Botnet Detection and Response

The Network is the Infection

David Dagon

dagon@cc.gatech.edu
Georgia Institute of Technology
College of Computing

OARC Workshop, 2005

Outline

Georgia Tech Campus
(Cross Sectional View)

based on joint work with:

- *UMass CS:* Cliff Zou
- *GaTech CS:* Sanjeev Dwivedi, Robert Edmonds, Wenke Lee, Richard Lipton, and Merrick Furst
- *GaTech ECE:* Julian Grizzard

Outline

Definition: Bots
Hard to Define; Easy to Detect

Definitions, Examples

- Definition: autonomous programs automatically performing tasks, absent a real user.
- Benign bots
  - countless examples at http://www.botknowledge.com/
- Gray-area bots
  - Blogbots, e.g., wikipedia, xanga Note: http://en.wikipedia.org/wiki/Wikipedia:Bots
  - Other examples: xdcc, fserve bots for IRC
  - Trainer bots (MMORPGs)
- Malicious bots
  - Key characteristics: process forking, with network and file access, and propagation potential.

Definition: Botnets

Bot*nets*: Also hard to define

- Definition: networks of autonomous programs capable of acting on instructions.
- Again, gray areas: FServe bot farms, spider farms, etc.
- Today, just a narrow definition:
  - organized network of malicious bot clients

Key Insights

- The network is the infection.
- We must track bot*nets*, not just bots

Definition: Botnets

Bot*nets*: Also hard to define

- Definition: networks of autonomous programs capable of acting on instructions.
- Again, gray areas: FServe bot farms, spider farms, etc.
- Today, just a narrow definition:
  - organized network of malicious bot clients

Key Insights

- The network is the infection.
- We must track bot*nets*, not just bots

Botnets as a Root Cause

Botnets are a Root Problem

- Spam bots
- Click fraud
- Large-scale identity theft; “vicpic” sites
- Proxynets (for launching other attacks)

Lightning Attacks

The short vulnerability-to-exploitation window makes bots particularly dangerous.

– Emerging Cybersecurity Issues Threaten Federal Information Systems, GAO-05-231

Botnets as a Root Cause

Botnets are a Root Problem

- Spam bots
- Click fraud
- Large-scale identity theft; “vicpic” sites
- Proxynets (for launching other attacks)

Lightning Attacks

The short vulnerability-to-exploitation window makes bots particularly dangerous.

– Emerging Cybersecurity Issues Threaten Federal Information Systems, GAO-05-231

Botnet vs Bot Detection

What's the Difference?

Why track both bots and botnets?

Bot Detection Benefits

- RE $\rightarrow$ signature IDS (content)
- *Partial* victim identification
  - Response Policy: RBL, Quarantine
  - Host vulnerability analysis

Botnet vs Bot Detection

What's the Difference?

Why track both bots and botnets?

Bot*net* Detection Benefits

- Critical Infrastructure Protection, prioritize on harm to *network*, not just victims.
- RE $\rightarrow$ signature IDS (flows)
- *More Complete* victim identification
  - Remediation Policies: Windows 2003 Network Access Protection (NAP), ISP quarantines

Botnet Propagation I

email

- Requires user interaction, social engineering
- Easiest method; common.
- Interesting: pidgin English affects propagation.

instant message

- Various: social eng., file xfer, vulnerabilities

Botnet Propagation II

remote software vulnerability

- Often, no interaction needed
- Predator, Prey and Superpredator: worms vs. worms (dabber)

web page

Plain vanilla malware, or even Xanga ghetto botnets

“seed” botnets

- Botnets create botnets.
- Used for upgrades.
- *Most significant for detection*

Command and Control Taxonomy

Goals:

- Anticipate future botnet structures
- Taxonomy of botnet controls

> *An "important and sensible goal for an attack taxonomy ... should be to help the defender" – R. Maxion*

Thus, create a taxonomy based on detection opportunities, instead of random bot/botnet characteristics.

Command and Control Taxonomy

Resources

- Public, private
- Botmaster's administrative control over a resource

Rallying Services

1. Medium used for rallying
2. E.g., HTTP, IRCd, DNS tunnel, etc.
3. Reminder: public and private versions of the above

Command and Control Taxonomy

Resources (cont'd)

- Public, private
- Botmaster's administrative control over a resource

Name Services

1. `hosts(5)`, e.g., corrupting `WINDOWS/system32/drivers/etc/hosts`
2. DNS, public and private
3. DDNS, public/private
4. Hit lists

Command and Control Taxonomy I

RFC Compliance

The degree of standards compliance.

- E.g., non-responsive IRCd
- Ad-hoc protocols.
  - P2P
  - port-knocking
  - Tunneling (NSTx, sinit, bobax)

Command and Control Taxonomy II

Activity Level

The degree to which bots are in constant contact with botmaster.

- *Time division:* periodic phone in, flow-based, sessionless, stateless
- *Proximity:* delegation of contact; clique connections

Insight

Note: other lists possible. Key: organize them into categories. Can we detect these *categories*?

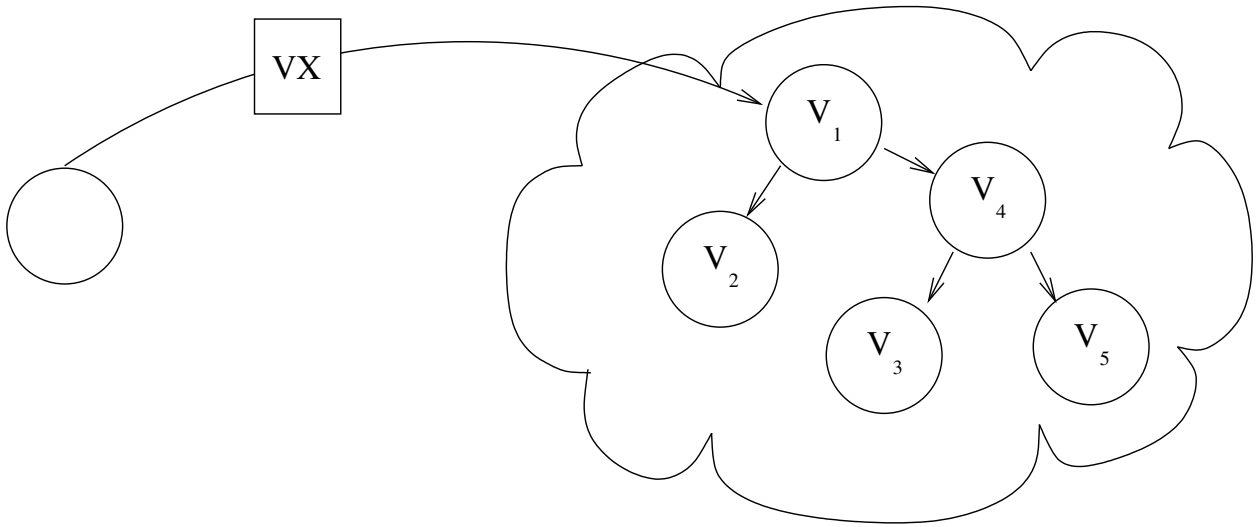
The Rallying Problem

Let's focus on "rallying" to identify detection opportunities.

- C&C used to rally victims
  - Detecting C&C $\Rightarrow$ detecting botnet
  - Goal: detect C&C *during* formation
- Therefore, reason like an attacker
- Attacker design goals:
  - Robustness
  - Mobility
  - Stealth
- Assumption: The attackers are always motived by these three goals.

The Rallying Problem

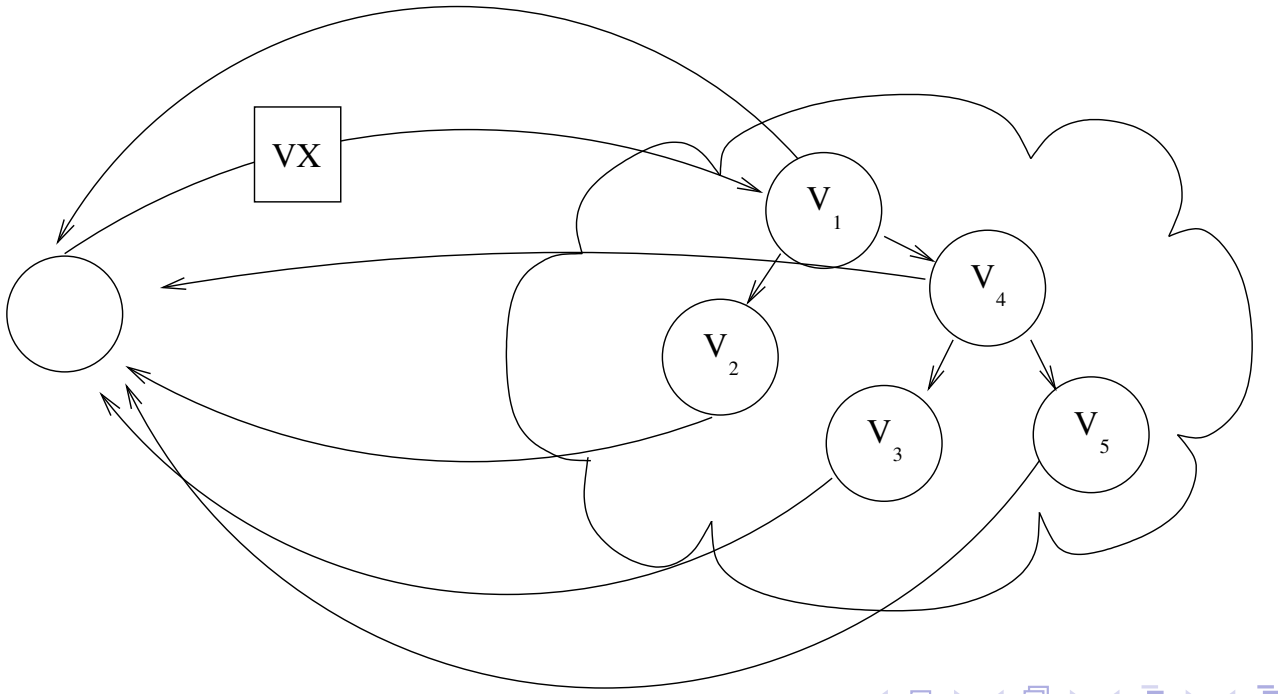
- Suppose we create virus
  - Download vx code; fiddle; compile
  - Uses email propagation/social engr.
- We mail it...



- Welcome to the 1980s. What if we want to *use* victim resources?

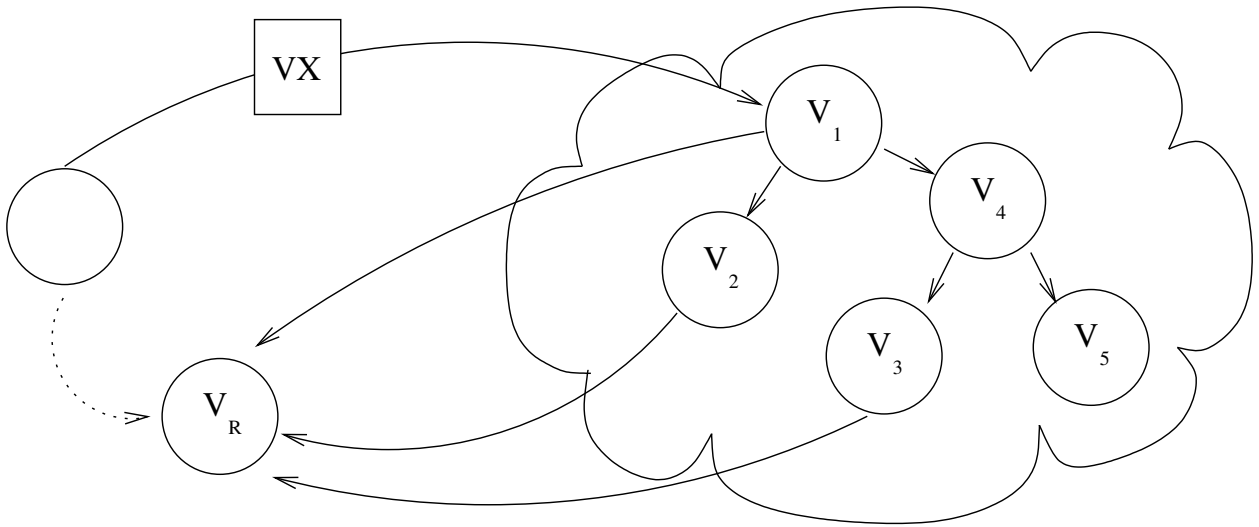
Simple Rallying I

- Naively, we could have victims contact us...
- Problems
  - VX must include author's address (no stealth)
  - Single rallying point (not robust)
  - VX has hard-coded address (not mobile)



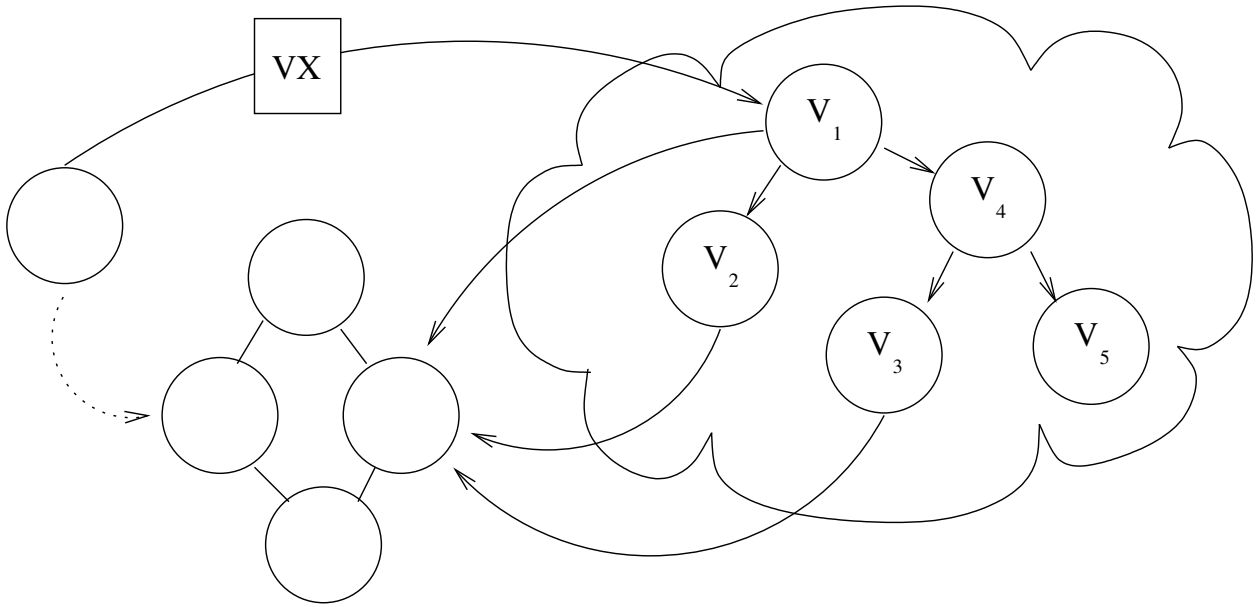
Simple Rallying II

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



Simple Rallying III

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



Detection In-Protocol

Numerous ad-hoc bot detection frameworks:

- IRCd, public (DDD, Gnuworld)
- IRCd, private (RWTH Aachen)
- E-mail (CipherTrust ZombieMeter; everyone else)
- AV/Managed network sensing (Sophos)
- Obvious detection (existing blackhole mining)

Problem:

- Largely post-attack
- Largely cannot detect structure (rain drop analogy)
- Expensive to monitor (requires spam filter banks, or difficult IRCd manipulations)
- Trivially evaded

Detection Strategies

What should we do instead of in-protocol sensing?

- Better approach: find invariant observable by sensors
- Bot must always exhibit some behaviors
- If we can sense, we can perform detection

One idea: DNS-based detection

Protocol Agnostic Detection: DNS

Intuition

www.example.com/products
www.example.com/home
botnet1.example.org
botnet2.example.org

$\overbrace{\text{3LD}}^{\textit{class 1}}$.SLD.TLD/$\underbrace{\text{subdir1/subdir2}}_{\textit{class 2}}$

Incentives for Subdirectories

- lower skills (dns updates vs `mkdir`)
- less risk (fewer $ transactions)
- lower cost (package 3LD deals)

Detecting DDNS Bots

Canonical DNS Request Rate

$$C_{SLD_i} = R_{SLD_i} + \sum_{j=1}^{|SLD_i|} R_{3LD_j}$$

This is analogous to summing the children for a tree rooted on SLD_i.

Key Assumption

DNS server is not authoritative for many zones with high 3LD count.

$\rightarrow$ Dyn DNS Providers!

Detecting DDNS Bots

Canonical DNS Request Rate

$$C_{SLD_i} = R_{SLD_i} + \sum_{j=1}^{|SLD_i|} R_{3LD_j}$$

This is analogous to summing the children for a tree rooted on SLD_i.

Key Assumption

DNS server is not authoritative for many zones with high 3LD count.
→ Dyn DNS Providers!

Detecting DDNS Bots

Canonical DNS Request Rate

$$C_{SLD_i} = R_{SLD_i} + \sum_{j=1}^{|SLD_i|} R_{3LD_j}$$

This is analogous to summing the children for a tree rooted on SLD_i.

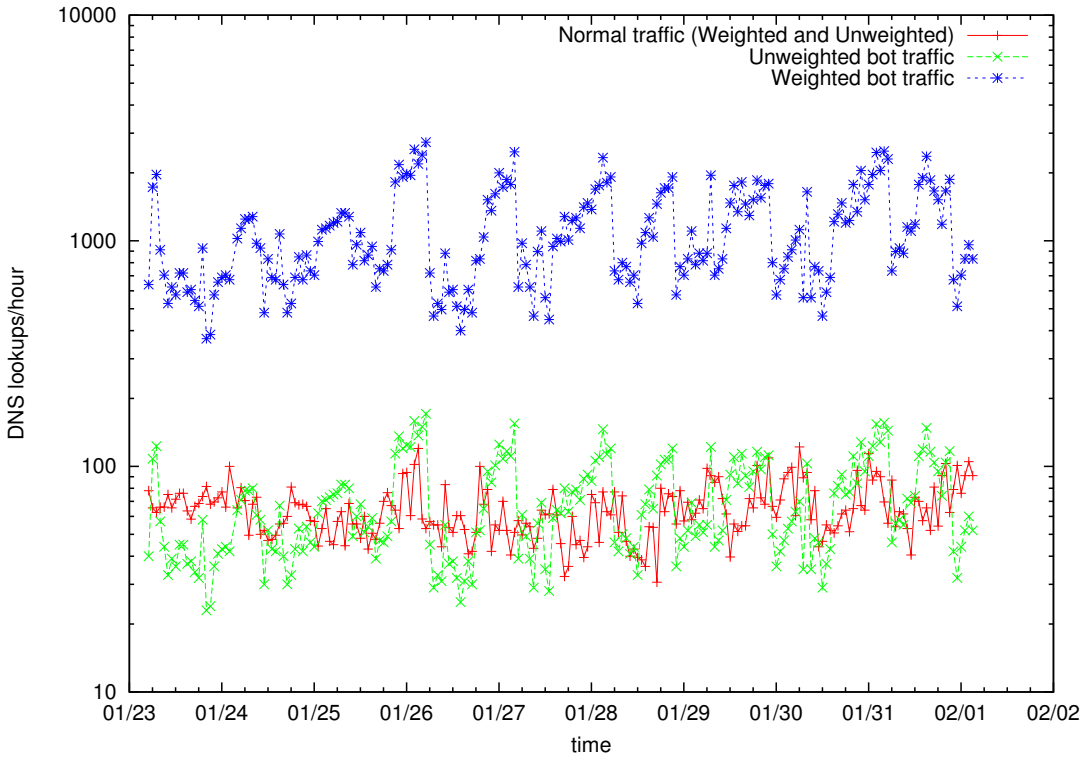
Use Chebyshev's inequality:

$$P(|X - \mu| \geq t) \leq \frac{\sigma^2}{t} \quad (1)$$

This is analogous to summing the children for a tree rooted on SLD_i.

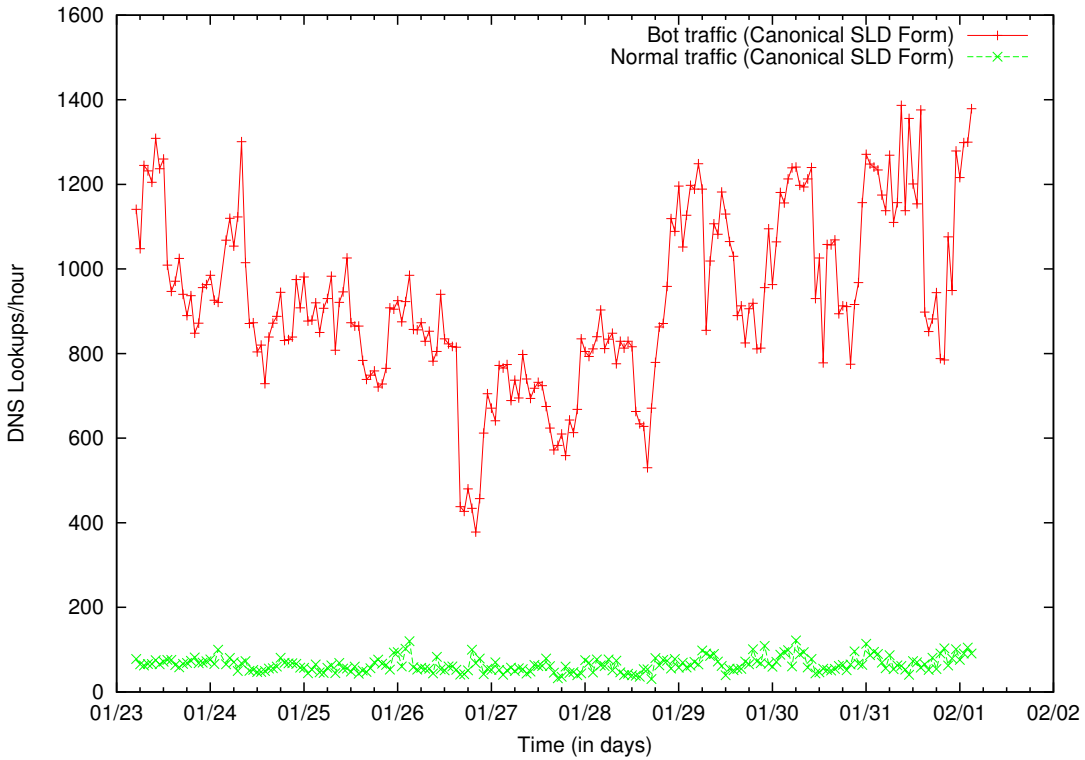
DDNS-Based Detection

- For DDNS customers, botnets tend to use subdomains; legitimate directories use subdirectories
- We can use SLD/3LD-ratios to identify botnet traffic



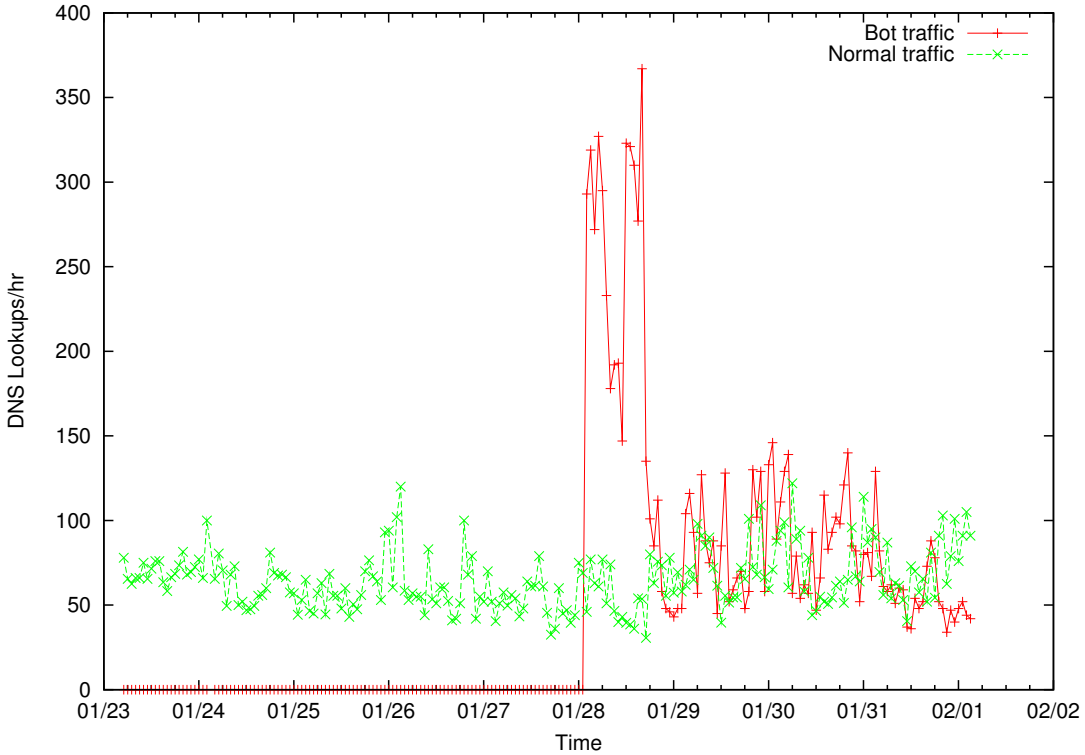
DDNS-Based Detection

- For DDNS customers, botnets tend to use subdomains; legitimate directories use subdirectories
- We can use SLD/3LD-ratios to identify botnet traffic



Detecting DDNS Bots

Does Chebyshev's inequality always work?



Detecting DDNS Bots

DNS Density Comparison

$$d^2(x, \bar{y}) = (x - \bar{y})' C^{-1} (x - \bar{y}) \quad (2)$$

- variable vectors (features):
  - x - new observation
  - $\bar{y}$ - trained normal profile
- C – inverse covariance matrix for each member of training data

Detecting DDNS Bots

Simplified Distance Measure

- Mahalanobis distance considers variance and average request rate‘
  - Thus, good for outlier detection
- We can assume independence of each feature in normal
  - DNS requests more likely not correlated
  - Thus, drop covariance matric C
  - Also done in Wang, Stolfo, etc.

$$d(x, \bar{y}) = \sum_{i=0}^{n-1} \left(\frac{|x_i - \bar{y}_i|}{\bar{\sigma}_i} \right) \tag{3}$$

Detecting DDNS Bots

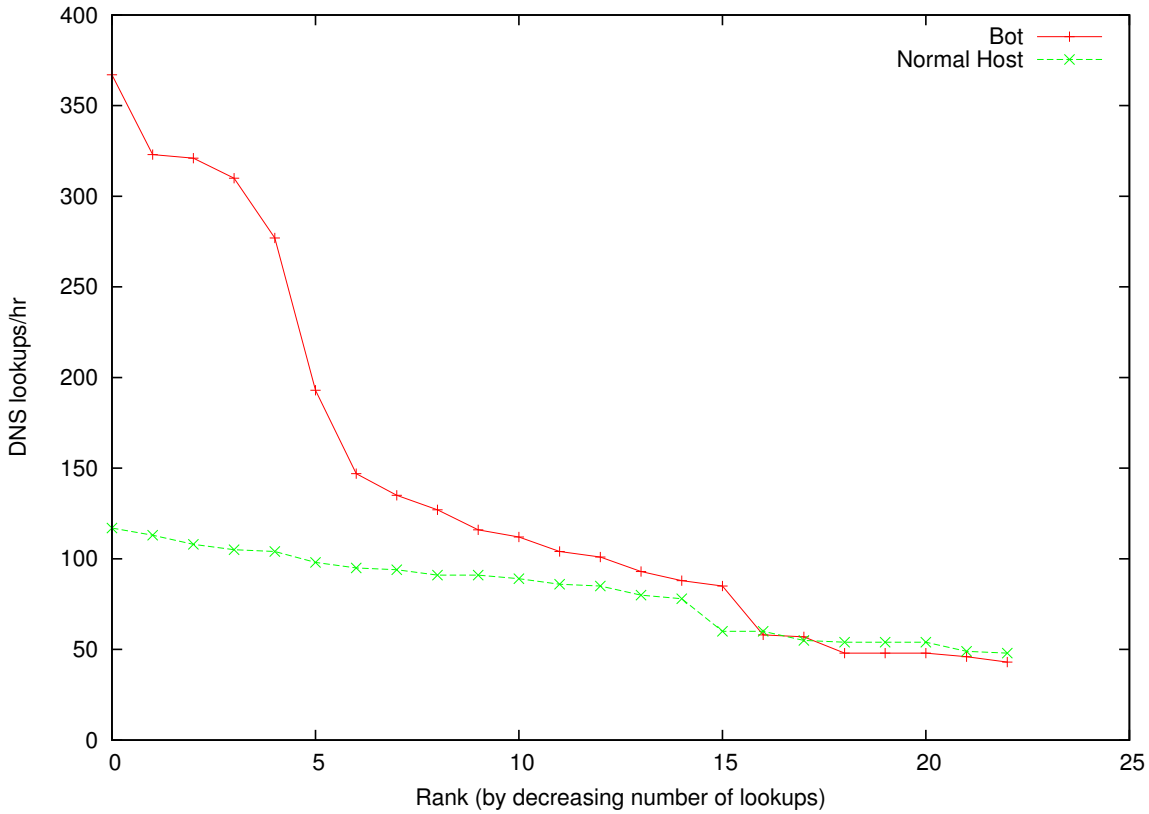
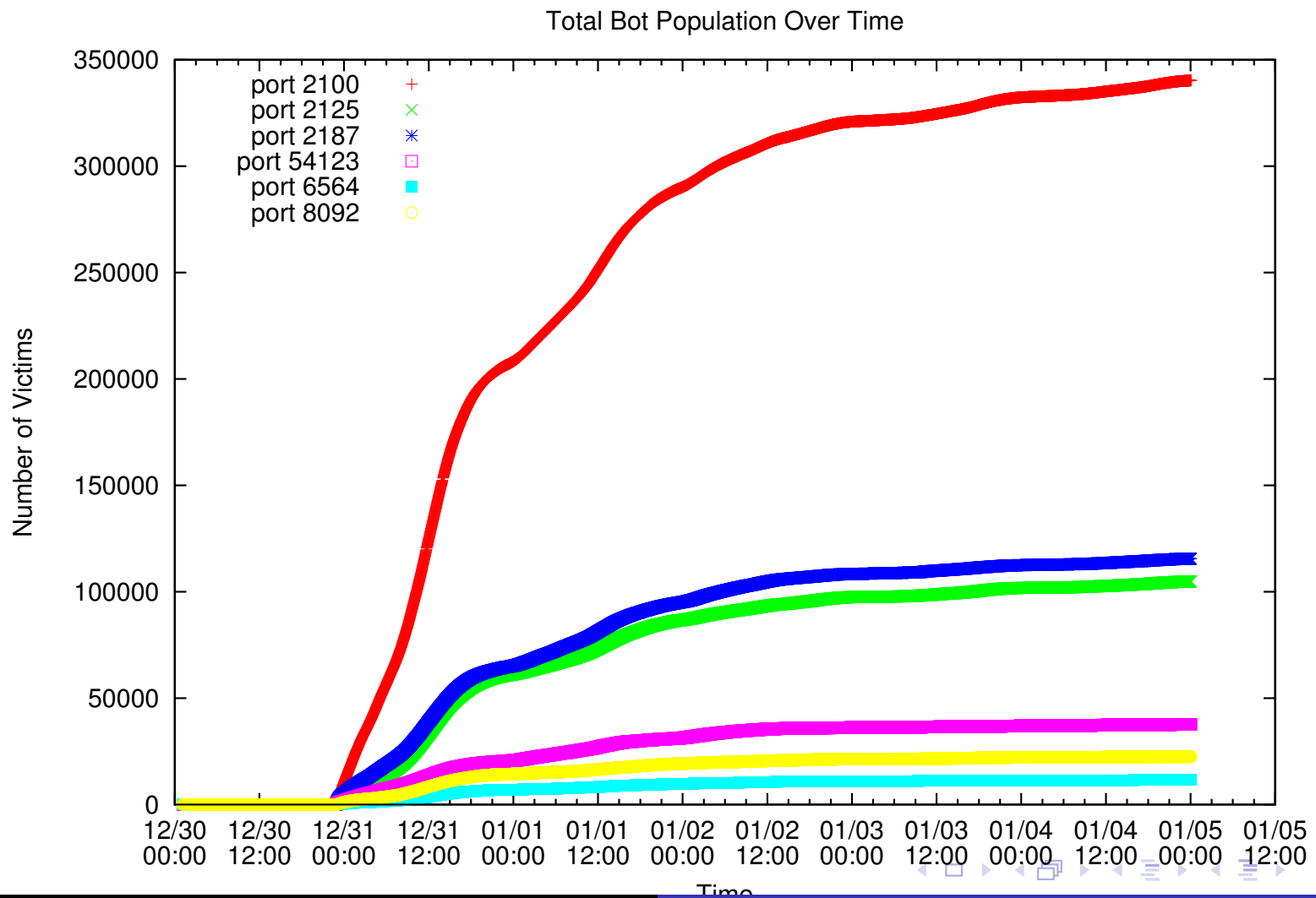


Figure: Comparison of Sorted DNS Rates

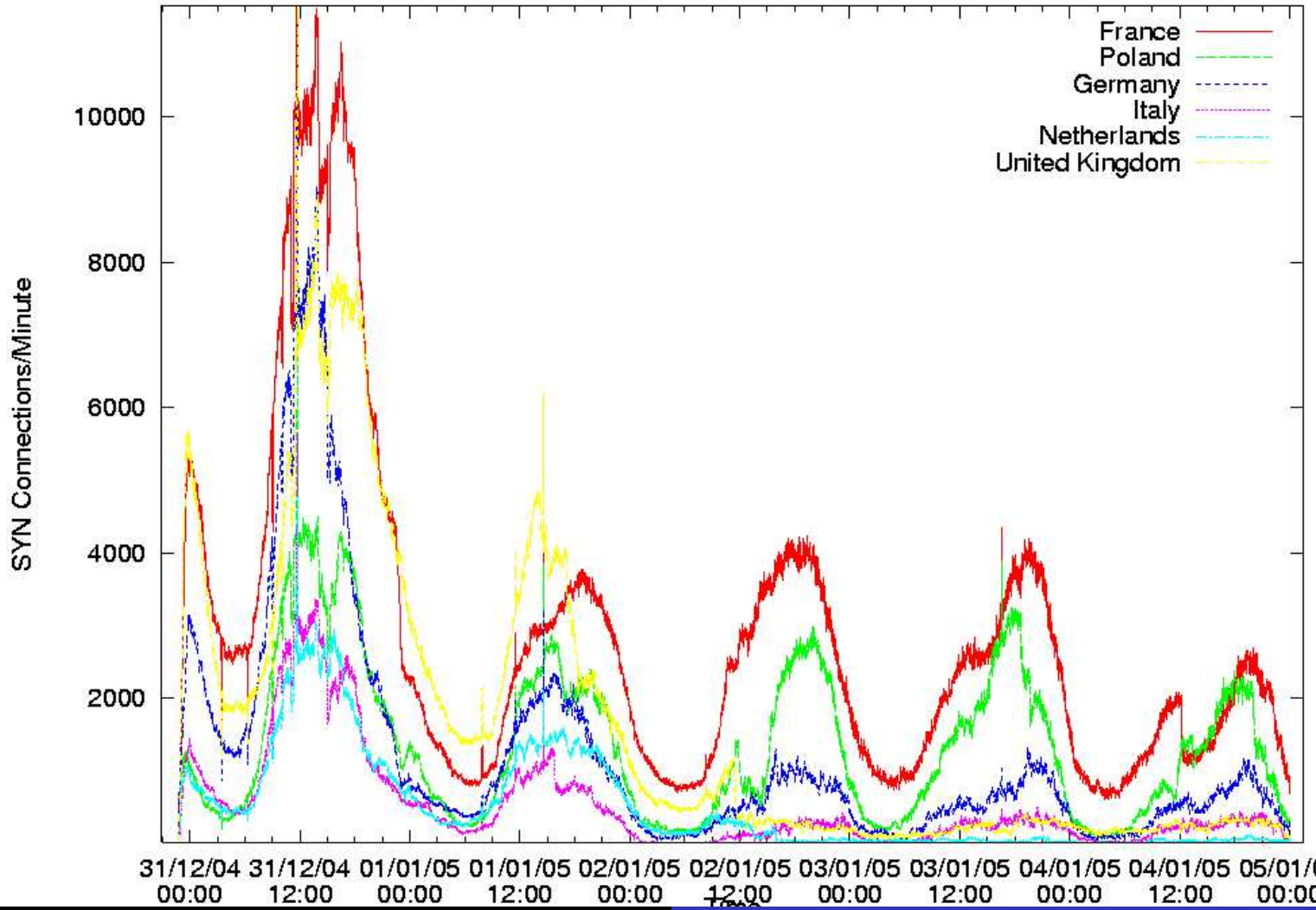
Response Options

- Response options include:
  - DNS Removal
  - Passive Logging (blackhole)
  - Passive Monitoring (sinkhole)
    - TCP-layer 4 timeout games
    - Application-layer delays
  - Interactive Monitoring
    - Proxynet/Man-in-middle
    - Fingerprinting hosts: clock skew, OS services, IP, time, etc.
    - Bot Application versioning
    - Removal interactions (Caution!)
- For today: victim epidemiology, and sinkholing

Victim Epidemiology: Total Population

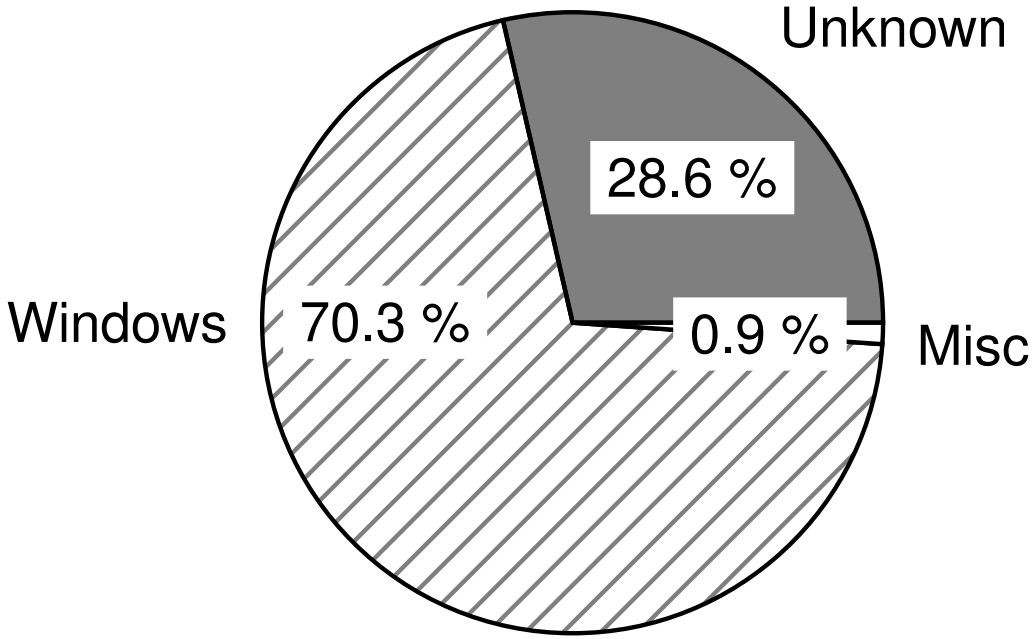


Victim Epidemiology: Country of Origin

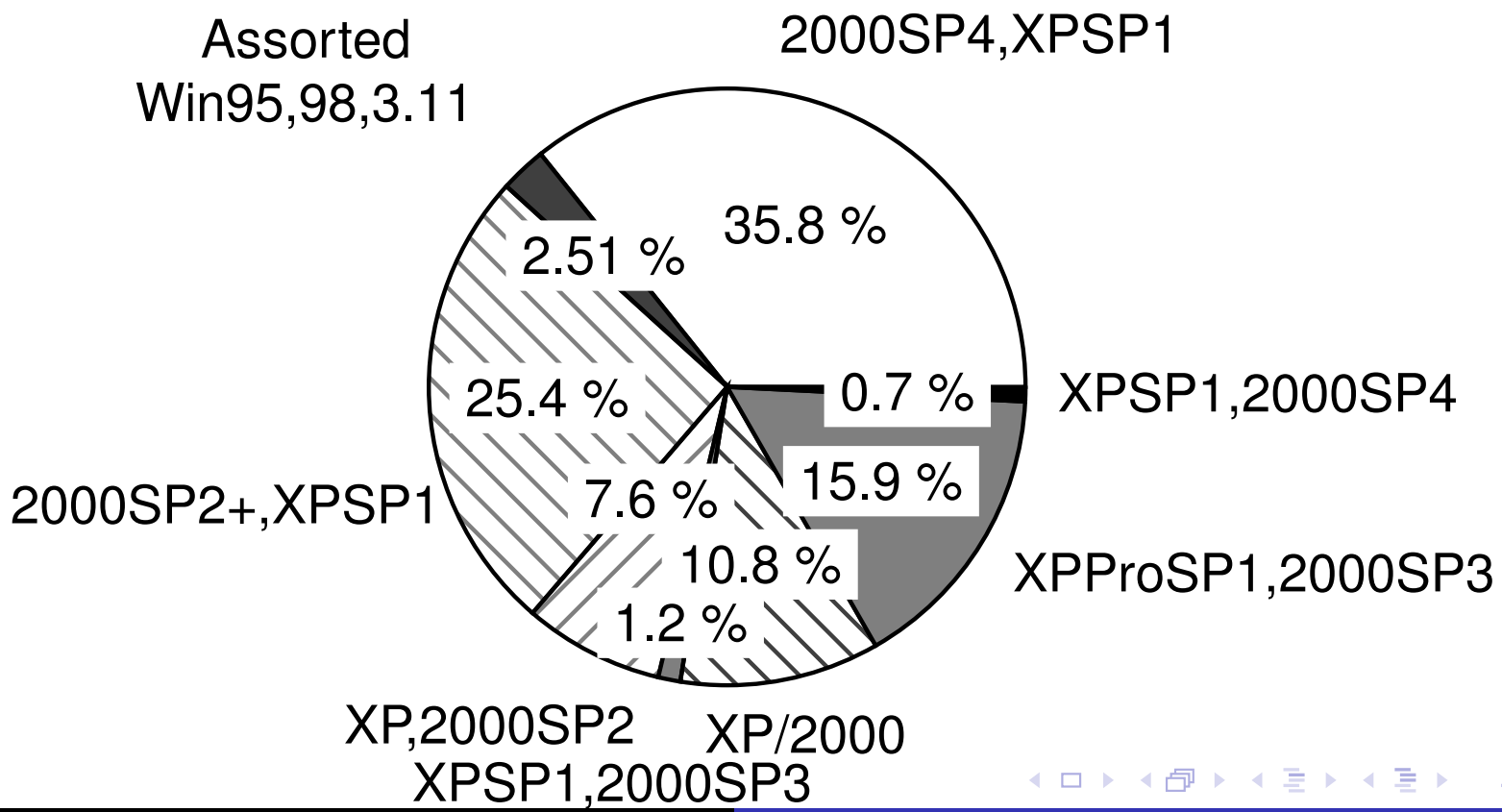


Victim Epidemiology: All

53K botnet



Victim Epidemiology: Windows-Only



Population Estimates

- How complete?
- Analysis of closed systems. Lincoln-Peterson
- two independent samples, M, and C, for the mark and capture sets.
- Second is merely random set in $\binom{N}{C}$.
  - Define: M – individuals marked by the first sample,
  - C – individuals observed in the second,
  - R – number in both.

  With R conditioned on M and C, the distribution of R is hypergeometric:

$$f(R|M,C) = \frac{\binom{M}{R}\binom{N-M}{C-R}}{\binom{N}{C}}$$

Population Estimates

If the mark and capture population samples are suitably large percentages of the total population, i.e., $M + C \geq N$, the estimate $\hat{N}$ is unbiased even for small sample sizes.

$$\hat{N} = \frac{(M+1)(C+1)}{R+1} + 1 \tag{4}$$

may not always yield sufficiently large mark and capture samples to estimate $\hat{N}$.
With a normal distribution for $\hat{N}$, we can further calculate a 95% confidence interval for this population as $\hat{N} \pm 1.96\sqrt{v}$, where:

$$v = \frac{(M+1)(C+1)(M-R)(C-R)}{(R+1)^2(R+2)}$$

Policy Implications for Sinkhole Collection

Policy First; Data Second

Large data collection efforts always have policy implications. Upfront, we consider:

- Privacy issues (granularity of clock skew)
- Use of Census data

Census of Victim OS/Patch-level

- Priority rank research into services
- Policy implication of discontinued/pay patch systems
- Concrete analysis of “Monoculture” concerns

Population Estimates

- How to improve?
- Dymanic models needed (non-closed population)
- Pen tester trend: Interaction with victim services (139, 445) to probe patch level.
- Borrow Broido's TTL work
- Add `p0f` dbs for NATing routers
- Add behavioral parameter:
  - estimate of cache-flushing behavior (cf., Wessels & Fomenkov's "Wow" paper)
  - purpose/use of botnet (e.g., spam, DDoS, click fraud)

Summary

- So far:
  - The Network is the Infection
  - Goal: detect botnets, not just bots
  - Existing botnet detection serendipitous, fragile
  - Taxonomy can direct towards solution
  - DDNS-based detection feasible
- Not discussed:
  - Expand DNS monitoring (future talk: algos and hardware)
  - Expanded RE
  - Traceback, LEO involvement
  - Threat metrics (cumulative bw estimation, key cracking potential, evasion potential)
  - Graph theoretic detection (P2P, TOR-based botnets)

Need Data/Malware?

- I have source for hundreds of bots, terabytes of pcaps
- If you're a researcher, and need samples or data:
  - Let's exchange PGP keys
  - and check with our advisors, net admins, etc.