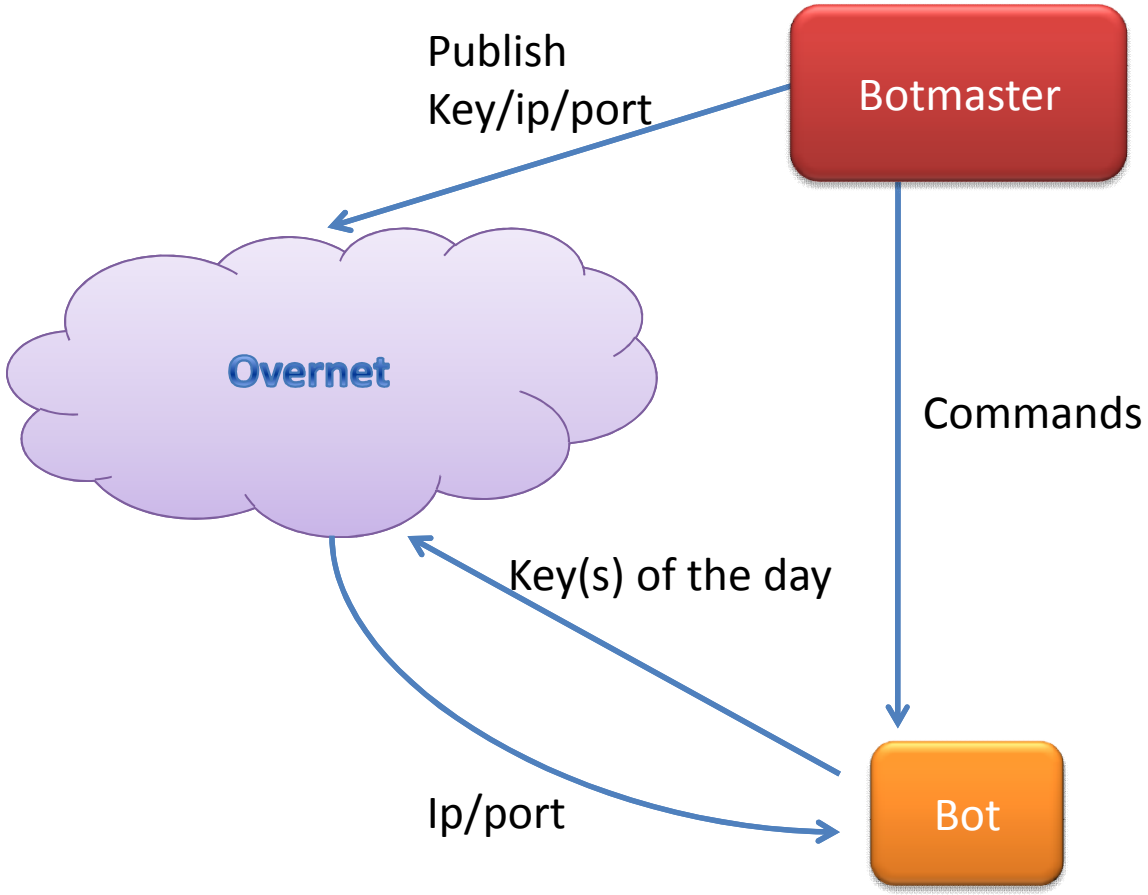
- Publishing and Searching
  - A “key” is an identifier used to retrieve information
  - Keys are stored by 20 nodes close to the key
  - Publisher periodically republishes keys
  - Botmaster publishes to a list of well known “mailboxes”
  - Each new bot looks for those mailboxes and retrieves the intended information
- Message types:
  - Hello
  - Kid (KeyID)Route request/response
  - Publish request/response
  - Key Search request/response

Storm Network

- Storm Botnet Communication
  - Looks for peer by searching for keys
  - Key = f(day, rand), rand is a 5 bit number
- Keys can by identified through:
  - Reverse Engineering
  - Black box testing



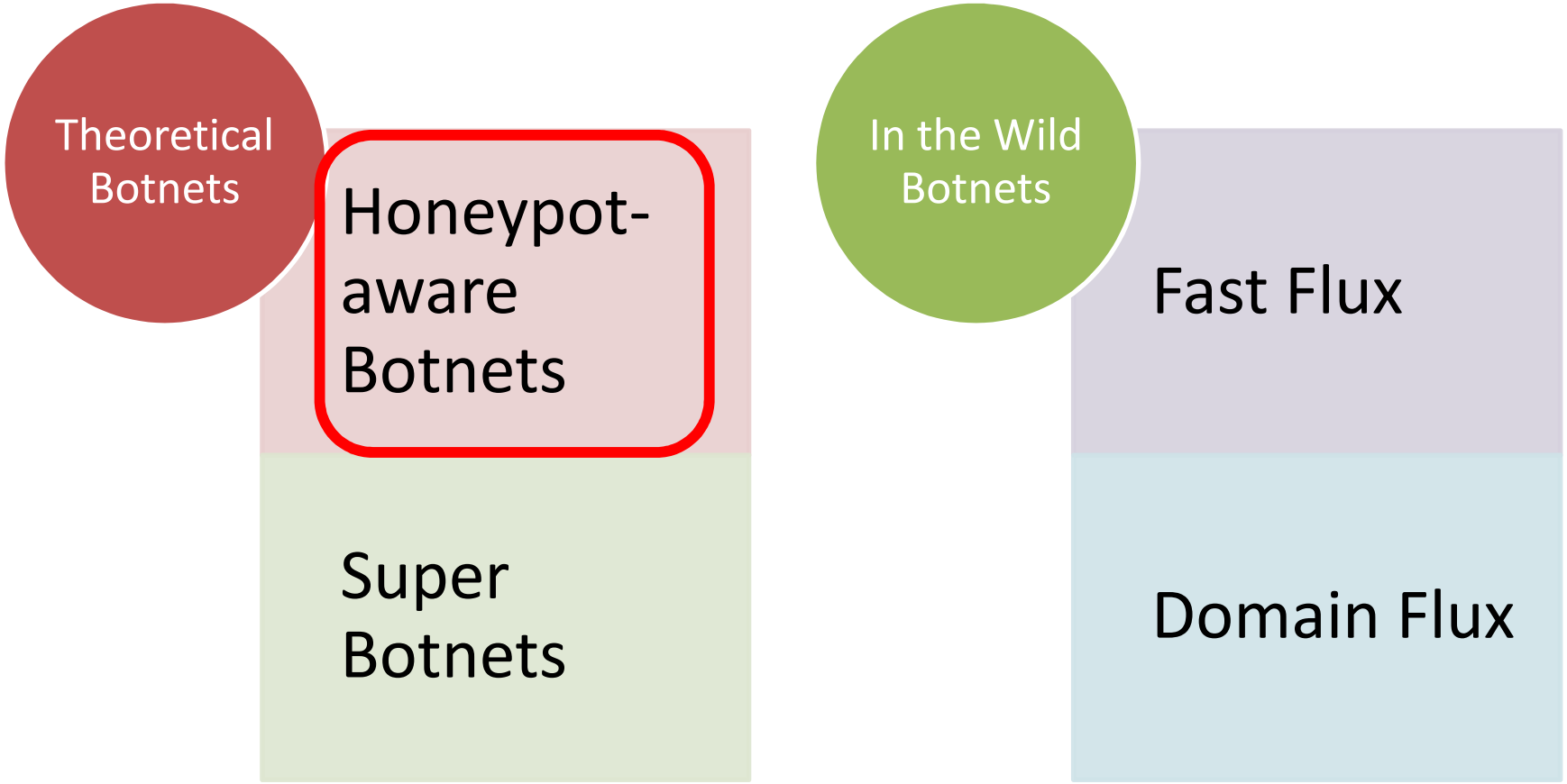
Storm network



Talk Outline – Module II

- Botnet-life time
- IRC-Based Botnets
- P2P- Botnets
- **New Trends in the Design of Botnets**

Botnets- New Trends

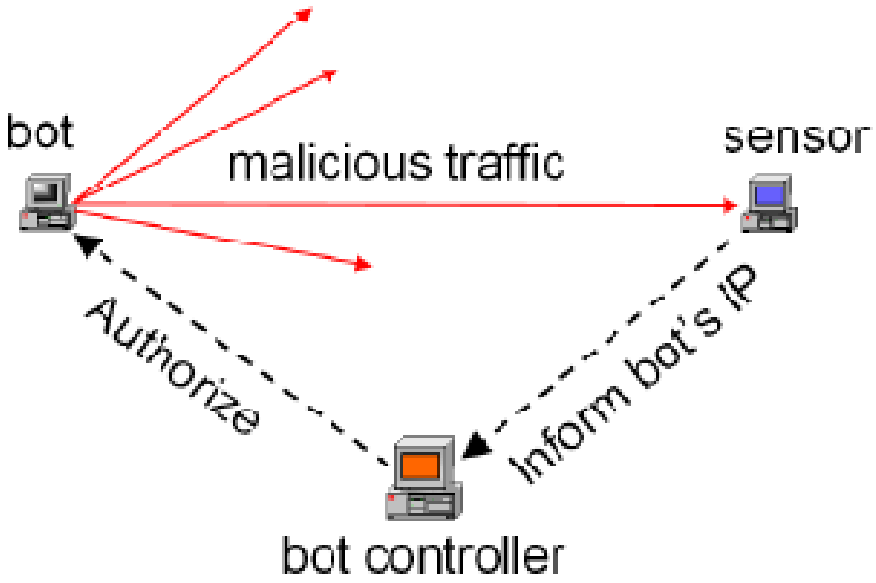


[C. C. Zou. et. al., DSN 2006]

Honeypot-Aware Botnet Construction Mechanism

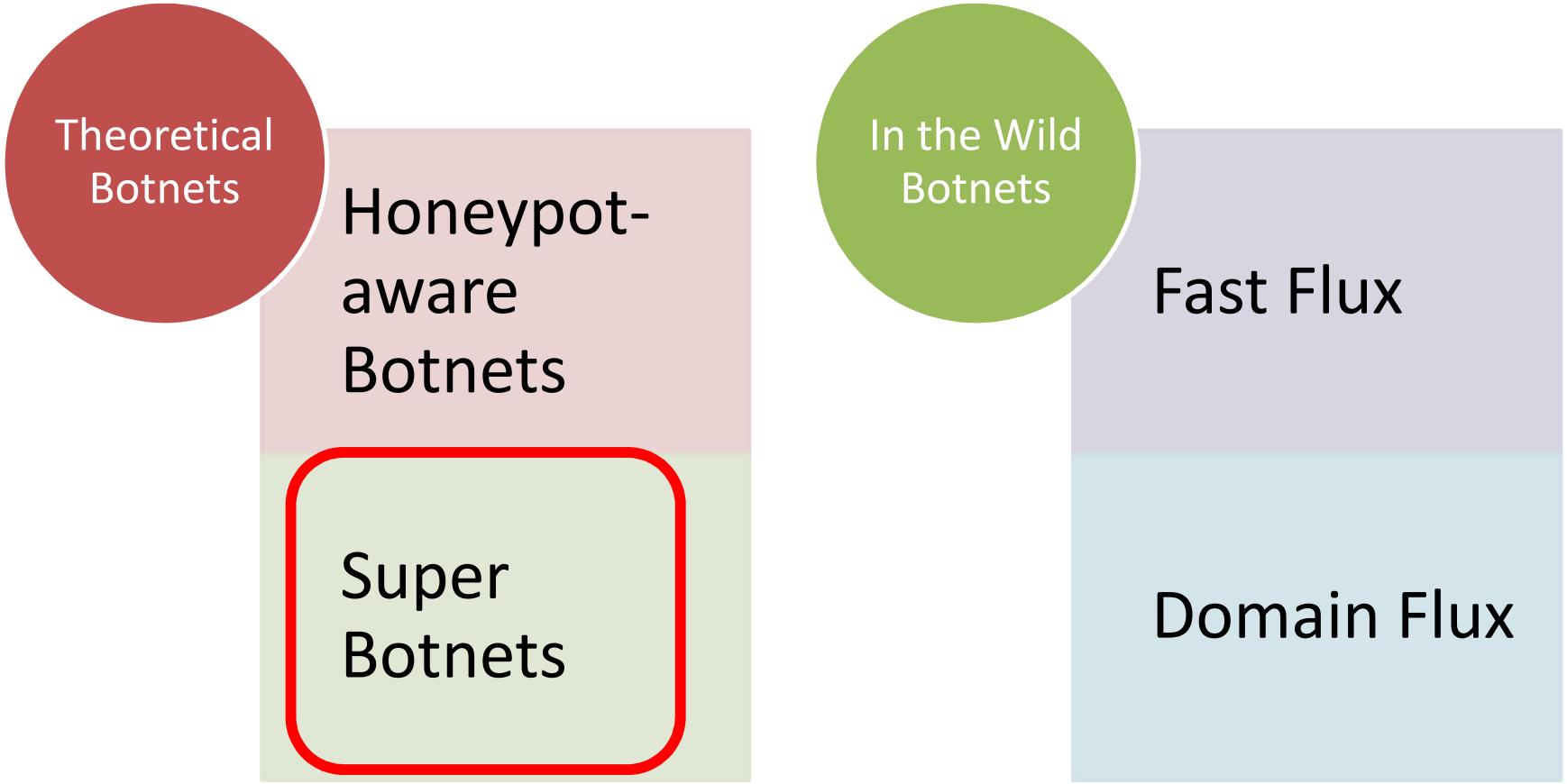
- Attackers can thwart botnet trapping techniques
- The general principle is to have an infected computer send out certain malicious or “faked” malicious traffic to one or several remote computers that are actually controlled by the botnet attacker
- These remote computers behave as “sensors” for the attacker
- If the sensors receive the “complete” and “correct” traffic from the infected host, then the host is considered “trusted” and is treated as a normal bot instead of a honeypot

Honeypot-Aware Botnet Construction Mechanism (Contd.)



[Source: C. C. Zou. et. al., DSN 2006]

Botnets- New Trends



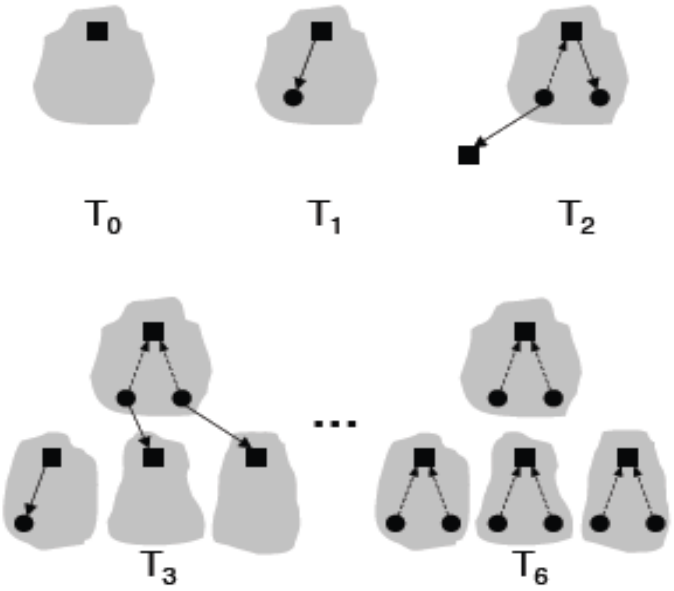
[R. Vogt. et. al., NDSS 2007]

Super Botnets

- An adversary can create a large number of small, independent botnets.
- By themselves, the smaller botnets can be exploited by the adversary in the usual way, such as being rented to spammers
- The botnets can be designed to be coordinated into a network of botnets →super-botnet
- A tree structured algorithm can be used to construct the super botnet

Super Botnets (Contd.)

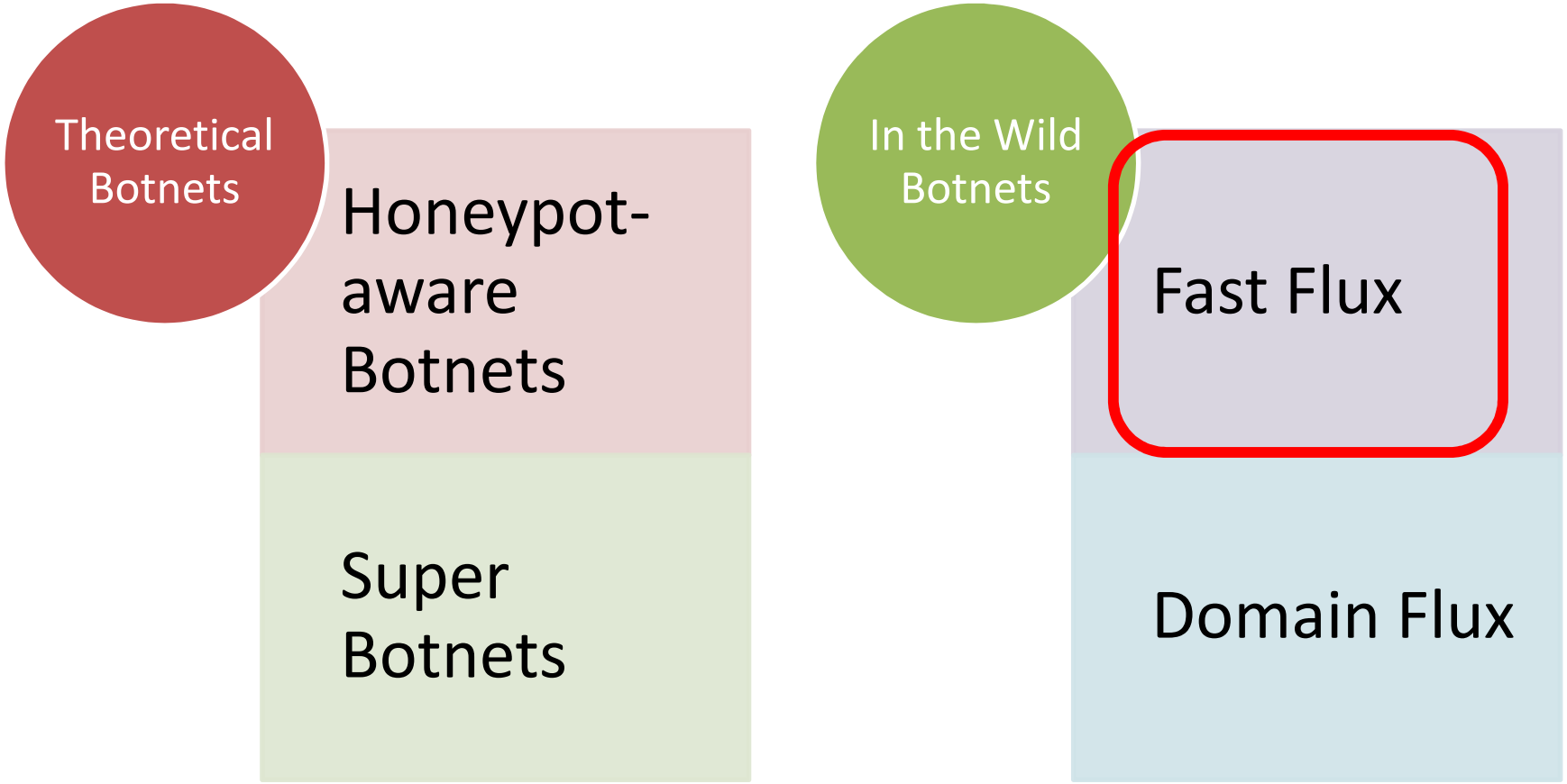
- This algorithm creates BOTNETS individual botnets, each consisting of HOSTS_PER_BOTNET zombies
- Each zombie infects at most SPREAD new hosts to bring the size of its botnet up to HOSTS_PER_BOTNET)
- If a zombie is not a C&C machine for a new botnet, it also learns the location of its botnet's C&C server.



Worm infection pattern with 3 hosts/botnet, 4 botnets, and a spread of 2. Shown are C&C servers (■), non-C&C infected machines (•), new infections (solid arrows), links to C&C servers (dashed arrows), and botnets formed (grey blobs).

[Source: R. Vogt. et. al., NDSS 2007]

Botnets- New Trends

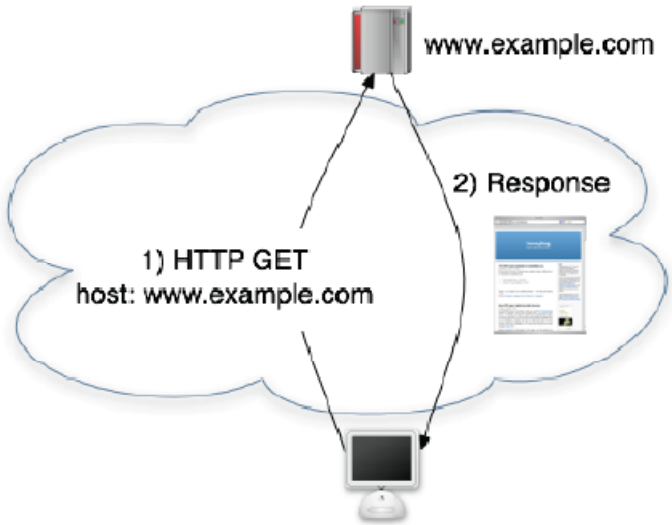


Continuous Availability- Legal Perspective

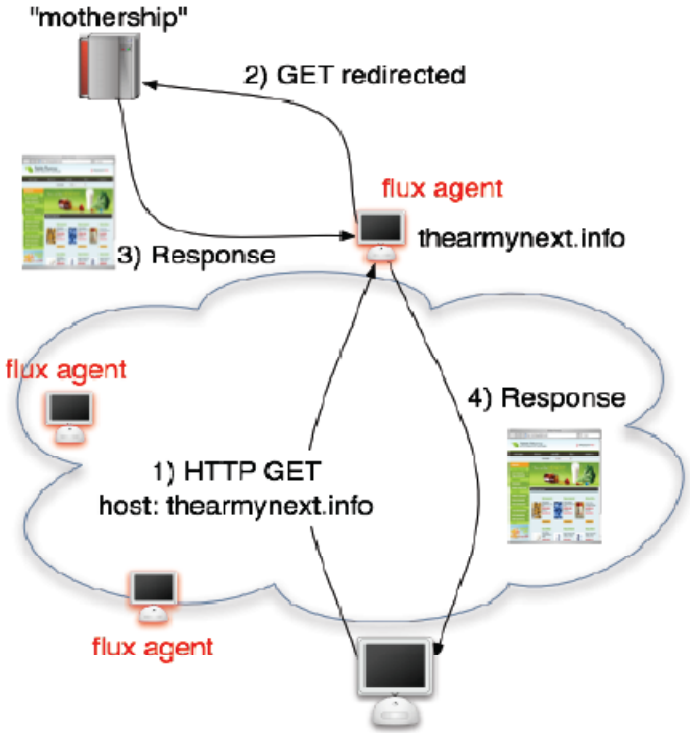
- If web servers are not online, the service can not be offered, resulting in loss of profit
- Problem
  - Hardware failures
  - Distributed Denial of Service Attacks
- Solution
  - Round Robin DNS
    - Distribute the load of incoming requests to several servers
  - Content Distribution DNS
    - Finds nearest server, and resolve to that instead of hitting the central servers

Continuous Availability- Illegal Perspective

- Examples:
  - A spammer who run a website to sell pharmaceutical products, adult content, etc.
  - A phisher who runs a web site to steal sensetive information from victims
  - A botherder who runs a website to direct large botnet
- Problem: These websites are subject to blocking or attack by defenders
- Solution: Provide service resilience thruogh fast flux networks



Content retrieval process for benign HTTP server



Content retrieval process for content being hosted in fast-flux service network

[source: T. Holz. et. al., NDSS 2008]

Fast Flux Example

```
;; ANSWER SECTION:
thearmynext.info. 600 IN A 69.183.26.53
thearmynext.info. 600 IN A 76.205.234.131
thearmynext.info. 600 IN A 85.177.96.105
thearmynext.info. 600 IN A 217.129.178.138
thearmynext.info. 600 IN A 24.98.252.230

;; ANSWER SECTION:
thearmynext.info. 600 IN A 213.47.148.82
thearmynext.info. 600 IN A 213.91.251.16
thearmynext.info. 600 IN A 69.183.207.99
thearmynext.info. 600 IN A 91.148.168.92
thearmynext.info. 600 IN A 195.38.60.79
```

IP address returned in A record	Reverse DNS lookup for IP address	ASN	Country
69.183.26.53	69.183.26.53.adsl.snet.net.	7132	US
76.205.234.131	adsl-76-205-234-131.dsl.hstntx.sbcglobal.net.	7132	US
85.177.96.105	e177096105.adsl.alicedsl.de.	13184	DE
217.129.178.138	ac-217-129-178-138.netvisao.pt.	13156	PT
24.98.252.230	c-24-98-252-230.hsd1.ga.comcast.net.	7725	US

Fast Flux DNS

- BotHerders interested in reliability reuse ideas from RRDNS and CDN

- As long as a single IP responds, the entire service is online
- Fast Flux: Fast change in DNS answers
  - Return only a subset of IP addresses from available pool
  - Return different subset after TTL expires

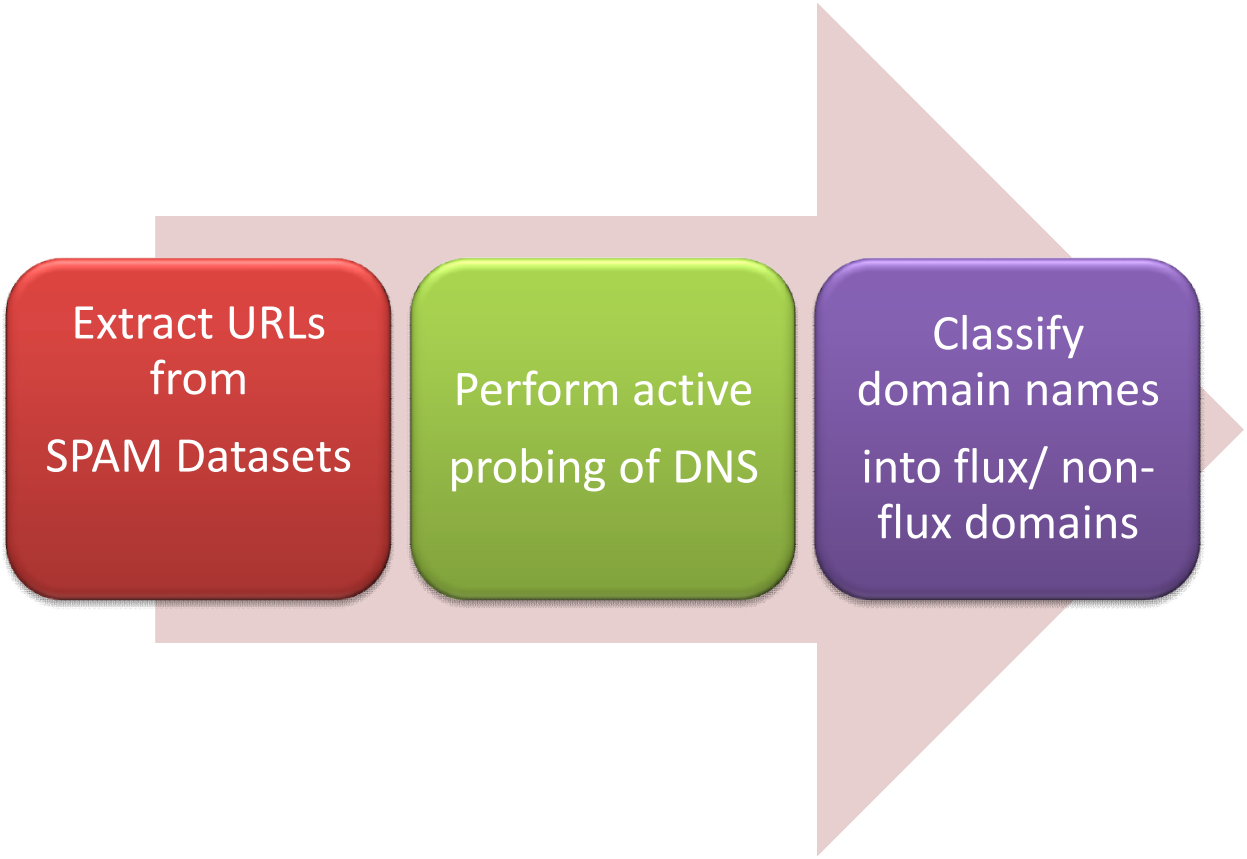
FFN Characteristics

- Short time-to-live (TTL)
- The set of resolved IPs (i.e., the flux agents) returned at each query changes rapidly, usually after every TTL
- The overall set of resolved IPs obtained by querying the same domain name over time is often very large
- The resolved IPs are scattered across many different networks

Research in the area of FFNs

FFN Detection	• Active Approach • Passive Approach
FFN Characterization	• Similarity of Scam • Rate of Change • Rate of Accumulation • Sharing across campaigns

FFN Detection- Active Approach



Step 1. Extract Domain Names from SPAM Datasets

```
Delivered-To: em-ca-bruceg@em.ca
Received: (qmail 31828 invoked from network); 1 Nov 2009 04:52:06 -0000
Received: from na3nkfn (ppp-202-176-138-59.revip.asianet.co.th [202.176.138.59])
  by churchill.factcomp.com ([24.89.90.248])
  with SMTP via TCP; 01 Nov 2009 04:52:06 -0000
Message-ID: <000701ca5aaf$5ad3b140$ae78a462@chc.net.au>
Reply-To: "Fritz Burris" <fritz.burrisrb@chc.net.au>
From: "Fritz Burris" <fritz.burrisrb@chc.net.au>
To: <bruceg@em.ca>
Subject: Cheapest Medications on the PLanet!
Date: Sat, 31 Oct 2009 23:54:12 -0500
MIME-Version: 1.0
Content-Type: text/plain;
      format=flowed;
      charset="windows-1250"
      reply-type=original
Content-Transfer-Encoding: 7bit
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Office Outlook, Build 11.0.5510
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2800.1807

Order Pharmacy Medications_Online!
No Doctor Needed!
Browse Our Selection Today! -> http://rxthatbeatsallothers.com
```


Step2. Perform Active Probing of DNS

- 1 ~$ dig eventdraw . com
- 2
- 3 ; <<>> DiG 9.4.2--P1 <<>> eventdraw . com
- 4 ; ; g l o b a l opt i ons : printcmd
- 5 rxthatbeatsallothers. com. 120 IN A 2 0 3 . 1 8 6 . 2 3 4 . 1 0 9
- 6 rxthatbeatsallothers. com. 120 IN A 2 1 0 . 6 . 1 0 3 . 8
- 7 rxthatbeatsallothers. com. 120 IN A 2 1 9 . 2 4 0 . 7 9 . 5 8
- 8 rxthatbeatsallothers. com. 120 IN A 2 2 1 . 1 2 7 . 2 . 2 4 3
- 9 rxthatbeatsallothers. com. 120 IN A 2 2 1 . 1 4 5 . 7 2 . 8 1
- 10 rxthatbeatsallothers. com. 120 IN A 2 4 . 1 1 5 . 3 3 . 2 1 0

Step 3. Classify Domain Names into FF/ Non FF

- FFSN restrictions
  - IP diversity
  - No physical flux agent control, no uptime guarantee
- Possible distinguishing parameters
  - N_A, Number of unique A records in all DNS lookups (the entire pool)
  - N_{NS},Number of nameserver records in one single lookup
  - N_{ASN},Number of unique ASNs for all A records
  - TTL not considered. Legit sites can have low TTLs

Step 3 (Contd.)

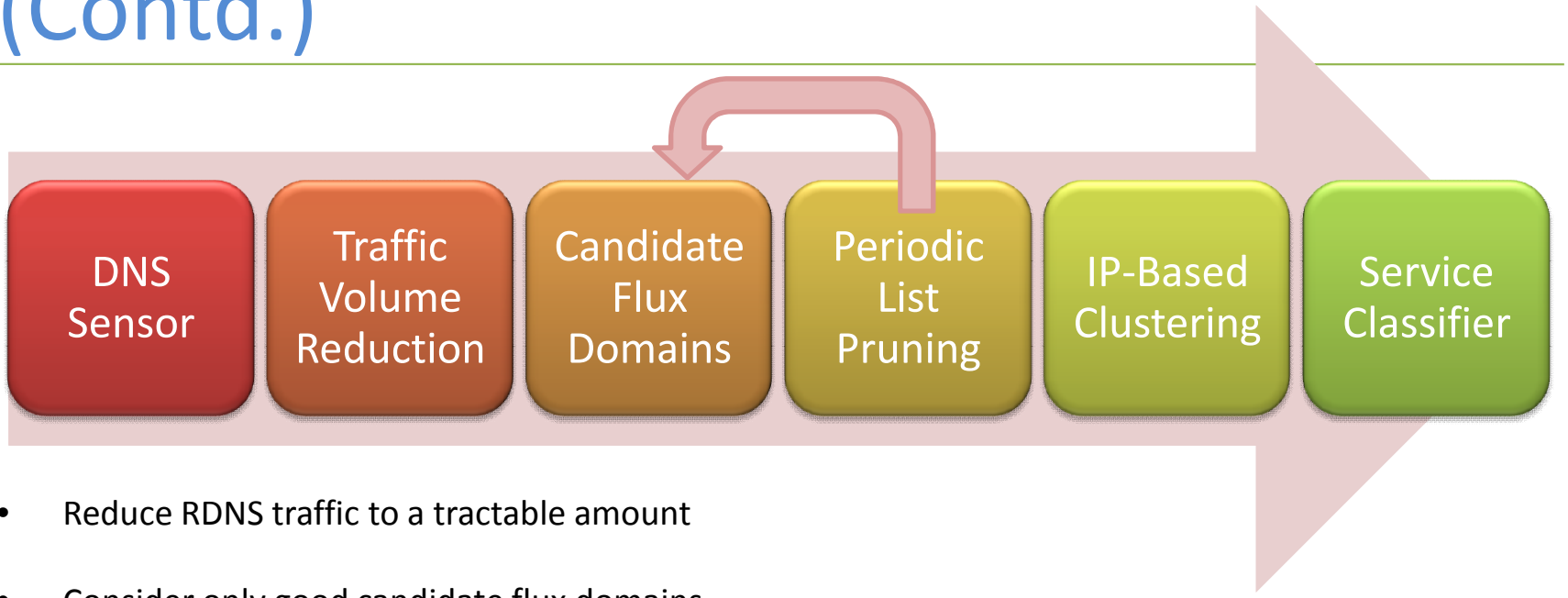
- Fluxiness
  - Total number of unique A records / Number of A records in a single lookup
  - Value of 1.0 implies subset = superset, common for benign domains
  - Value > 1.0 indicates CDNs and FFSNs
- Flux score
  - Vector x, (N_A, N_{ASN}, N_{NS})
  - $f(x) = W_1.N_A + w_2.N_{ASN} + w_3.N_{NS}$
  - $f(x) > b$ indicates a fast-flux service network
  - Turns out that $w_2 = 0$

[R. Perdisci. et. al., ACSAC 2009]

FFN Detection- Passive Approach

- Monitor R-DNS traffic generated by a large number of users
- Witness when a user clicks on malicious URLs
- Passively collect queried domains and resolved IPs

FFN Detection- Passive Approach (Contd.)



- Reduce RDNS traffic to a tractable amount
- Consider only good candidate flux domains
- May include legitimate/non-fux domains
- Group together domain names related to same network E.g., same flux network, same legitimate CDN, same NTP pool, etc.
- Classify each cluster of domains into either malicious flux or legitimate/non-flux

Classify Domain Names into FF/ Non FF

- A set of statistical features are used to distinguish flux domains and non-flux domains
  - FFN passive features
  - FFN Active features
- The C4.5 decision-tree classifier is applied to automatically classify a cluster as either malicious FF service or legitimate service

FFN Passive Features

- Number of resolved IPs
- Number of domains
- TTL per domain
- Network prefix diversity
- Number of domains per network
- IP Growth Ratio

FFN Active Features

- Organization diversity
- Country Code diversity
- Dynamic IP ratio
- Average Uptime Index

FFN Characterization

- The following results are based on studies conducted by:
  - [T. Holz. et. al., NDSS 2008]
  - [M. Konte. et. al., PAM 2009]

Similarity of scam pages

- The objective is to know how many scam pages are hosted by each IP address
- Problem: How to decide whether two pages are similar
- Solution: Use “string kernel”

String Kernel

- For pages p_1 and p_2,
  - Find all instances of a common string in p1 and p2
  - Multiply the occurrence in p1 by the occurrence in p2
  - Repeat with the next common string
  - Add all the multiplied occurrences

$$k(p_1, p_2) = \sum_{a \in A} \phi_a(p_1) \cdot \phi_a(p_2)$$

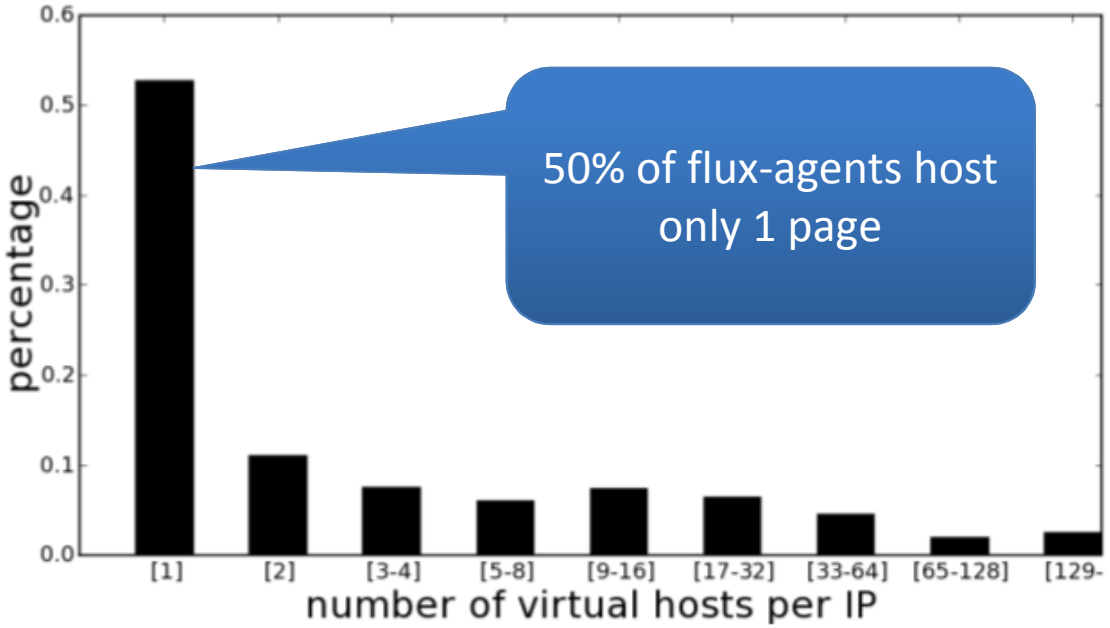
- Bound the result by normalizing it

$$\hat{k}(p_1, p_2) = \frac{k(p_1, p_2)}{\sqrt{k(p_1, p_1) \cdot k(p_2, p_2)}}$$

Grouping of web pages

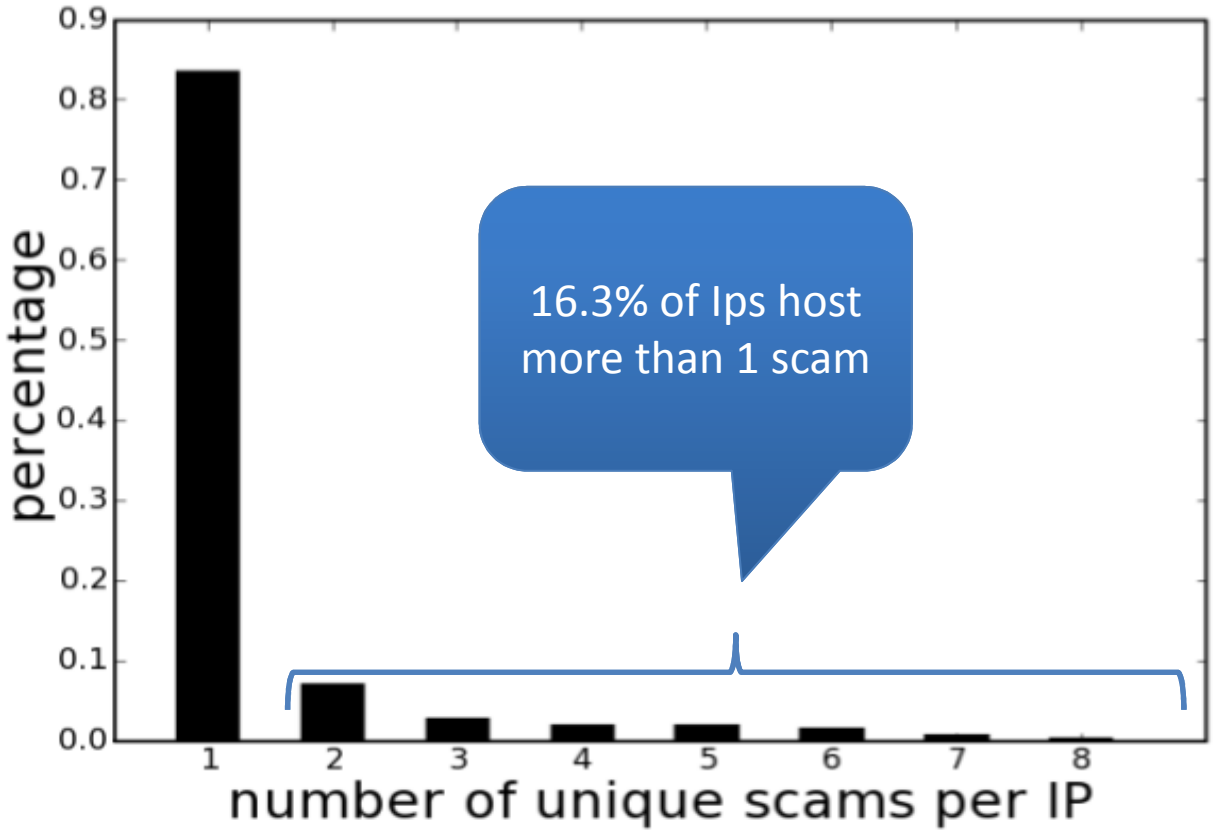
- Assign pages to the same group if k(pi, pj)>t, where the threshold t is 0<t<1
- Empirical study puts t=0.85

Distribution of virtual hosts per IP address per flux-agent



[source: T. Holz. et. al., NDSS 2008]

Distribution of unique scams per IP address per flux-agent



[source: T. Holz. et. al., NDSS 2008]

[M. Konte. et. al., PAM 2009]

Rate of Change of DNS Records

- Study:
  - Examine the rates at which fastflux networks redirect clients to different authoritative name servers (either by changing the authoritative nameserver's name or IP address), or to different Web sites entirely.
- Finding:
  - DNS TTL values do not differ fundamentally from other sites that do DNS-based load balancing
  - The rates of change differ fundamentally from legitimate load balancing activities
  - The rates of change differ across individual scam campaigns

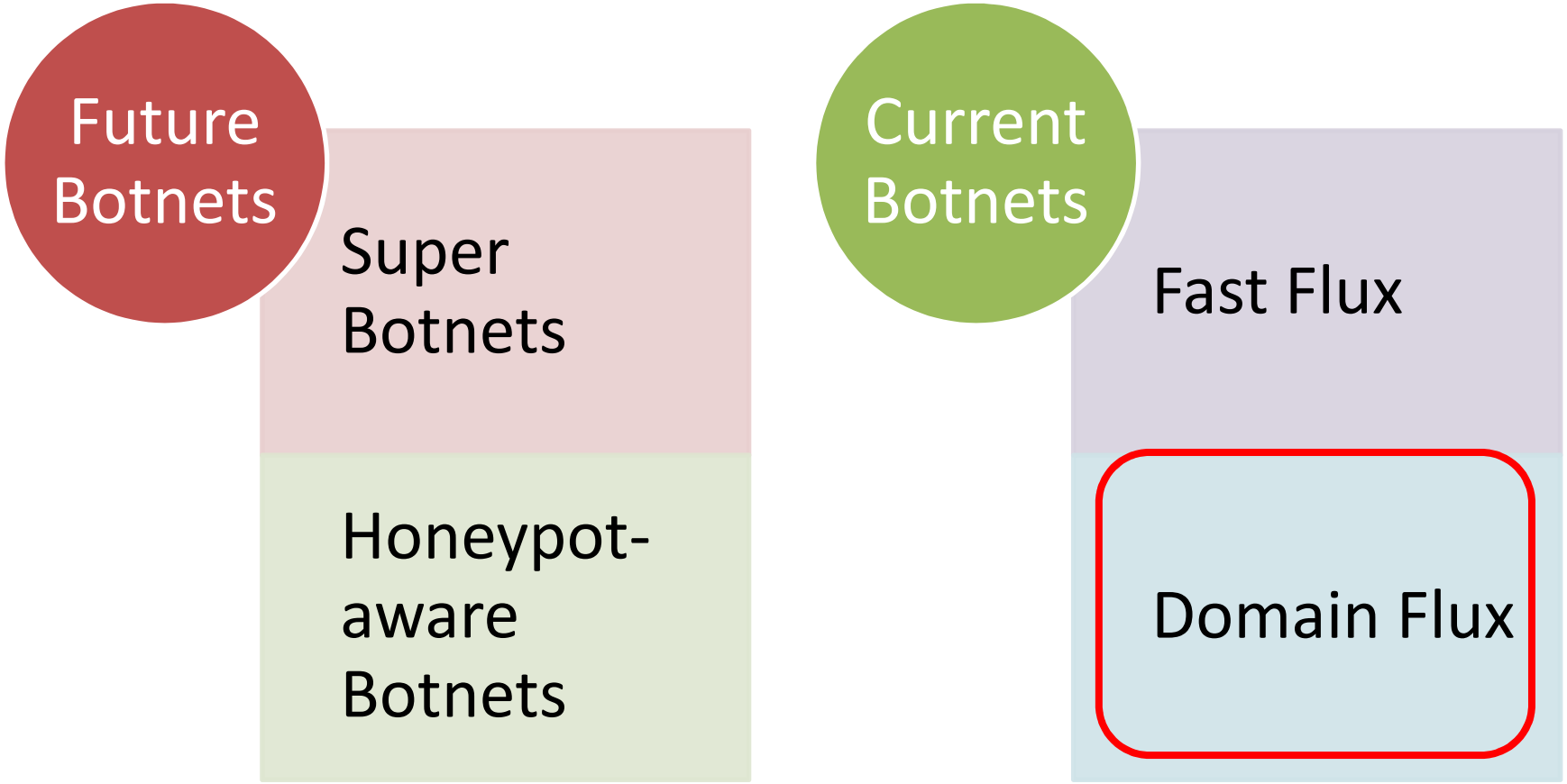
Rate of Accumulation

- Study:
  - The extent to which individual fast-flux networks "recruit" new IP addresses and how the rate of growth varies across different scam campaigns
- Finding:
  - There is a considerable amount of sharing of IP addresses across different scam campaigns
  - Different campaigns accumulate new IP addresses at different rates

Location of Change

- Study:
  - The extent to which fastflux networks change the Web servers to which clients are redirected.
- Finding:
  - Behavior differs by campaign
  - Many scam campaigns redirect clients by changing all three types of mappings, whereas most legitimate load-balancing activities only involve changes to A records.

Botnets- New Trends



[B. Stone-Gross, CCS 2009]

Domain Flux

- Fast-flux uses only a single domain name, which constitutes a single point of failure
- Torpig solves this issue by using a different technique for locating its C&C servers → domain flux
- If a domain is blocked, the bot simply rolls over to the following domain in the list
- Using the generated domain name dw, a bot appends a number of TLDs: in order, dw.com, dw.net, and dw.biz
- If all three connections fail, Torpig computes a “daily” domain, say dd, which in addition depends on the current day

Domain flux (Contd.)

```
suffix =  ["anj", "ebf", "arm", "pra", "aym", "unj",
    "ulj", "uag", "esp", "kot", "onv", "edc"]

def generate_daily_domain():
    t = GetLocalTime()
    p = 8
    return generate_domain(t, p)

def scramble_date(t, p):
    return (((t.month ^ t.day) + t.day) * p) +
        t.day + t.year

def generate_domain(t, p):
    if t.year < 2007:
        t.year = 2007
    s = scramble_date(t, p)
    c1 = (((t.year >> 2) & 0x3fc0) + s) % 25 + 'a'
    c2 = (t.month + s) % 10 + 'a'
    c3 = ((t.year & 0xff) + s) % 25 + 'a'
    if t.day * 2 < '0' ||  t.day * 2 > '9':
        c4 = (t.day * 2) % 25 + 'a'
    else:
        c4 = t.day % 10 + '1'
    return c1 + 'h' + c2 + c3 + 'x' + c4 +
        suffix[t.month - 1]
```

Listing 1: Torpig daily domain generation algorithm.

[Source: B. Stone-Gross, CCS 2009]

References

- T. Holz, C. Gorecki, K. Rieck, and F. Freiling. Measuring and detecting fast-flux service networks. In Network & Distributed System Security Symposium, 2008
- X. Hu, M. Knysz, and K. G. Shin. Rb-seeker: Auto-detection of redirection botnets. In Network & Distributed System Security Symposium, 2009
- M. Konte, N. Feamster, and J. Jung. Dynamics of online scam hosting infrastructure. In Passive and Active Measurement Conference, 2009
- J. Nazario and T. Holz. As the net churns: Fast-flux botnet observations. In International Conference on Malicious and Unwanted Software, 2008
- The Honeynet Project. Know your enemy: Fast-flux service networks, 2007
- E. Passerini, R. Paleari, L. Martignoni, and D. Bruschi. Fluxor: Detecting and monitoring fast-flux service networks. In Detection of Intrusions and Malware, and Vulnerability Assessment, 2008

References (Contd.)

- R. Perdisci, I. Corona, D. Dagon, W. Lee. Detecting Malicious Flux Service Networks through Passive Analysis of Recursive DNS Traces". Annual Computer Security Applications Conference, ACSAC 2009
- Brett Stone-Gross, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydlowski, Richard Kemmerer, Chris Kruegel, and Giovanni Vigna, "Your Botnet is My Botnet: Analysis of a Botnet Takeover," in Proceedings of the ACM CCS, Chicago, IL, November 2009
- P. Wang et.al., A systematic Study on Peer to Peer Botnets, IEEE ICCCN 2009
- VOGT, R., AYCOCK, J., and JACOBSON, M., “Army of botnets,” in Proceedings of the 14th Network and Distributed System Security Symposium (NDSS’07), 2007.
- ZOU, C. C. and CUNNINGHAM, R., “Honeypot-aware advanced botnet construction and maintenance,” in International Conference on Dependable Systems and Networks (DSN’06), 2006
- RAJAB, M., ZARFOSS, J., MONROSE, F., and TERZIS, A., “A multi-faceted approach to understanding the botnet phenomenon,” in Proceedings of ACM SIGCOMM/ USENIX Internet Measurement Conference (IMC’06), (Brazil), October 2006
- Thorsten Holz, et.al., Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on StormWorm, LEET 2008

Module III:
Botnet-Based Attacks

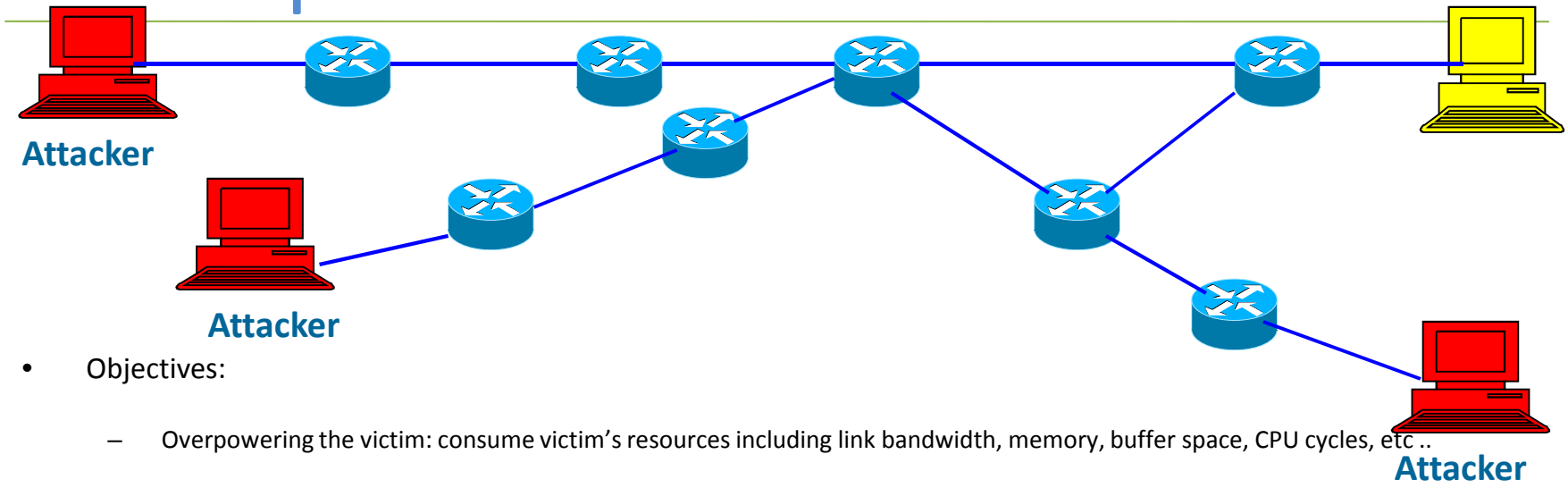
Talk Outline – Module III

- **DDoS Attacks**
- Spam
- Identity Theft
- Phishing
- Click Fraud

Denial of Service Attacks

- DoS attacks are malicious means of denying Internet services to legitimate users or processes
- In general, DoS attacks are easy to conduct, yet difficult to defeat
- The spread of attack tools and the easy access to them through search engines
- DoS attacks are developing more quickly than the defenses used to fight them
- Theoretically, any system connected to the Internet is considered to be a potential target

DoS Attacks- Objectives and Consequences



- Objectives:
  - Overpowering the victim: consume victim's resources including link bandwidth, memory, buffer space, CPU cycles, etc ..
  - Concealing attacker's identity
- Consequences:
  - Service not available
  - Network congestion and service degradation
  - Leads to enormous economical losses

DoS Attacks- Basis

- Attackers usually abuse the following characteristics of Internet protocols to perform DoS attacks

Facilitated DoS attacks that employ source IP Spoofing

  - Destination oriented routing: The routing protocols were designed to be destination oriented
  - Stateless nature of the Internet: Routers do not maintain any state information about forwarded packets
  - Lack of authenticity over the Internet: Without authentication, malicious Internet user[illegible] being easily detected o[illegible]

Facilitated DoS attacks that exploit the predictable operation of Internet protocols

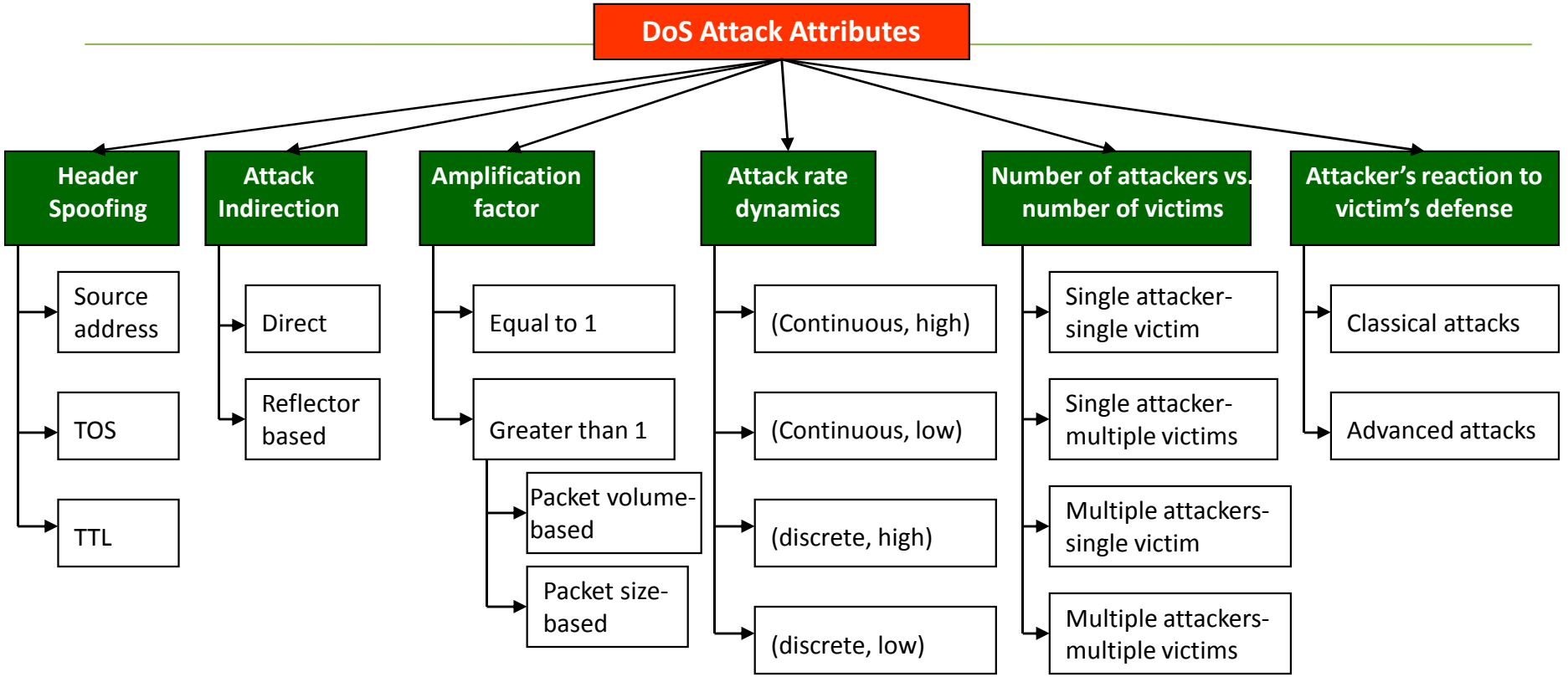
  - Deterministic nature of Internet protocols: This is not a design flaw, but is often necessary to the proper operation of Internet protocols

DoS Attacks

- To launch a powerful DoS attack, an attacker has to secure enough resources to achieve the desired damage to the victim
- Compromising thousands of computers is done in a phase, known as the recruitment phase, that precedes the actual DoS attack
  - The attacker performs extensive scanning of remote machines searching for vulnerabilities and security holes
  - The discovered vulnerabilities are exploited to break into the scanned systems. At this point, the attacker gets access to these systems, which are then called zombies or slaves
  - The attacker installs the attack tool on the compromised computers. At this point, the compromised computers become ready to participate in the attack, or even to be used in the recruitment of other computers.

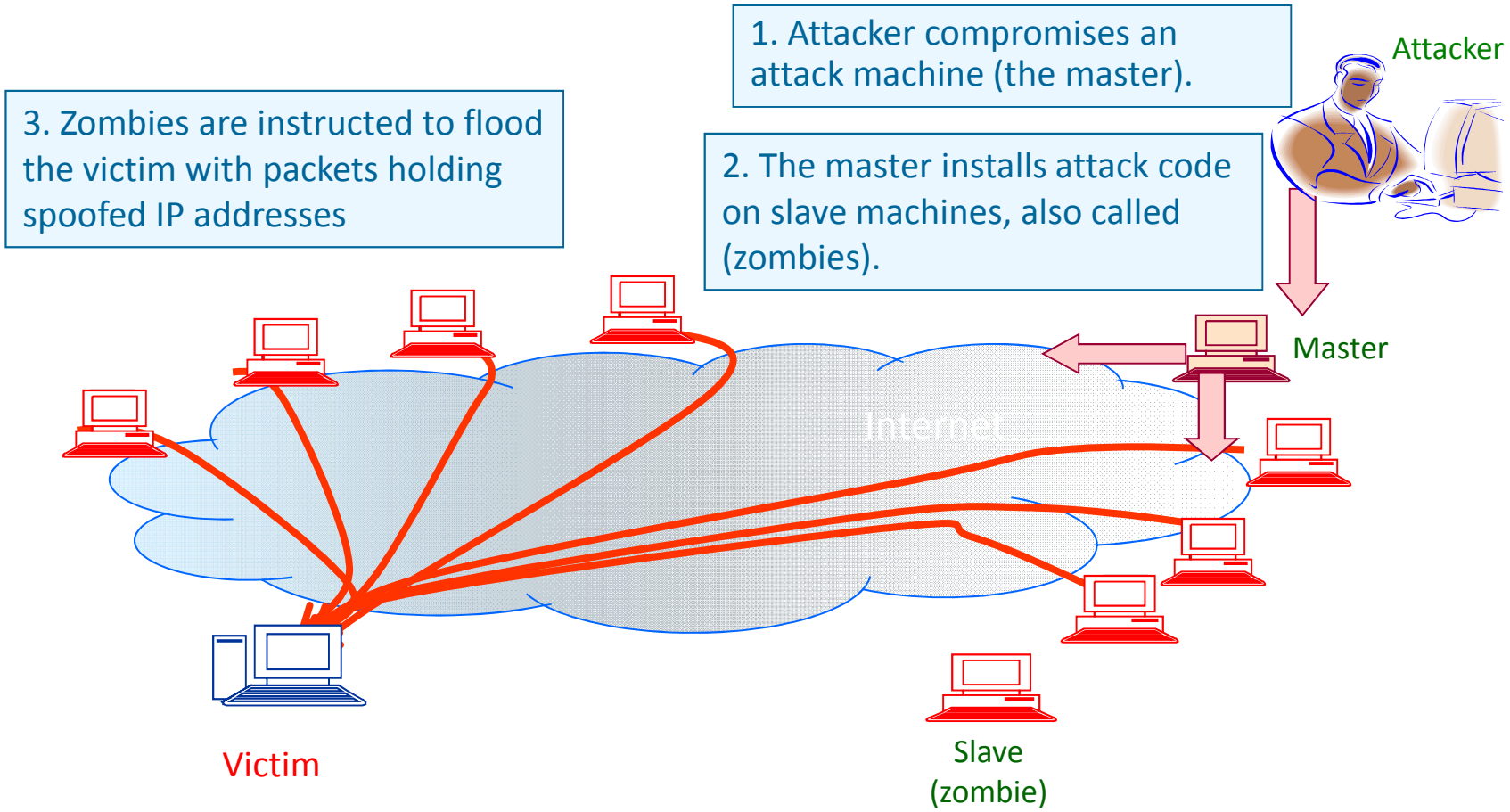
DoS Attack Attributes

- Before launching a DoS attack, an attacker should configure the attack tool in such a way as to achieve the desired damage to the victim
- This involves the specification of several attack attributes that shape the overall nature of the attack
- “Attribute” refers to certain aspect of an attack
  - Header spoofing
  - Attack indirection
  - Attack amplification factor
  - Attack rate dynamics
  - Number of attackers vs. number of victims
  - Attacker's reaction to the victim's defense



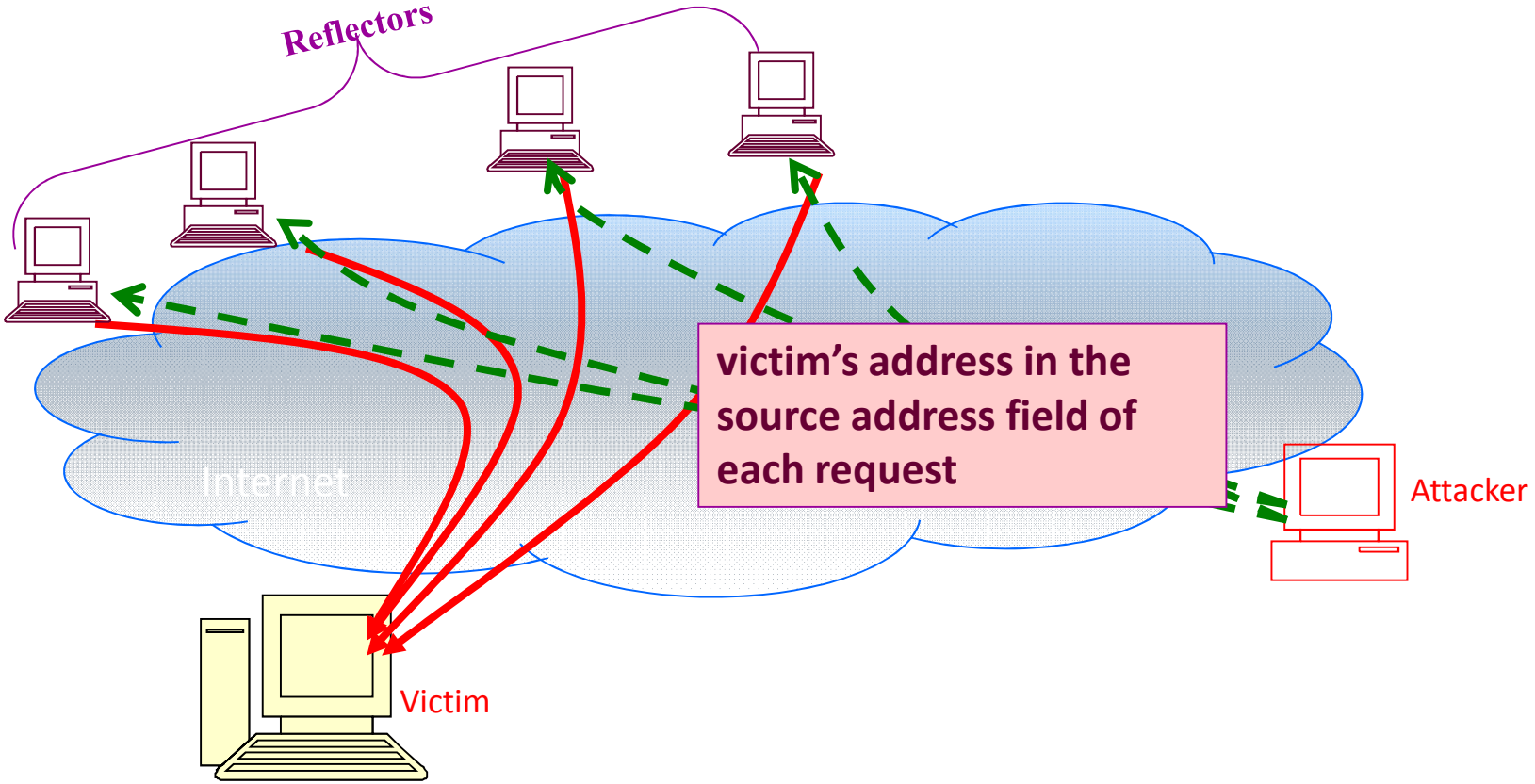
Attribute-Based Classification

Direct DoS Attacks



Indirect DoS Attacks

- Feasible in verity of request/reply based protocols (e.g., TCP, DNS, ICMP, and UDP)



Attack Amplification Factor

- Attack amplification refers to the amount of gain in resource (e.g., bandwidth) an attacker achieves for each emitted attack packet
- If the attacker emits an attack packet of size *x*, for which the victim receives an amount of traffic of size *y*, then we say that the amplification factor for this attack is $f = y/x$
- Most of direct DoS attacks have an amplification factor of 1
- In reflector-based DoS attacks, an amplification factor of more than one is usually noticeable
  - Number-based amplification (Example: smurf attack)
  - The second is packet size-based amplification (Example: DNS amplification attack)

Means-Based Classification

- This classification takes into consideration the means of performing a DoS attack

- Two categories:

  – Brute force-based attacks: adopt the idea of brute force resource exhaustion

  – Protocol exploitation-based attacks: adopt the idea of exploiting the deterministic nature of certain Internet protocols to significantly degrade their throughput without injecting a lot of traffic in the Internet

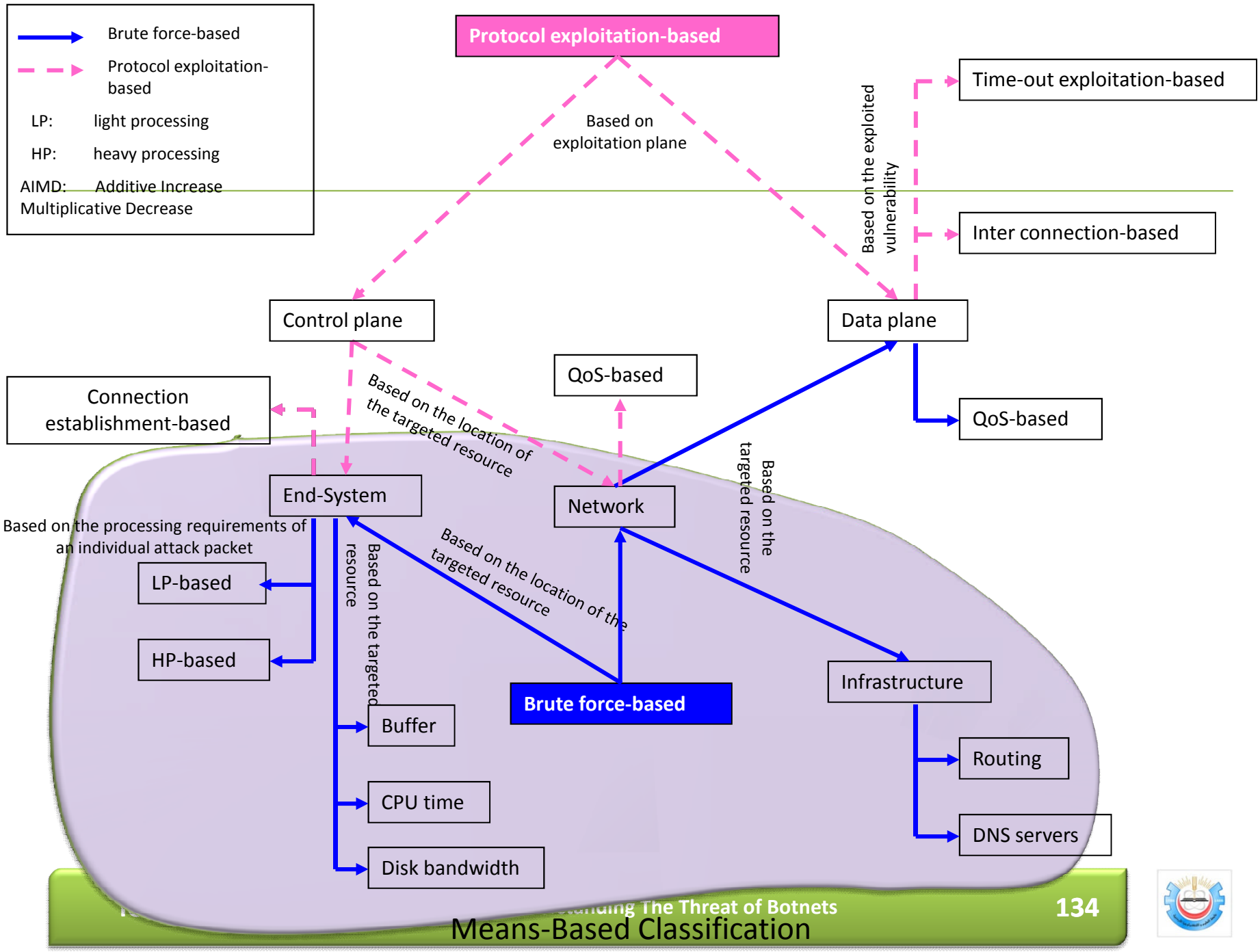
Brute force-Based DoS Attacks

- The target is located at an end system: The aim of these attacks is to occupy a disproportional amount of victim's resources for maximum amount of time

- The targeted resource could be victim's buffer space, bandwidth, CPU cycles, or a combination of them

  - Light processing-based: usually characterized by a very intensive attack rate that brings the total load beyond the victim's capacity
  - Heavy processing-based: usually characterized by submitting a large number of computationally intensive tasks to the victim
    - An authentication process
    - Downloading huge files from a Web or FTP server in overwhelming numbers

- The target is located inside the network
  - DNS Servers
  - BGP Routers
  - DiffServ domain (QoS-Based Attack)



Means-Based Classification

Botnet-Based DDoS Attacks

- Attacker form/Rent a BIG Botnet
  - Single botnets have numbered 1.5 million
  - Huge Aggregate Bandwidth → Flood many core links, small-medium ISPs
- Bots are instructed to launch DDoS Attacks against a given target
  - Send high volume of SYN packets (SYN flooding)
  - Issue thousands of requests to download a large file from the victim → mimic flash crowd

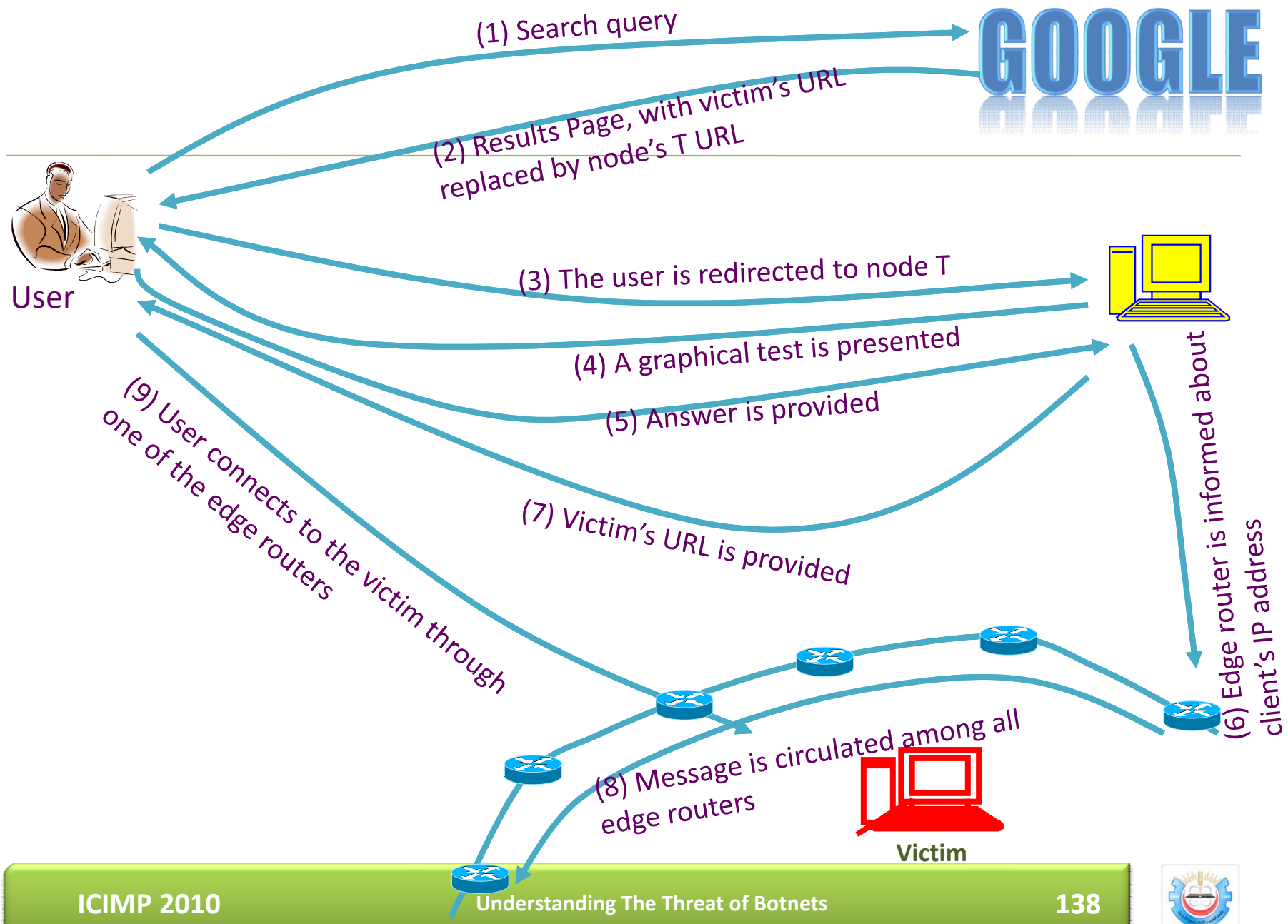
Countering Botnet-Based DDoS Attacks

- Kill-bots [S. Kandula et.al., USENIX NDSI 2005]: a kernel extension to protect Web servers against DDoS attacks that masquerade as flash crowds
  - Distinguishes human users from zombie machines by presenting a puzzle to the client. It provides authentication using graphical tests.
- Phalanx [C. Dixon et.al., USENIX NDSI 2008]: In Phalanx, a client communicating with a destination bounces its packets through a random sequence of end-host mailboxes
  - because an attacker does not know the sequence, they can disrupt at most only a fraction of the traffic, even for end-hosts with low bandwidth access links.

Countering Botnet-Based DDoS Attacks

- JUST-Google [B. Al-Duwairi. et. al., ICC 2009]:
- Website Traffic can be classified into
  - Category 1: Search engine referred traffic
  - Category 2: Direct access.
  - Category 3: Referral from other web pages.
  - Category 4: Attack traffic (usually originating from Botnets).

- Fact: Category 1 forms a great percentage of a Website traffic
  - Visiting a Web site is usually preceded by queering Google searching for a specific piece of information
  - In most cases, when a user fails to access a certain Web site, directly by typing its URL (Category 2), or through referrals from other web pages (Category 3) he/she would use a search engine to reach the Web site


(1) Search query
GOOGLE
(2) Results Page, with victim's URL replaced by node's T URL
User
(3) The user is redirected to node T
(4) A graphical test is presented
(5) Answer is provided
(6) Edge router is informed about client's IP address
(7) Victim's URL is provided
(9) User connects to the victim through one of the edge routers
(8) Message is circulated among all edge routers
Victim

Talk Outline – Module III

- DDoS Attacks
- **Spam**
- Identity Theft
- Phishing
- Click Fraud

[C. Kreibich. et. al., LEET 2008]

Spam

- Unsolicited commercial message
- Spam Problem dates back to the early-1990s
- Solving the Spam problem:
  - By maintaining “blacklists” of IP addresses
  - Filtering on spam content itself

- IP blacklists have forced the development of bot-based distribution networks that use compromised PC’s to relay messages and launder their true origin

- The use of filters based on statistical learning have in turn caused spammers to dynamically add textual polymorphism to their spam, thus evading the filters

Spam Campaigns

- Spammers divide their efforts into individual campaigns that are focused on a particular goal, whether it is selling a product, committing financial fraud, or distributing malware
- A spam campaigns typically consist of:
  - A target list of email addresses—either harvested via crawling or malware or purchased outright via underground markets
  - A set of subject and body text templates that are combined mechanically to create an individual message for each targeted address
- A spam campaign is executed by some distribution platform—typically a botnet—and this infrastructure can be reused by multiple campaigns

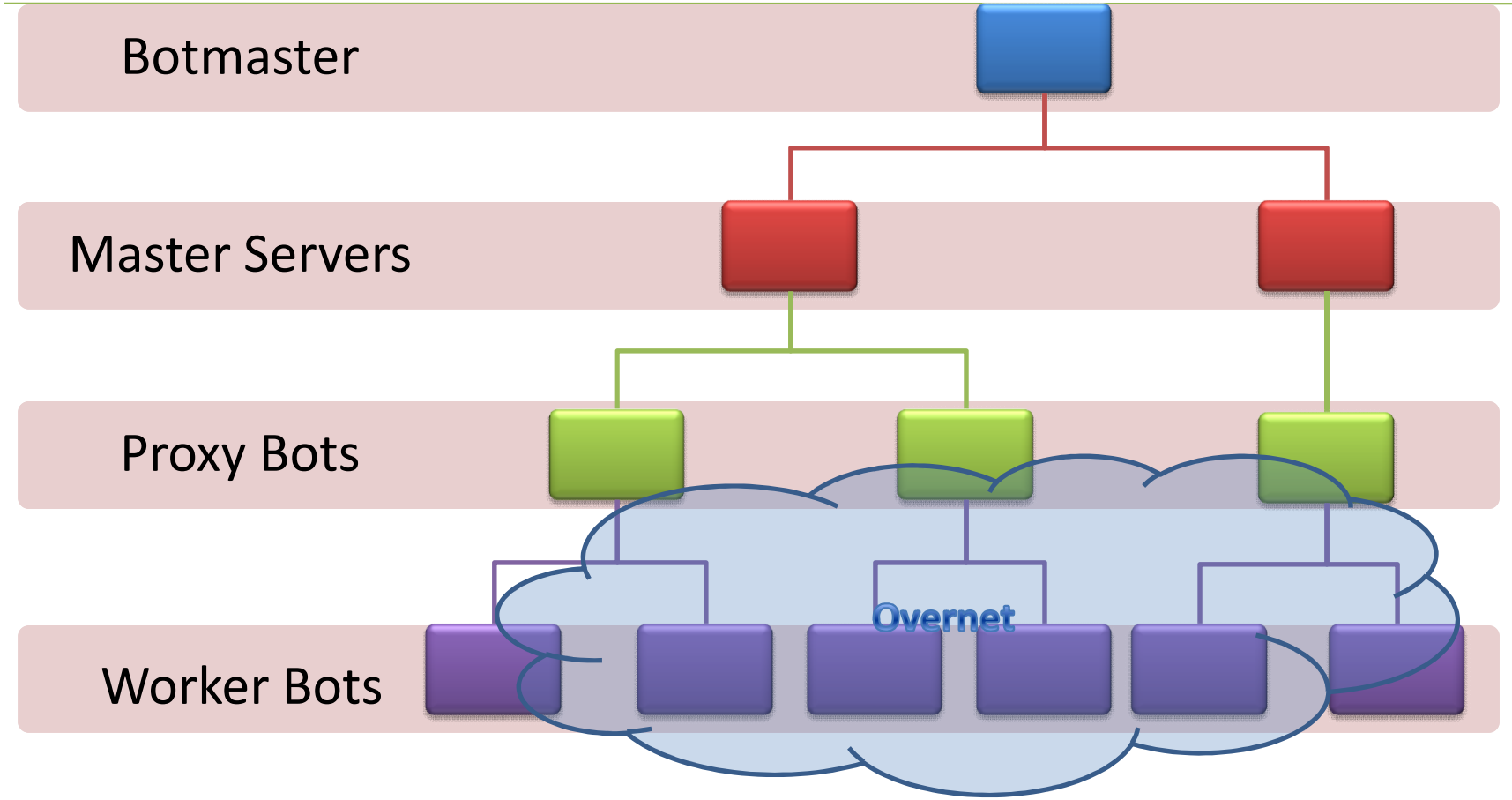
Spam Campaigns (Contd.)

- To achieve scalability
  - load of delivering a spam campaign must be balanced across the infrastructure
  - The infrastructure is typically responsible for the task of evading textual spam filters
  - generate each message algorithmically based on the campaign's text templates and a set of evasion rules, or macros

Spamming via Storm

- Storm employs a tiered coordination mechanism
  - Worker bots (at the lowest level): access a form of the Overnet peer-to-peer network to locate C&C proxy bots
    - Perform Spam
  - Proxies: Organize workers
    - Workers relay through the proxies requests for instructions and the results of executed commands, receiving from them their subsequent C&C
  - Master servers: Controlled directly by the botmaster Bullet-proof hosting sites: The proxies in turn interact with “bullet-proof hosting” sites under control of the botmaster

Storm Architecture



Storm- Message structure and propagation

- Update messages consist of three sections:
  - Template material
  - Sets of dictionaries containing raw text material to substitute into templates
  - Lists of target email addresses. These lists typically provide roughly 1,000 addresses per update message
- The infrastructure can report back failures, allowing the spammer to weed out addresses from their target list that are not viable

Spam Template

```
Received: from %^C0%^P%^R2-6^%:qwertyuiopasdfghjklzxcvbnm^%.%^P%^R2-6^%:qwertyuiopasdfghjkl ▷
                                        zxcvbnm^%^% ([%^C6%^I^%.%^I^%.%^I^%.%^I^%^%]) by ▷
                                        %^A^% with Microsoft SMTPSVC(%^Fsvcver^%); %^D^%
Message-ID: <%^O%^V6^%:%^R3-50^%^%%^V0^%>
From: <%^Fnames^%@%^Fdomains^%>
To: <%^0^%>
Subject: JOB $1800/WEEK - CANADIANS WANTED!
Date: %^D-%^R30-600^%^%

---------------------------------------------------------------

Received: from auz.xwzww ([132.233.197.74]) by dsl-189-188-79-63.prod-infinitum.com.mx with ▷
                         Microsoft SMTPSVC(5.0.2195.6713); Wed, 6 Feb 2008 16:33:44 -0800
Message-ID: <002e01c86921$18919350$4ac5e984@auz.xwzww>
From: <katiera@experimentalist.org>
To: <voelker@cs.ucsd.edu>
Subject: JOB $1800/WEEK - CANADIANS WANTED!
Date: Wed, 6 Feb 2008 16:33:44 -0800
```

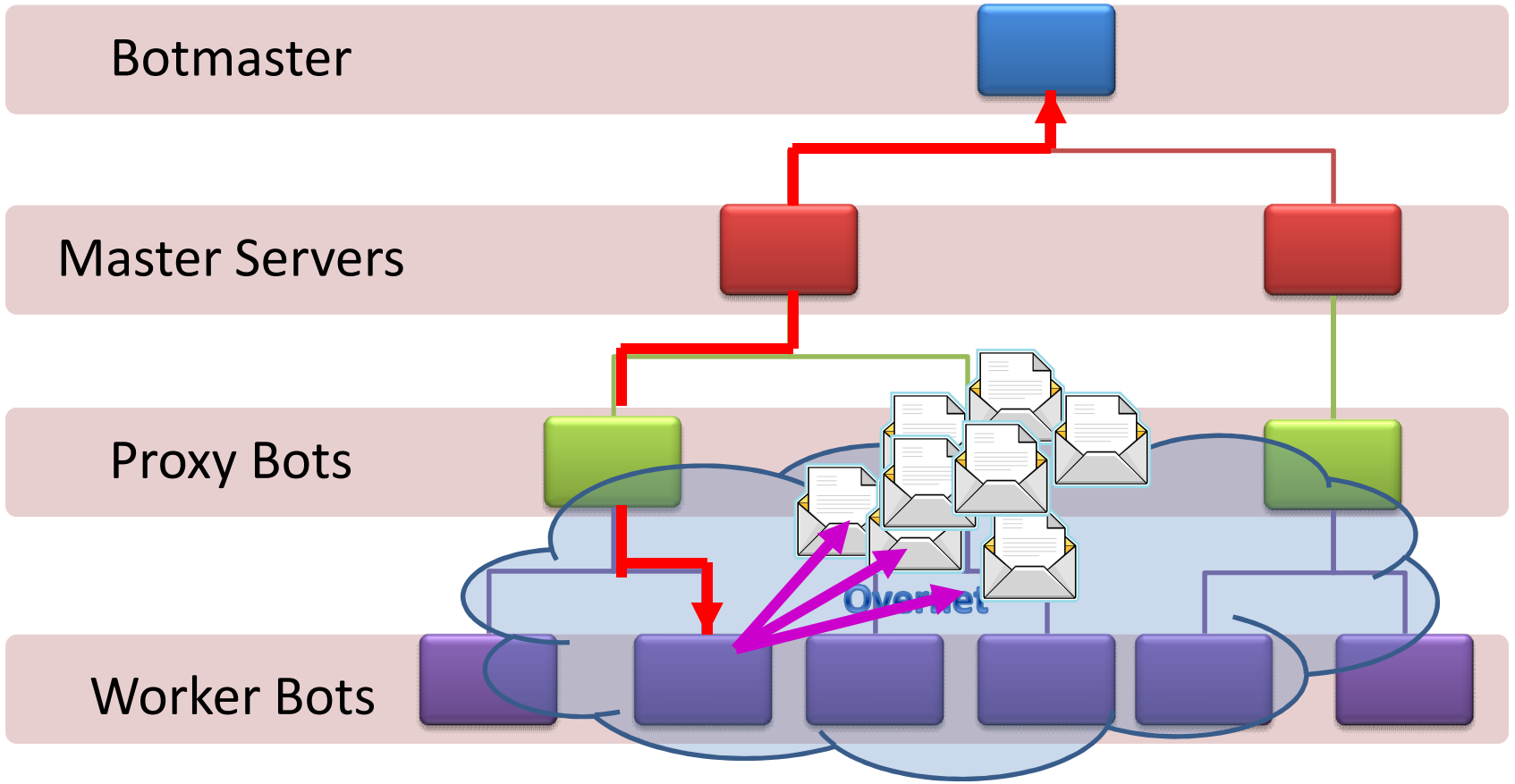
[Source: C. Kreibich., et. al., USENIX, LEET 2008]

Storm Setup

- New bots decide if they are proxies or workers
  - Inbound connectivity? Yes, proxy. No, worker
- Proxies advertise their status via encrypted variant of Overnet DHT P2P protocol
  - Master sends "Breath of Life" packet to new proxies to tell them IP address of master servers (RSA signature)
  - Allows master servers to be mobile if necessary
- Workers use Overnet to find proxies
- Workers send to proxy, proxy forwards to one of master servers in "safe" data center

Storm Architecture

Template, target addresses, Dictionary



Talk Outline – Module III

- DDoS Attacks
- Spam
- **Identity Theft**
- **Phishing**
- Click Fraud

[B. Stone-Gross, et.al, ACM CCS 2009]

Identity Theft through the Torpig Botnet

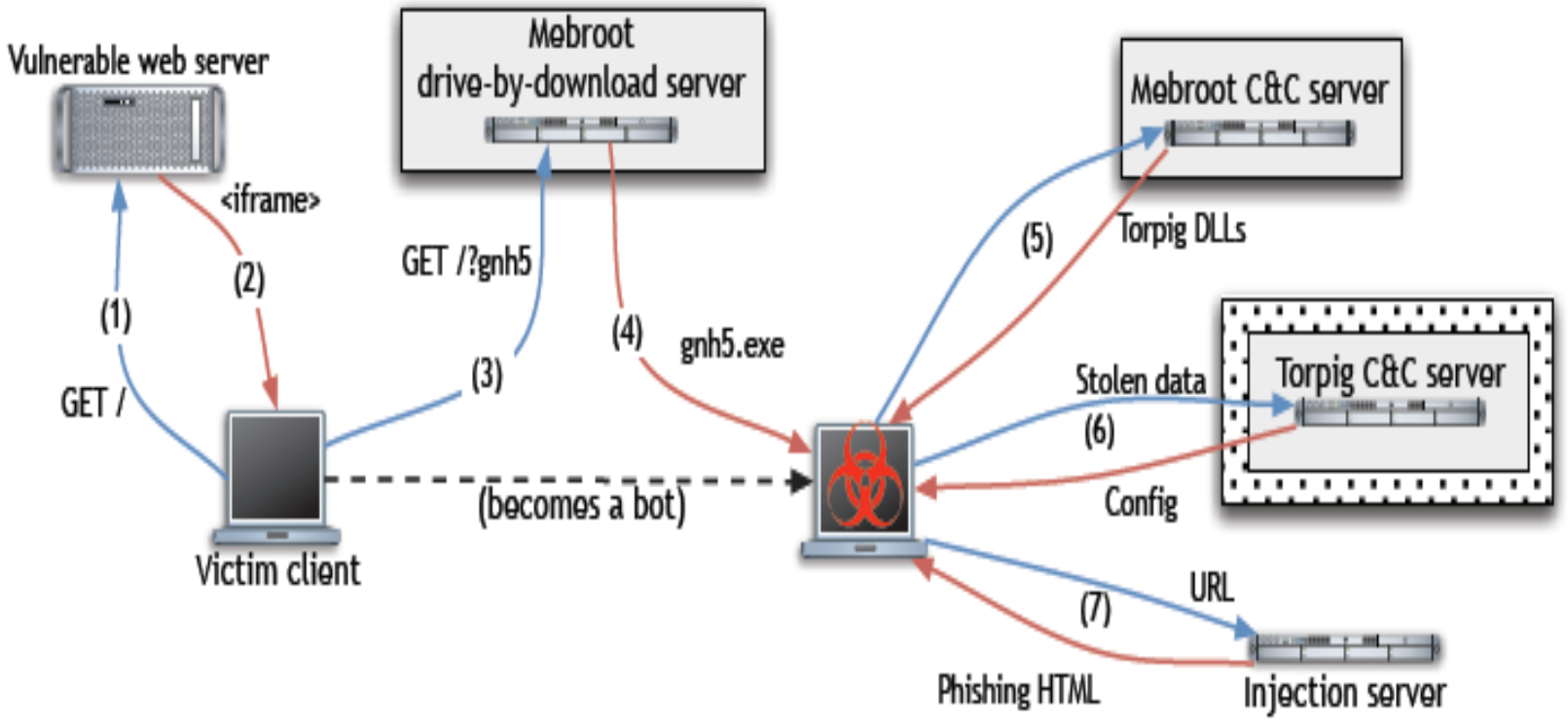
- Torpig botnet is a type of malware that is typically associated with bank account and credit card theaft
- " It is one of the most advanced pieces of crimeware ever created" [M. Shields, BBC news, 2008]
- Features:
  - Sophisticated techniques to steal data
  - Complex network infrastructure
  - Vast financial damage

Torpig Botnet- Basic Operation

- Torpig has been distributed to its victims as part of Mebroot
  - Mebroot* is a rootkit that takes control of a machine by replacing the system's Master Boot Records (MBR)
  - This allows Mebroot to be executed at boot time, before the operating system is loaded, and to remain undetected by most anti-virus tools
- Victims are infected through drive-by-download attacks

* Rootkits is a type of malware that attempt to hide their presence on a system, typically by compromising the communication conduit between an Operating System and its users.

[Source: B. Stone-Gross, et.al, ACM CCS 2009]



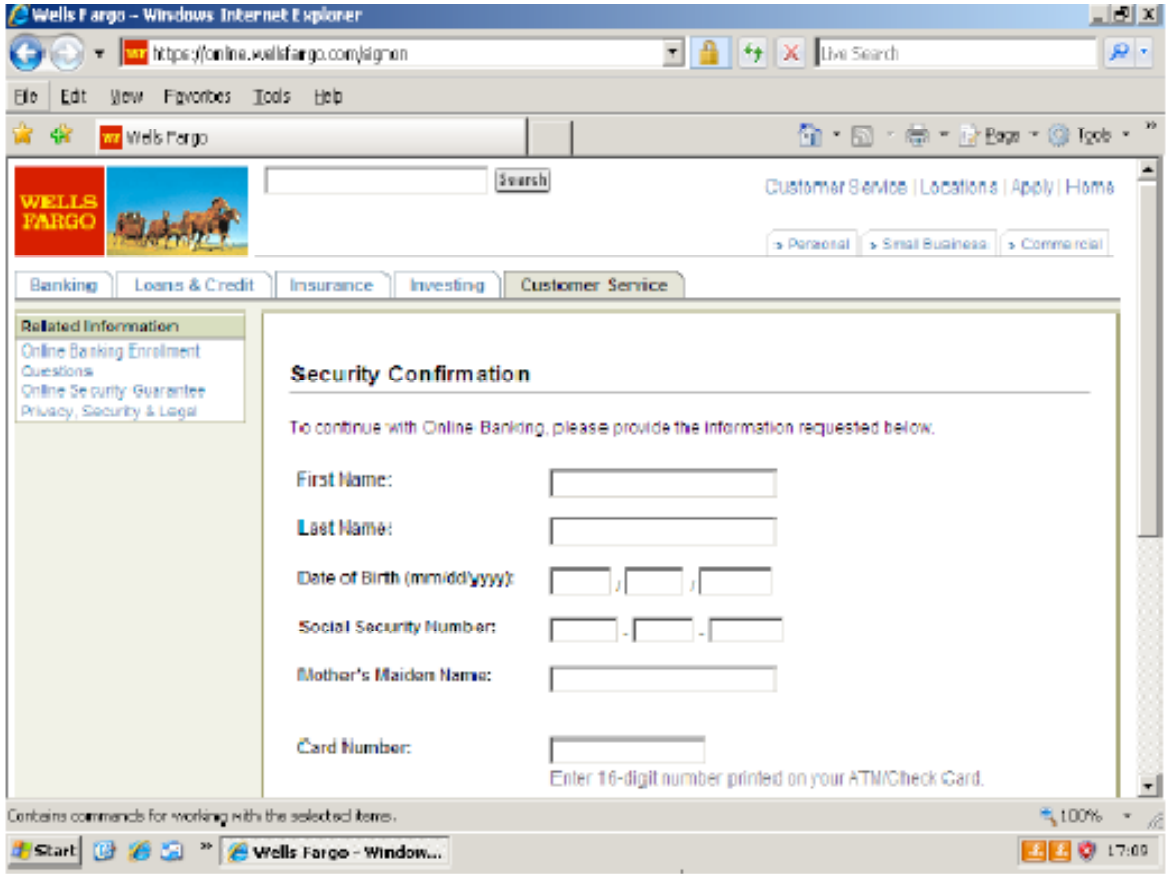
(6) Torpig contacts the Torpig C&C server to upload the data stolen since the previous reporting time

Phishing through The Torpig Botnet

- Torpig uses phishing attacks to actively elicit additional, sensitive information from its victims, which, otherwise, may not be obtained

- First, whenever the infected machine visits one of the domains specified in the configuration file (typically, a banking web site), Torpig issues a request to an injection server
  - The server's response specifies a page on the target domain where the attack should be triggered (we call this page the trigger page and it is typically set to the login page of a site), a URL on the injection server that contains the phishing content (the injection URL), and a number of parameters that are used to fine tune the attack (e.g., whether the attack is active and the maximum numberof times it can be launched)
- The second step occurs when the user visits the trigger page. At that time, Torpig requests the injection URL from the injection server and injects the returned content into the user's browser (7).

- This content typically consists of an HTML form that asks the user sensitive information such as credit card numbers and social security numbers.


Wells Fargo - Windows Internet Explorer
https://online.wellsfargo.com/signon
File Edit View Favorites Tools Help
Wells Fargo
Page Tools
WELLS FARGO
Search
Customer Service | Locations | Apply | Home
› Personal › Small Business › Commercial
Banking Loans & Credit Insurance Investing Customer Service
Related Information
Online Banking Enrollment Questions
Online Security Guarantee
Privacy, Security & Legal
Security Confirmation
To continue with Online Banking, please provide the information requested below.
First Name:
Last Name:
Date of Birth (mm/dd/yyyy):
Social Security Number:
Mother's Maiden Name:
Card Number:
Enter 16-digit number printed on your ATM/Check Card.
Contains commands for working with the selected items.
100%
Start
Wells Fargo - Window...
17:09

Data Collected by Torpig

Country	Institutions (#)	Accounts (#)
US	60	4,287
IT	34	1,459
DE	122	641
ES	18	228
PL	14	102
Other	162	1,593
Total	410	8,310

Table 3: Accounts at financial institutions stolen by Torpig.

Domain	Account Credentials	Type
google.com	8,291	Search/Email
facebook.com	7,812	Social Networking
myspace.com	7,214	Social Networking
netlog.com	4,528	Social Networking
libero.it	4,374	Search/Email/ISP
yahoo.com	4,029	Search/Email
nasza-klasa.pl	3,628	Social Networking
alice.it	3,348	Search/Email/ISP
live.com	3,133	Search/Email
hi5.com	3,090	Social Networking

Table 5: Top web account credentials sent by Torpig victims.

Data Type	Data Items (#)
Mailbox account	54,090
Email	1,258,862
Form data	11,966,532
HTTP account	411,039
FTP account	12,307
POP account	415,206
SMTP account	100,472
Windows password	1,235,122

Table 1: Data items sent to our C&C server by Torpig bots.

[Source: B. Stone-Gross, et.al, ACM CCS 2009]

Talk Outline – Module II

- DDoS Attacks
- Spam
- Identity Theft
- Phishing
- **Click Fraud**

Click fruad

- Pay-per-click advertising
  - Publishers display links from advertisers
  - Advertising networks act as middlemen
    - Sometimes the same as publishers (*e.g.,* Google)
- Click fraud: botnets used to click on pay-per-click ads
- Motivation
  - Competition between advertisers
  - Revenue generation by bogus content provider

Click Fraud Botnets

References

- J. Mirkovic and P. Reiher, A Taxonomy of DDoS Attacks and Defense Mechanisms," in Proc. ACM SIGCOMM Computer Communications Review, vol. 34, no. 2, April 2004. pp. 39-54
- V. Paxson, An analysis of using re°ectors for distributed denial-of-service attacks," in Proc. ACM SIGCOMM Computer Communication Review, vol. 31, no. 3, July 2001
- H. Wang, A. Bose, M. El-Gendy, and K. G. Shin, IP Easy-pass: Edge Resource Access Control," in Proc. of IEEE INFOCOM 2004, Hong Kong, China, March 2004
- S. Kandula, D. Katabi, M. Jacob, and A. Burger, Botz-4-Sale: Surviving DDos Attacks that Mimic Flash Crowds," in Proc. USENIX NSDI 2005. To appear, Boston, MA, May 2005
- A. Kuzmanovic and E. Knightly, \Low-rate TCP-targeted denial of service attacks," in Proc. ACM SIGCOMM 2003, Karlsruhe, Germany, August 2003

References (Contd.)

- S. Ebrahimi, A. Helmy, S. Gupta, A Systematic Simulation-based Study of Adverse Impact of Short-lived TCP Flows on Long-lived TCP Flows", in IEEE INFOCOM 2005, Miami, FL, March 2005

- J. Bellardo and S. Savage, 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions", in Proc. USENIX Security Symposium, Washington D.C., August 2003

- C. Kreibich, C. Kanich, K. Levchenko, B. Enright, V. Paxson, and S. Savage. On the Spam Campaign Trail. In Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats, 2008A

- Ramachandran and N. Feamster. Understanding the Network-level Behavior of Spammers. In ACM SIGCOMM, 2006.

Module IV:
Botnet Detection

[Goufie Gu, PhD thesis, Georgia Tech 2008]

Botnet Detection Challanges

- Bots are stealthy on infected machines
- Bot infection is usually a multi-faceted and multi-phased process, incorporating several computing assets, multiple bidirectional network flows, and different infection stages
- Bots are dynamically evolving
- Botnets can have a very flexible design of C&C channels

Botnet Detection Approaches

Honeypot-based Tracking

- [E. Cooke. et. al., USENIX SRUTI 2005]
- [D. Dagon. et. al., NDSS 2006]
- [M. Collins. et. al., IMC 2007]
- [P. Barford. et. al., Special workshop on Malware Detection]
- [F. Freiling. et. al., ESORICS 2005]

Heuristic-based

- [J. R. Binkley. et. al., USENIX SRUTI 2006]
- [A. Ramachandran. et. al., USENIX SRUTI 2006]
- [J. Goebel. et. al., USNIX HotBots 2007]

Traffic Analysis based

- [T. F. Yen. et. al., DIMVA 2008]
- [W. T. Strayer. et. al., LCN 2006]
- [A. Karasardis. et. al., USENIX HotBots 2007]
- [G. Gu. et. al., USENIX Security 2008]
- [G. Gu. et. al., USENIX Security 2007]
- [G. Gu. et. al., NDSS 2008]

Talk Outline – Module IV

- **Honeypot-based Detection**
- Hueristic-Based Detection
- Traffic Analysis-based Detection

Botnet Measurment

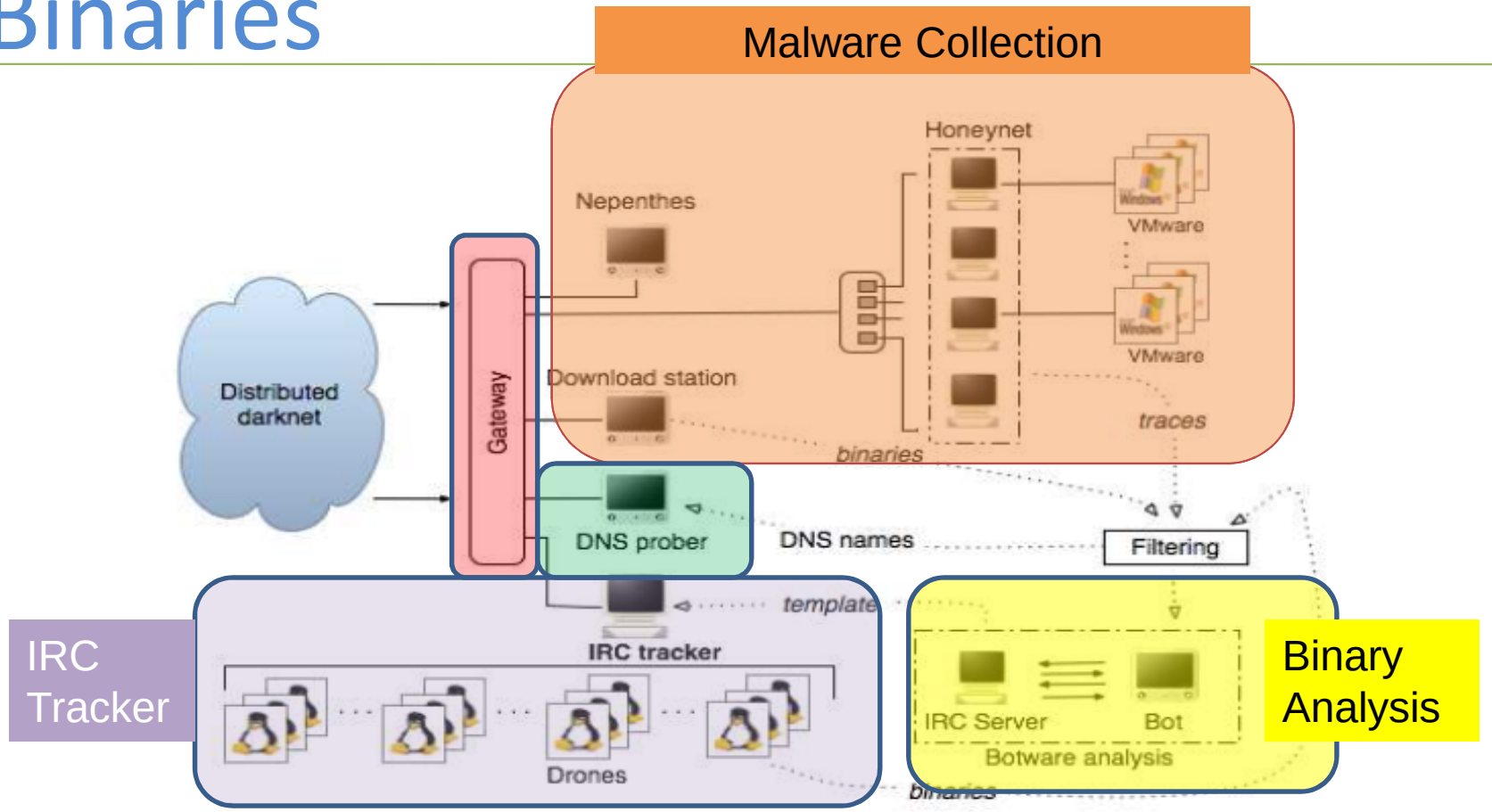
- Measurement studies can help us understand the botnet threat
- Measurement studies focused mainly on:
  - Botnet dynamics [E. Cooke. et. al., USENIX SRUTI 2005]
  - Global diurnal behavior of botnets using DNS sinkholing technique [D. Dagon. et. al., NDSS 2006]
  - The relationship between botnets and scanning/spamming activities [M. Collins. et. al., IMC 2007]
  - Examining the bot source code to provide an inside look at the botnets. Examples: analyzing the structural similarities, defense mechanisms, and command and control capabilities, of major bot families [P. Barford. et. al., Special workshop on Malware Detection]
  - Using honeypots to track botnets,[F. Freiling. et. al., ESORICS 2005], [Moheeb Abu Rajab, et.al, IMC 2006]

[Moheeb Abu Rajab, et.al, IMC 2006.]

Honeypot-based Detection- Main steps

- Acquiring and analyzing a copy of a bot
  - Using honeypots and special analysis software
- Infiltrating the Botnet by connecting to the IRC channel with a specially crafted IRC client
- Collecting information about means and techniques used by the Botnet

Measuring Botnets- Collecting Bot Binaries

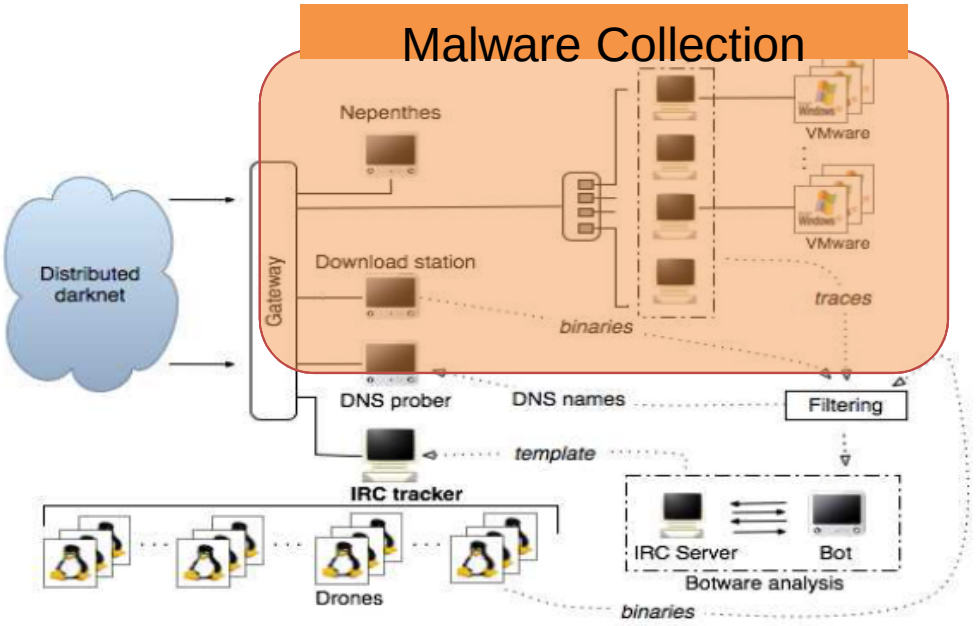


[Source: Moheeb Abu Rajab, et.al, IMC 2006.]

Darknet : Denotes an allocated but unused portion of the IP address space.

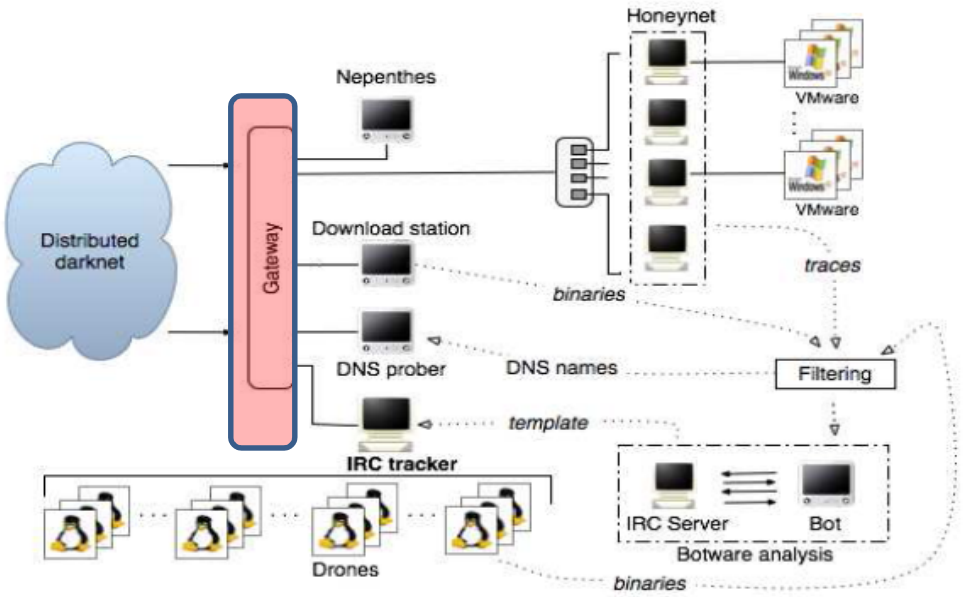
Malware Collection

- nepenthes mimics the replies generated by vulnerable services i order to collect the first stage expl (typically a Windows shellcode)
- Honeynets also used along with nepenthes
- Catches exploits missed by nepenthes
- Consists of number of honeypots running unpatched instances of Windows XP in a virtualized environment
- Infected honeypot compared with base to identify Botnet binary



Gateway

- Routing to different components
- Firewall : Prevent outbound attacks & self infection by honeypots
- Detect & Analyze outgoing traffic for infections in honeypot
- Only 1 infection in a honeypot
- Several other functions

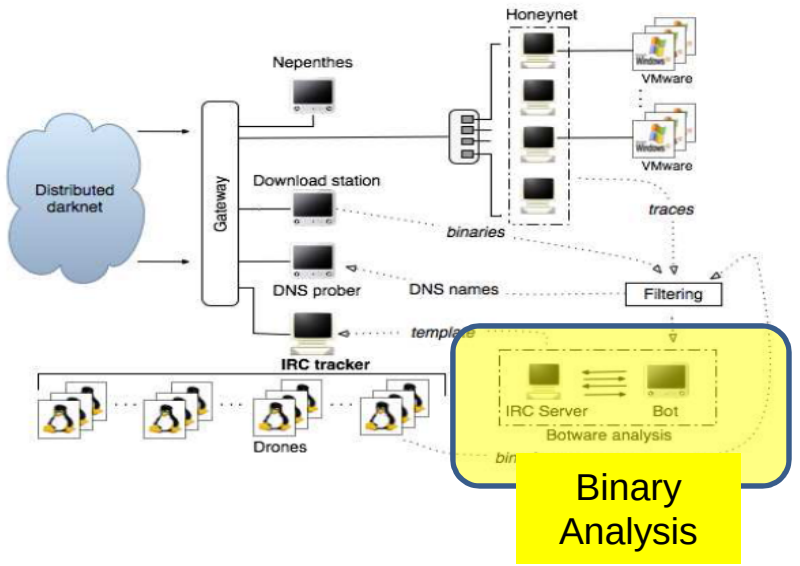


Binary Analysis

- Each collected binary is executed on a clean image of Windows XP instantiated as a virtual machine on the client
- Two phases are performed:
  - Phase 1: Creation of a network fingerprint:

    fnet = <DNS, IPs, Ports, scan>
  - Phase 2: Extraction of IRC-related features:

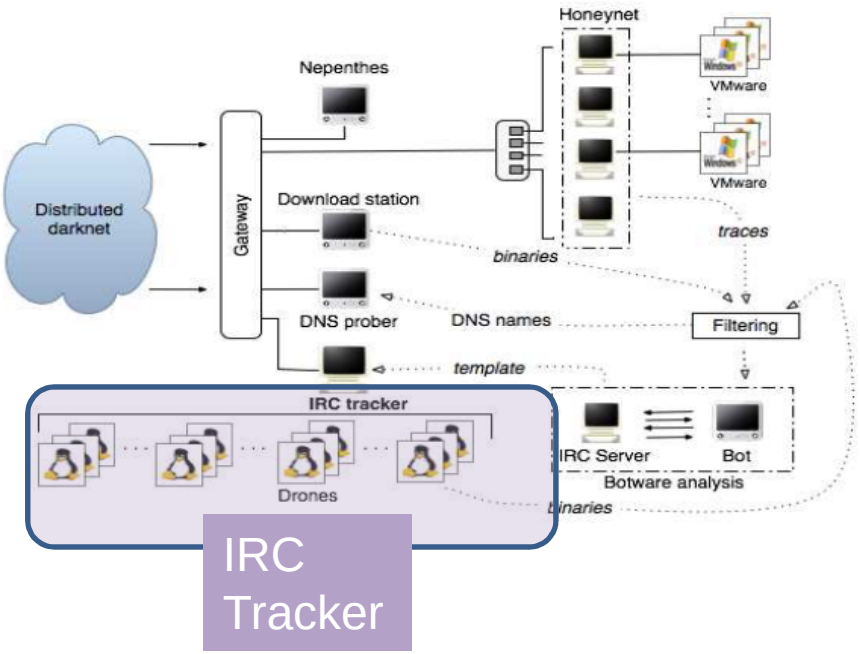
    firc = <PASS, NICK, USER, MODE, JOIN>



- IRC Server learns Botnet "dialect" - Template
- Learn how to correctly mimic bot's behavior - Subject bot to a barrage of commands

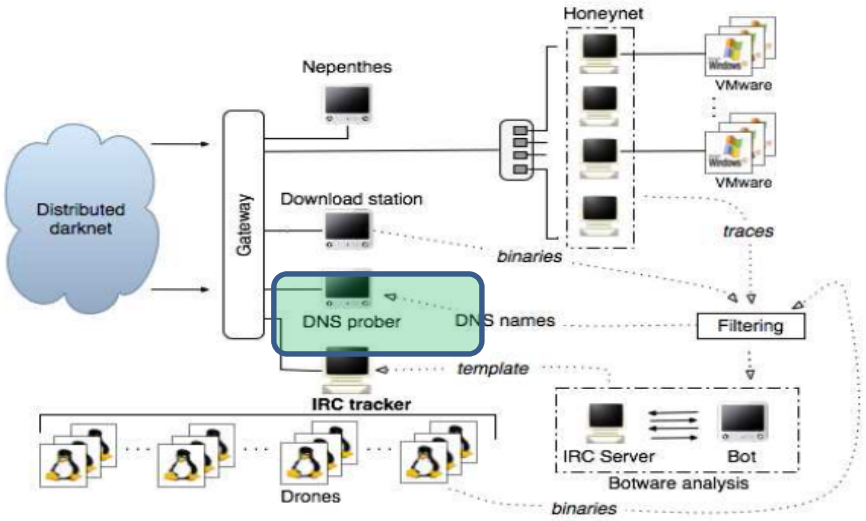
IRC Tracker (A view from within the Botnet)

- The IRC tracker (also called a *drone) is a modified IRC client* that can join a specified IRC channel and automatically answer directed
- queries based on the template Connect to real IRC server
- The drone operates in the wild, and pretends to dutifully follow any commands from the botmaster, and provides realistic responses to her commands
- Drones modified and used to act as IRC Client by the tracker to Cover



DNS Tracker

- Exploiting the fact that most bots issue DNS queries to resolve the IP addresses of their IRC servers Tracker uses DNS requests
- probe the caches of a large number of DNS servers in order to infer the footprint of a particular botnet, defined here as the total number of DNS servers giving cache hits
- A cache hit implies that at least one client machine has queried the DNS server within the lifetime (TTL) of its DNS entry
- Has 800,000 entries after reduction



Limitations of Honeypot based detection

- Low-interaction honeypots such as Nepenthes [13] can capture attacks from only a limited number of known exploits that they faithfully emulate
- Honeypots are mainly designed to capture malware that propagates via scanning for remote vulnerabilities
- There is no guarantee on the frequency or volume of malware captured using this approach because
- Malware may avoid scanning the networks with “known” honeypots [17], and it can detect virtual machine environments commonly used for honeypots
- Honeypots report infections on only their decoy machines; they generally cannot directly tell which non-decoy machines in the enterprise network are members of a botnet.

Talk Outline – Module IV

- Honeypot-based Detection
- **Hueristic-Based Detection**
- Traffic Analysis-based Detection

Heuristic-based Botnet Detection

- Combining both IRC statistics and TCP work weight (i.e., anomaly scanning activity) for detecting IRC-based otnets [J. R. Binkley. et. al., USENIX SRUTI 2006]
  - This approach is useful only for detecting certain botnet instances, i.e., IRC bots that perform scanning

- Signature-based IRC botnet detection systems that matches known nickname patterns of IRC bots[J. Goebel. et. al., USNIX HotBots 2007]

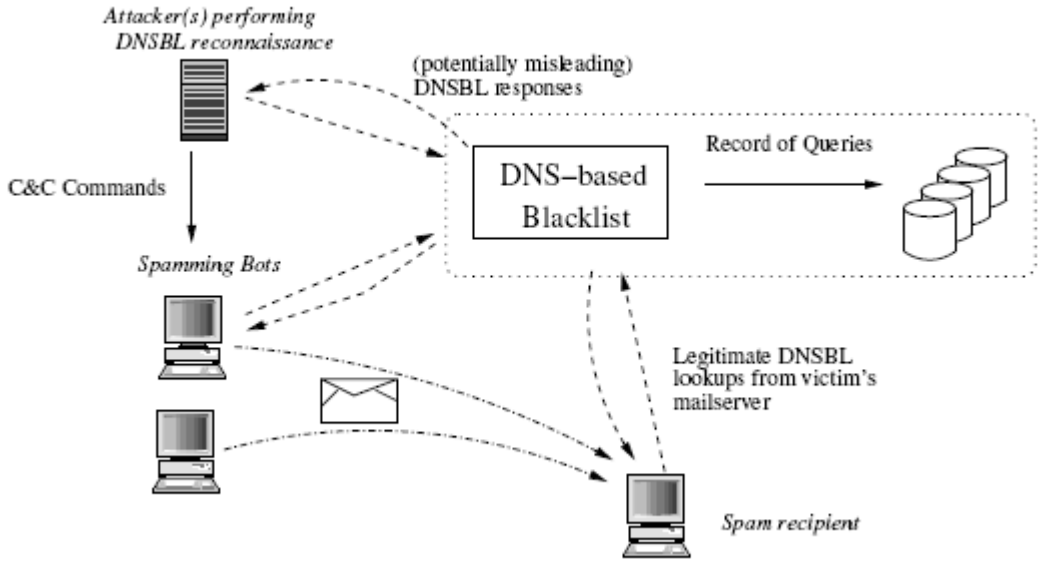
- Using DNSBL (DNS blacklist) counter-intelligence to locate botnet members that generate spam [A. Ramachandran. et. al., USENIX SRUTI 2006]

[A. Ramachandran. et. al., USENIX SRUTI 2006]

DNS Blacklisting

- Many Internet Service Providers (ISPs) and enterprise networks use DNSBLs to track IP addresses that originate spam
  - Future emails sent from these IP addresses can be rejected
- Botmasters are known to sell clean bots (i.e., not listed in any DNSBL) at a premium
- Botmasters themselves must perform reconnaissance lookups to determine their bots blacklist status
  - It is possible to perform counter intelligence to discover bot identities

DNSBL-based Spam Mitigation Architecture



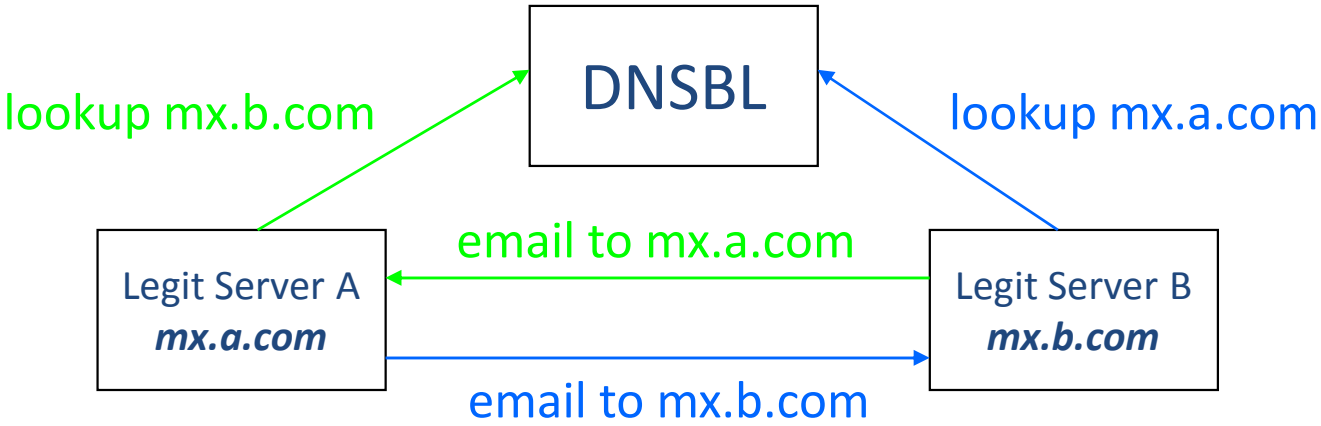
[Source: A. Ramachandran. et. al., USENIX SRUTI 2006]

Detecting Reconnaissance

- Key Requirement: Distinguish reconnaissance queries from queries performed by legitimate mail servers
- The Solution: Develop heuristics based on the spatial and temporal properties of a DNSBL Query Graph
- Two heuristics
  - spatial heuristic
  - Temporal heuristic

Hurietics

- Spatial Heuristic: Legitimate mail servers will perform queries and be the object of queries.

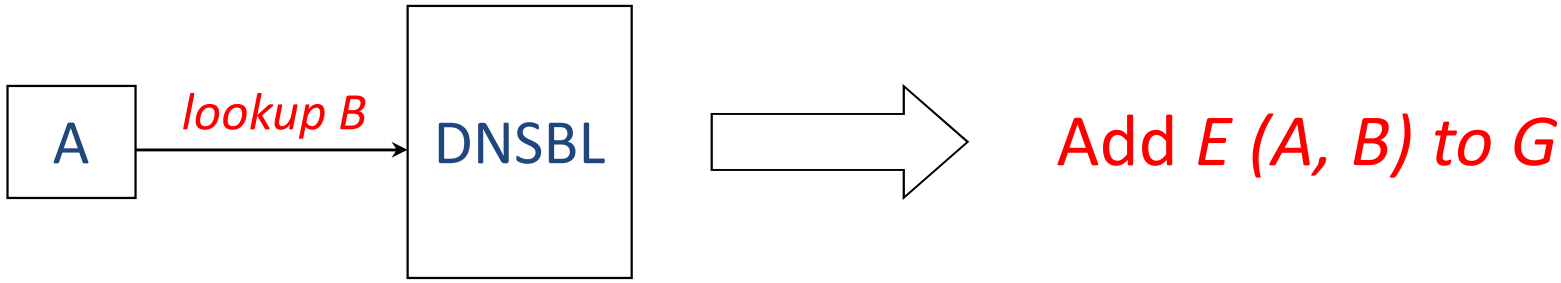


  - Hosts issuing reconnaissance queries usually will not be queried

- Temporal Heuristic: Legitimate lookups reflect arrival patterns of legitimate email

Applying the Spatial Heuristic

- Construct the directed DNSBL Query Graph G



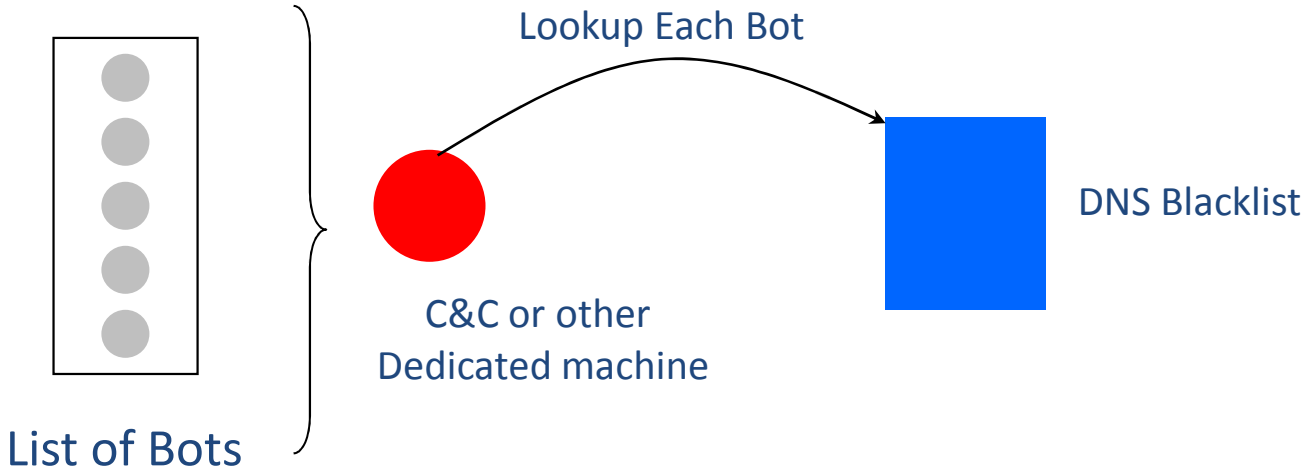
- Extract nodes (and their connected components) with the highest values of the spatial metric λ, where λ = (Out-degree/In-degree)

Reconnaissance Techniques

- Third-party reconnaissance
- Self-reconnaissance
- Distributed reconnaissance

Third-Party Reconnaissance

- Third-party performs reconnaissance query



- Relatively easy to detect using the spatial metric

Other Techniques

- Self-Reconnaissance
  - Each bot looks itself up
  - This should not happen normally (at least, not en-masse)
    – thus, easy to detect
- Distributed Reconnaissance
  - Bots perform lookups for other bots
  - Complex to deploy and operate

Talk Outline – Module IV

- Honeypot-Based Detection
- DNS Black List-Based Detection
- **Traffic Analysis -Based Detection**

Traffic Analysis Based-Botnet Detection

- Inspect network traffic traces looking for Botnet footprints
- Traffic that follows certain pattern or exhibits specific behavior is classified as Botnet traffic
- Usually not able to detect emerging Botnet types

Traffic Analysis Based-Botnet Detection--Examples

- Bothunter: regardless of the C&C structure and network protocol, if they follow pre-defined infection live cycle

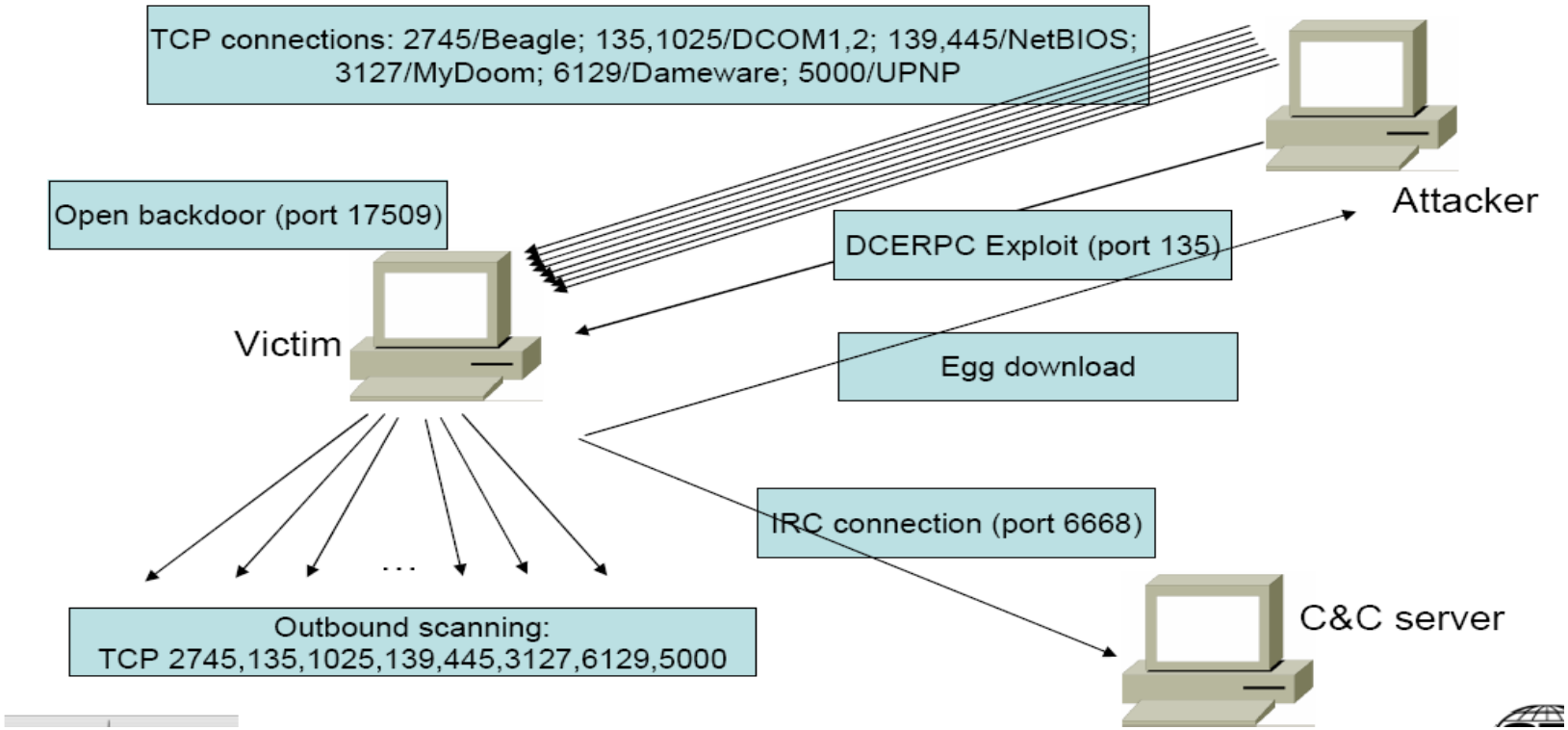
- Botsniffer:works for IRC and http, can be extended to detect centralized C&C botnets

[G. Gu. et. al., Usenix 2007]

BotHunter system-detection on single infected client

- Detecting Malware Infection Through IDS-Driven Dialog Correlation
- Monitors **two-way communication** flows between internal networks and the Internet for signs of bot and other malware
- Correlates dialog trail of inbound intrusion alarms with outbound communication patterns

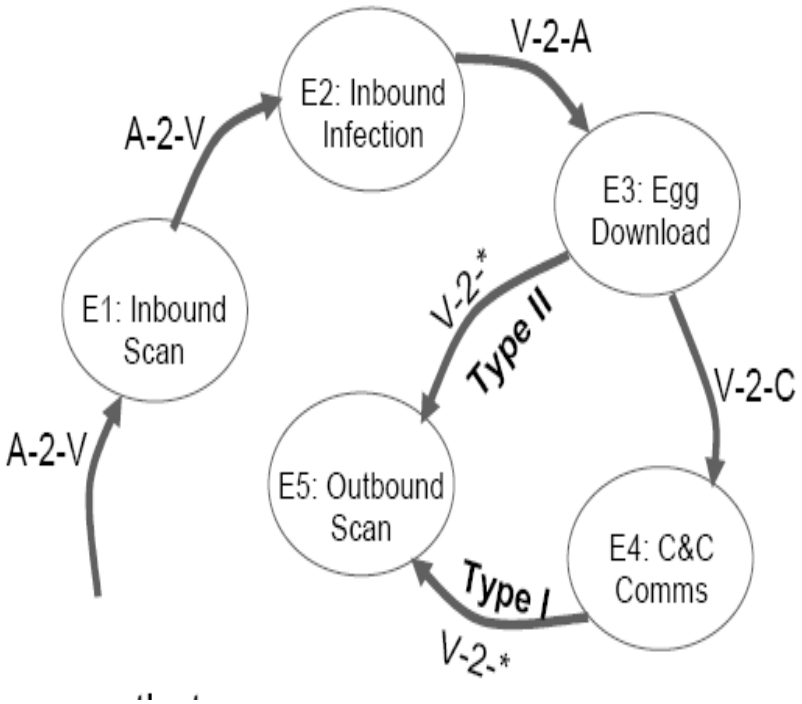
Bot infection case study: Phatbot



[Source: G. Gu. et. al., Usenix 2007]

Dialog-based Correlation

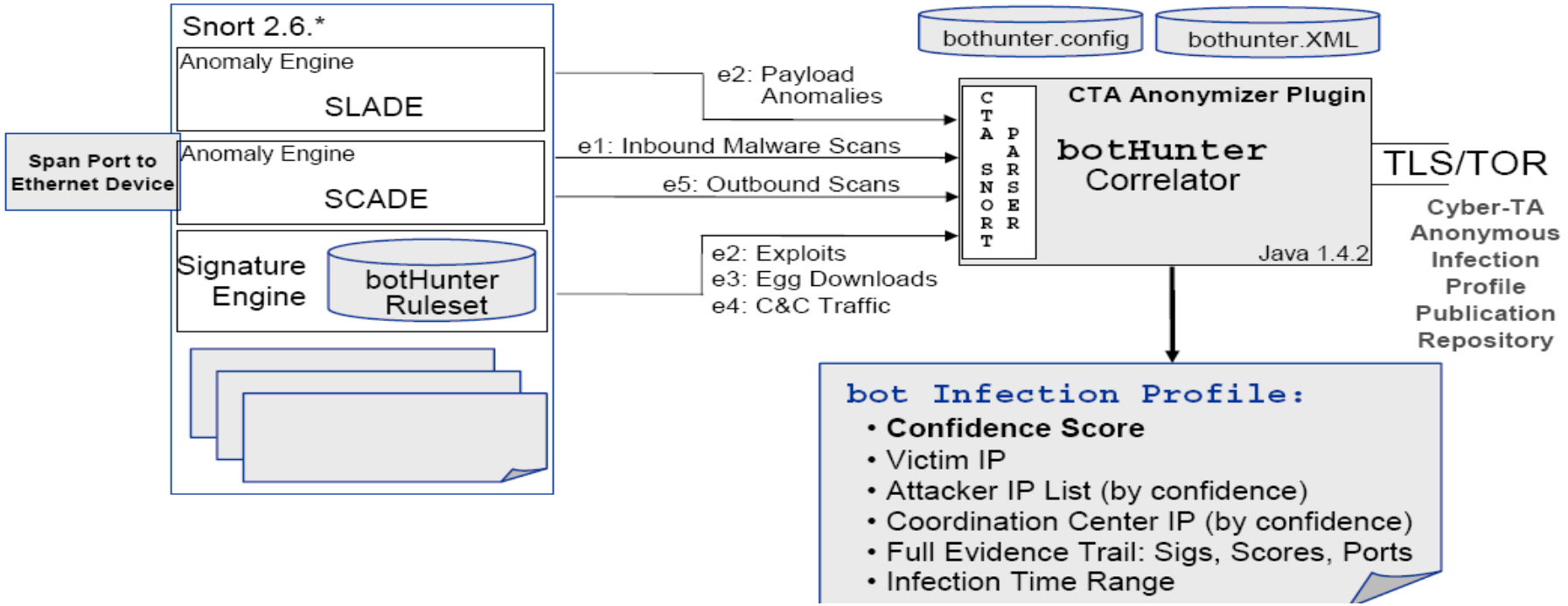
- BotHunter employs an ***Infection Lifecycle Model*** to detect host infection behavior



- E1: External to Internal Inbound Scan
- E2: External to Internal Inbound Exploit
- E3: Internal to External Binary Acquisition
- E4: Internal to External C&C Communication
- E5: Internal to External Outbound Infection Scanning

[Source: G. Gu. et. al., Usenix 2007

Bothunter Architecture

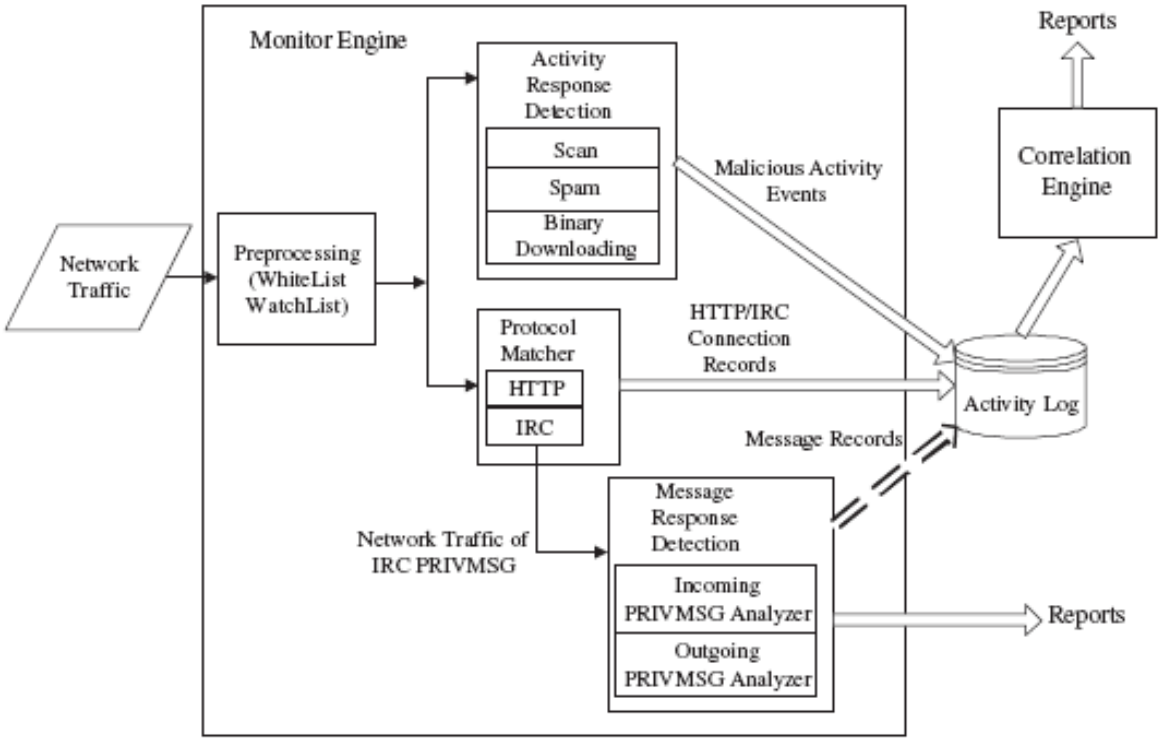


[G. Gu. et. al., Usenix 2008]

BotSniffer- Detecting IRC and HTTP based Botnets

- A network anomaly based botnet detection system
- Explores the spatial-temporal correlation and similarity of Botnet C&C
- Based on the intuition that since bots of the same botnet run the same bot program, they are likely to respond to the botmaster's commands in a similar fashion
- Employs several correlation and similarity analysis algorithms to identify botnet traffic

BotSniffer Architecture



Correlation Engine

- Based on two properties
- Response crowd
  - a set of clients that have (message/activity) response behavior

  -A Dense response crowd: the fraction of clients with message/activity behavior within the group is larger than a threshold (e.g., 0.5).
- A homogeneous response crowd
  - Many members have very similar responses

Revisit Botnet Definition

- “A coordinated group of malware instances that are controlled by a botmaster via some C&C channel”
- We need to monitor two planes
  - C-plane (C&C communication plane): “who is talking to whom”
  - A-plane (malicious activity plane): “who is doing what”

C-Plane clustering

- Temporal related statistical distribution information in
  - BPS (bytes per second)
  - FPH (flow per hour)
- Spatial related statistical distribution information in
  - BPP (bytes per packet)
  - PPF (packet per flow)

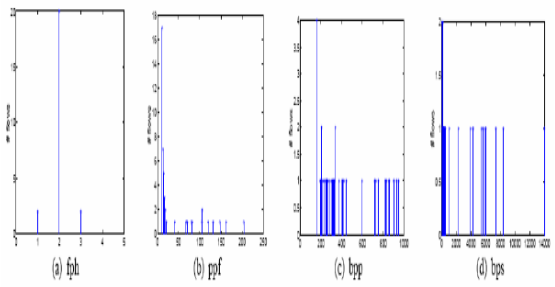


Figure 4: Visit pattern (shown in distribution) to Google from a randomly chosen normal client.

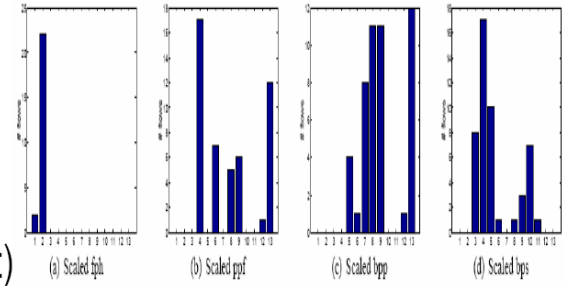


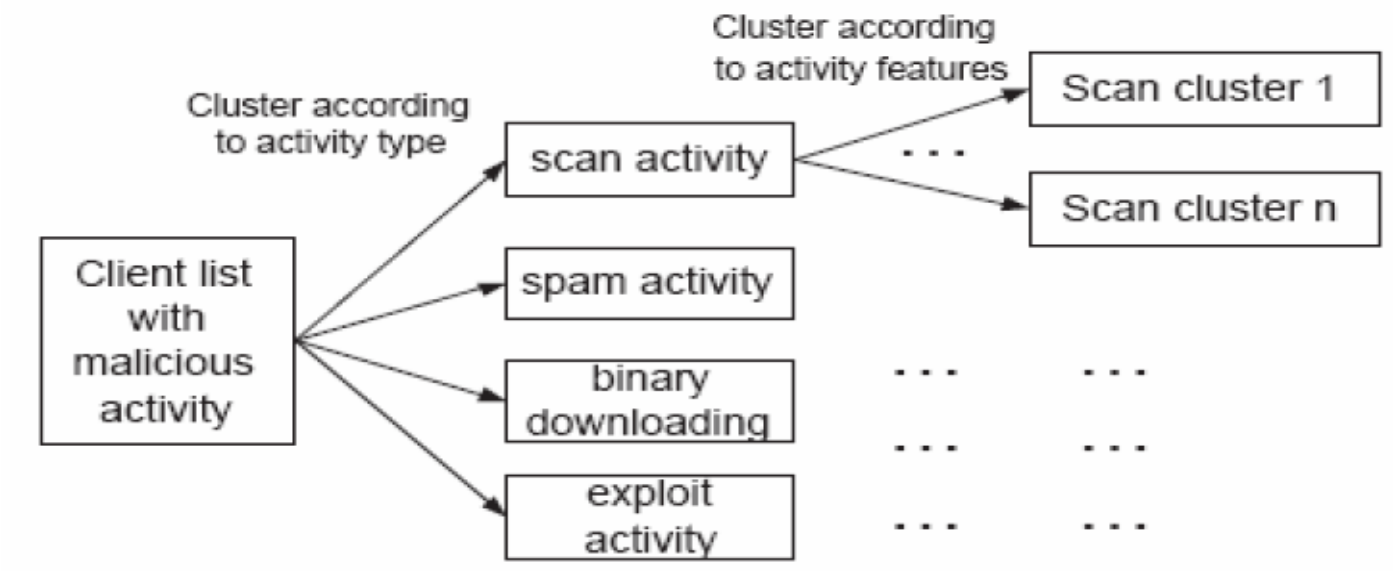
Figure 5: Scaled visit pattern (shown in distribution) to Google for the same client in Figure 4.

- What characterizes a communication flow (Cflow)

between a local host and a remote service?

– <protocol, srcIP, dstIP, dstPort>

A-plane clustering



Cross-clustering

- Two hosts in the same A-clusters and in at least one common C-cluster are clustered together

References

- COOKE, E., JAHANIAN, F., and MCPHERSON, D., “The zombie roundup: Understanding, detecting, and disrupting botnets,” in Proceedings of USENIX SRUTI’05, 2005.
- DAGON, D., ZOU, C., and LEE, W., “Modeling botnet propagation using timezones,” in Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS’06), February 2006
- COLLINS, M., SHIMEALL, T., FABER, S., JANIES, J., WEAVER, R., SHON, M. D., and KADANE, J., “Using uncleanliness to predict future botnet addresses,,” in Proceedings of ACM/USENIX Internet Measurement Conference (IMC’07), 2007
- BARFORD, P. and YEGNESWARAN, V., “An Inside Look at Botnets.” SpecialWorkshop on Malware Detection, Advances in Information Security, Springer Verlag, 2006
- FREILING, F., HOLZ, T., and WICHERSKI, G., “Botnet Tracking: Exploring a Root-cause Methodology to Prevent Denial of Service Attacks,” in Proceedings of 10th European Symposium on Research in Computer Security (ESORICS’05), 2005

References (Contd.)

- BINKLEY, J. R. and SINGH, S., "An algorithm for anomaly-based botnet detection," in Proceedings of USENIX SRUTI'06, pp. 43–48, July 2006.
- RAMACHANDRAN, A., FEAMSTER, N., and DAGON, D., "Revealing botnet membership using DNSBL counter-intelligence," in Proceedings of USENIX SRUTI'06, 2006.
- GOEBEL, J. and HOLZ, T., "Rishi: Identify bot contaminated hosts by irc nickname evaluation," in Proceedings of USENIX HotBots'07, 2007
- GU, G., ZHANG, J., and LEE, W., "BotSniffer: Detecting botnet command and control channels in network traffic," in Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08), 2008

References (contd.)

- YEN, T.-F. and REITER, M. K., “Traffic aggregation for malware detection,” in Proceedings of the Fifth GI International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA’08), 2008
- STRAYER, W. T., WALSH, R., LIVADAS, C., and LAPSLEY, D., “Detecting botnets with tight command and control,” in Proceedings of the 31st IEEE Conference on Local Computer Networks (LCN’06), 2006
- KARASARIDIS, A., REXROAD, B., and HOEFLIN, D., “Wide-scale botnet detection and characterization,” in Proceedings of USENIX HotBots’07, 2007
- GU, G., PERDISCI, R., ZHANG, J., and LEE, W., “BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection,” in Proceedings of the 17th USENIX Security Symposium (Security’08), 2008
- GU, G., PORRAS, P., YEGNESWARAN, V., FONG, M., and LEE, W., “BotHunter: Detecting malware infection through ids-driven dialog correlation,” in Proceedings of the 16th USENIX Security Symposium (Security’07), 2007

Thank You !