STEVENS
Institute of Technology

Resilient Botnets

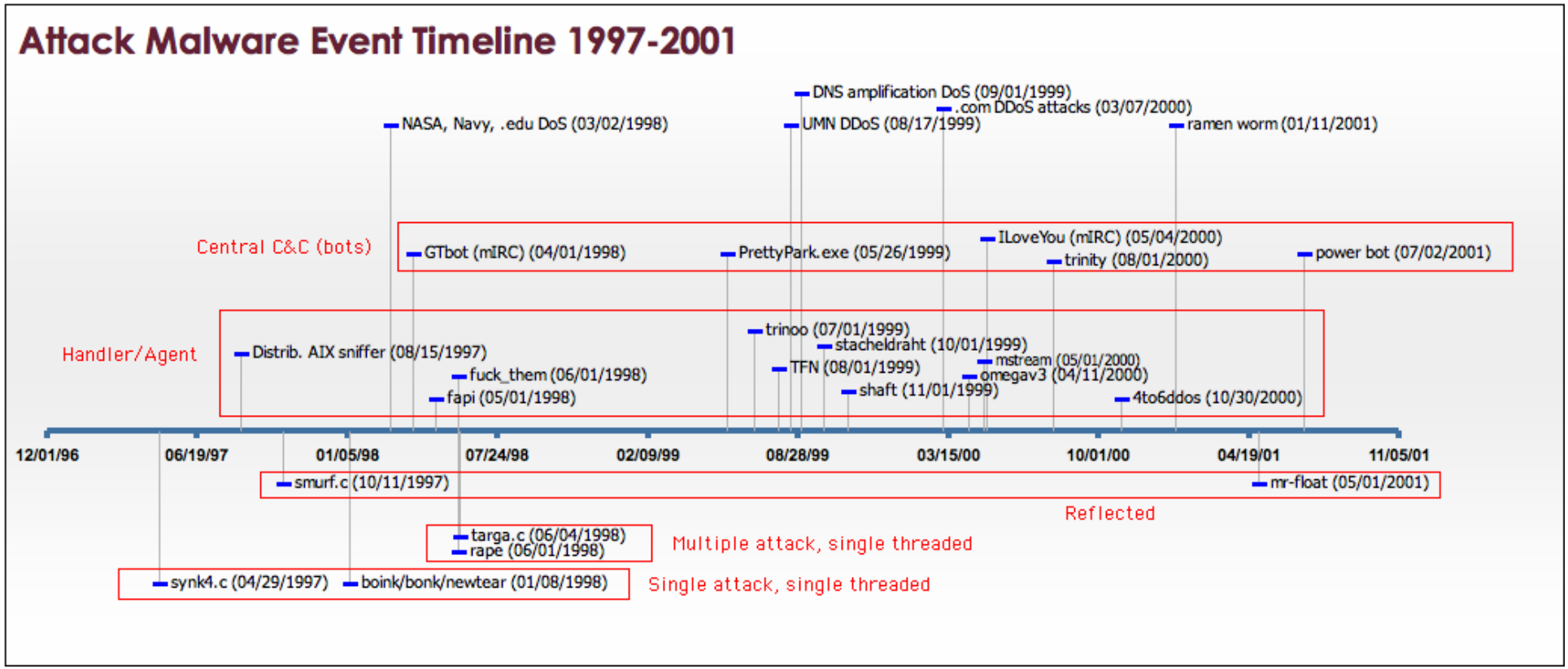
Dr. Sven Dietrich
Computer Science Department
Stevens Institute of Technology
spock@cs.stevens.edu

22 September 2009 version

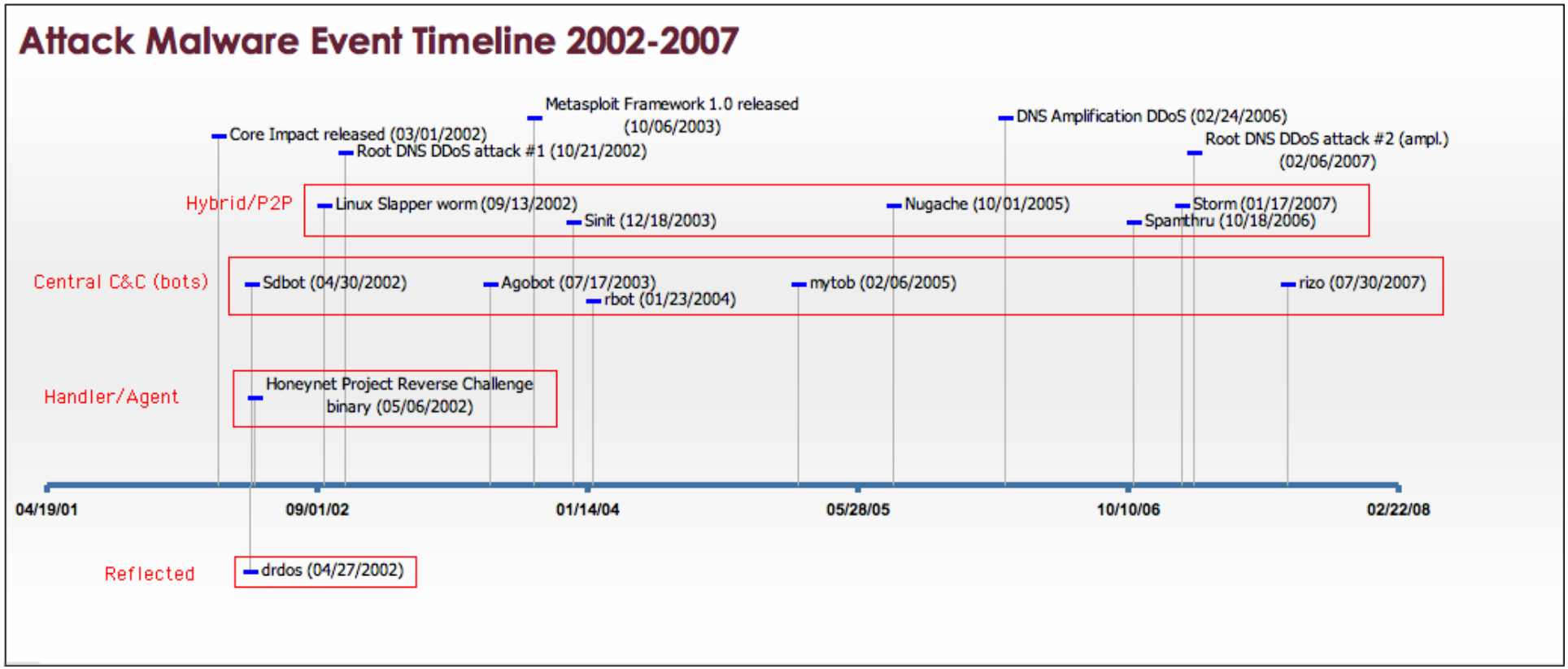
Malware generations

- Single-threaded, single-attack
- Single-threaded, multiple-attack
- Single-threaded, reflected/amplified
- Distributed, handler-agent
- Distributed, central command and control
- Distributed, hybrid
  - Specific structures, e.g. Peer-to-Peer (P2P)

Timeline (1)


Attack Malware Event Timeline 1997-2001
DNS amplification DoS (09/01/1999)
.com DDoS attacks (03/07/2000)
NASA, Navy, .edu DoS (03/02/1998)
UMN DDoS (08/17/1999)
ramen worm (01/11/2001)
Central C&C (bots)
GTbot (mIRC) (04/01/1998)
PrettyPark.exe (05/26/1999)
ILoveYou (mIRC) (05/04/2000)
trinity (08/01/2000)
power bot (07/02/2001)
Handler/Agent
trinoo (07/01/1999)
Distrib. AIX sniffer (08/15/1997)
stacheldraht (10/01/1999)
mstream (05/01/2000)
fuck_them (06/01/1998)
TFN (08/01/1999)
omegav3 (04/11/2000)
fapi (05/01/1998)
shaft (11/01/1999)
4to6ddos (10/30/2000)
12/01/96
06/19/97
01/05/98
07/24/98
02/09/99
08/28/99
03/15/00
10/01/00
04/19/01
11/05/01
smurf.c (10/11/1997)
mr-float (05/01/2001)
Reflected
targa.c (06/04/1998)
rape (06/01/1998)
Multiple attack, single threaded
synk4.c (04/29/1997)
boink/bonk/newtear (01/08/1998)
Single attack, single threaded

Timeline (2)


Attack Malware Event Timeline 2002-2007
Metasploit Framework 1.0 released (10/06/2003)
Core Impact released (03/01/2002)
Root DNS DDoS attack #1 (10/21/2002)
DNS Amplification DDoS (02/24/2006)
Root DNS DDoS attack #2 (ampl.) (02/06/2007)
Hybrid/P2P
Linux Slapper worm (09/13/2002)
Sinit (12/18/2003)
Nugache (10/01/2005)
Storm (01/17/2007)
Spamthru (10/18/2006)
Central C&C (bots)
Sdbot (04/30/2002)
Agobot (07/17/2003)
rbot (01/23/2004)
mytob (02/06/2005)
rizo (07/30/2007)
Handler/Agent
Honeynet Project Reverse Challenge binary (05/06/2002)
04/19/01
09/01/02
01/14/04
05/28/05
10/10/06
02/22/08
Reflected
drdos (04/27/2002)

Peer-to-Peer

- Linux Slapper worm (2002)
- Agobot, Sinit (2003)
- Phatbot, Zindos (2004)
- Nugache (2005)
- SpamThru (2006)
- Storm (2007)
- …

More about P2P

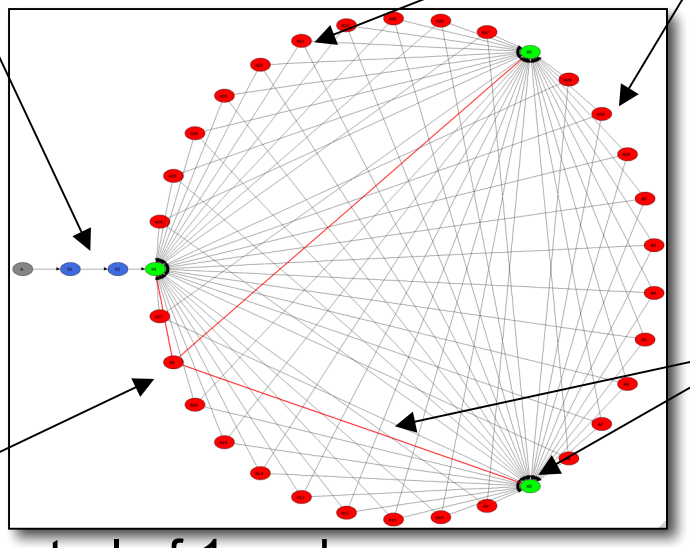
- P2P has been a goal for years (>5?)
- New topologies leading to survivability of botnets (Storm, Nugache)
- New command and control mechanisms

Comparing Structure/Path

30 Attack “agents”

Stepping stones

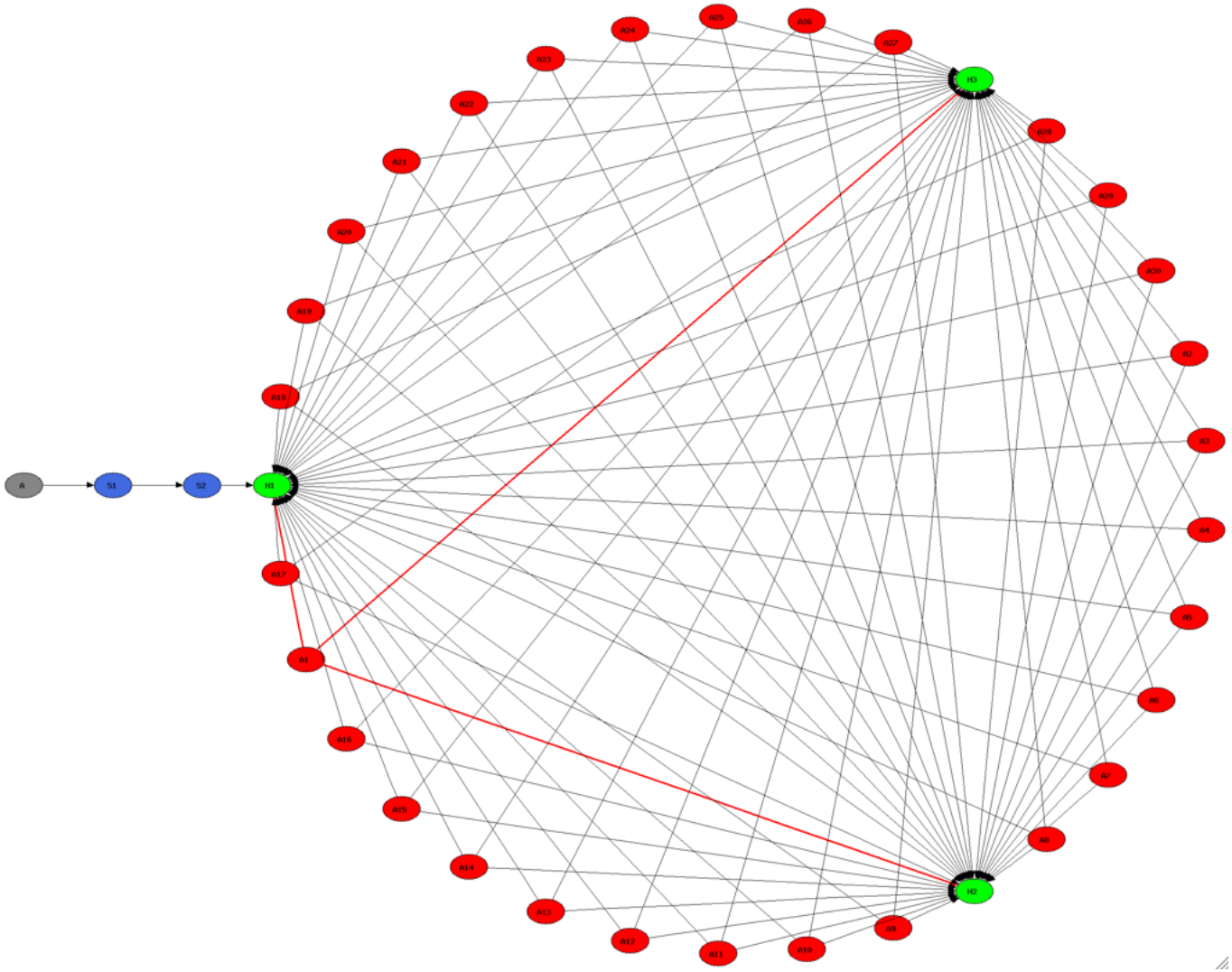
Attacker ↔ **S1** ↔ **S2**



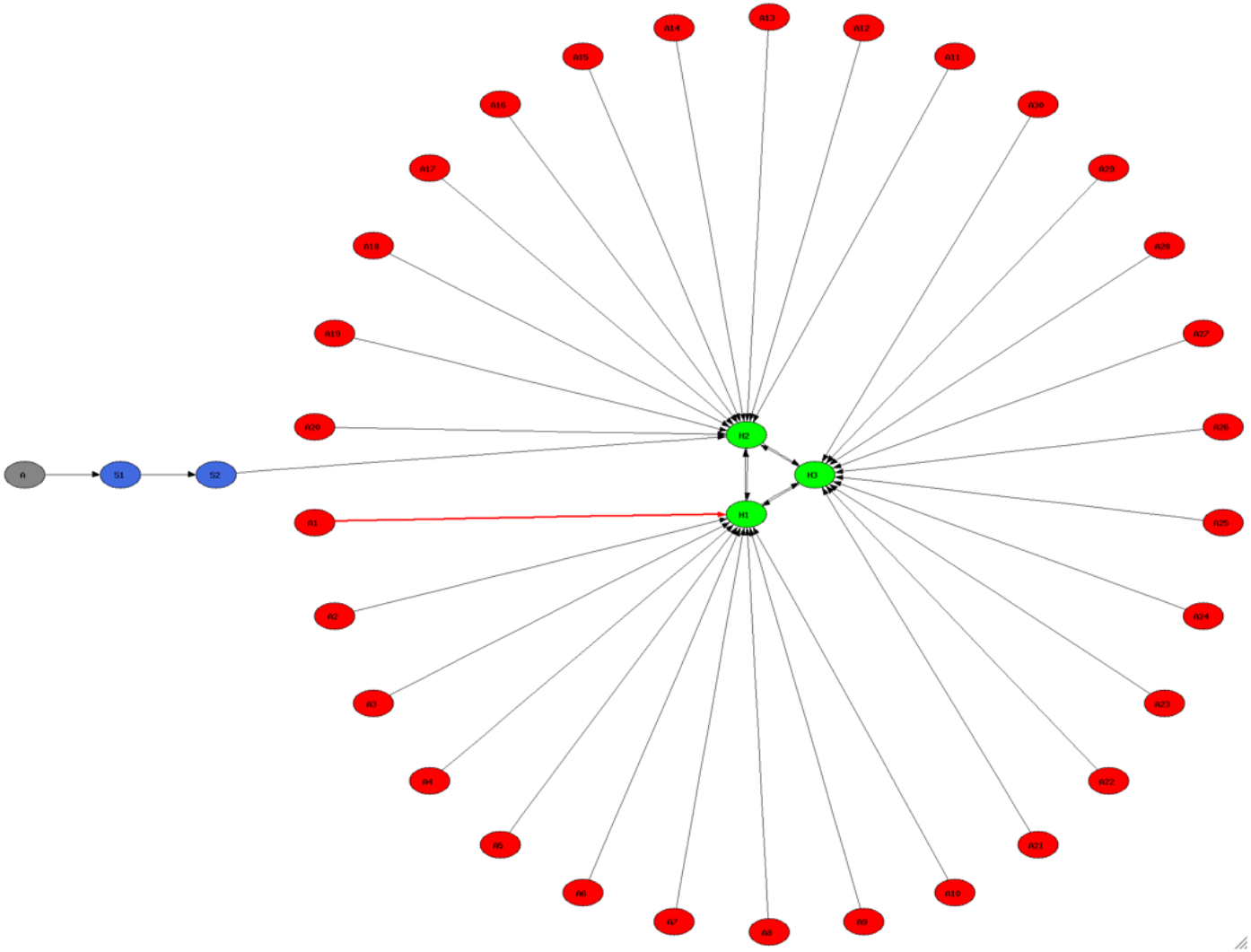
Command & Control (C&C)

- Assume we have control of 1 node to start investigation

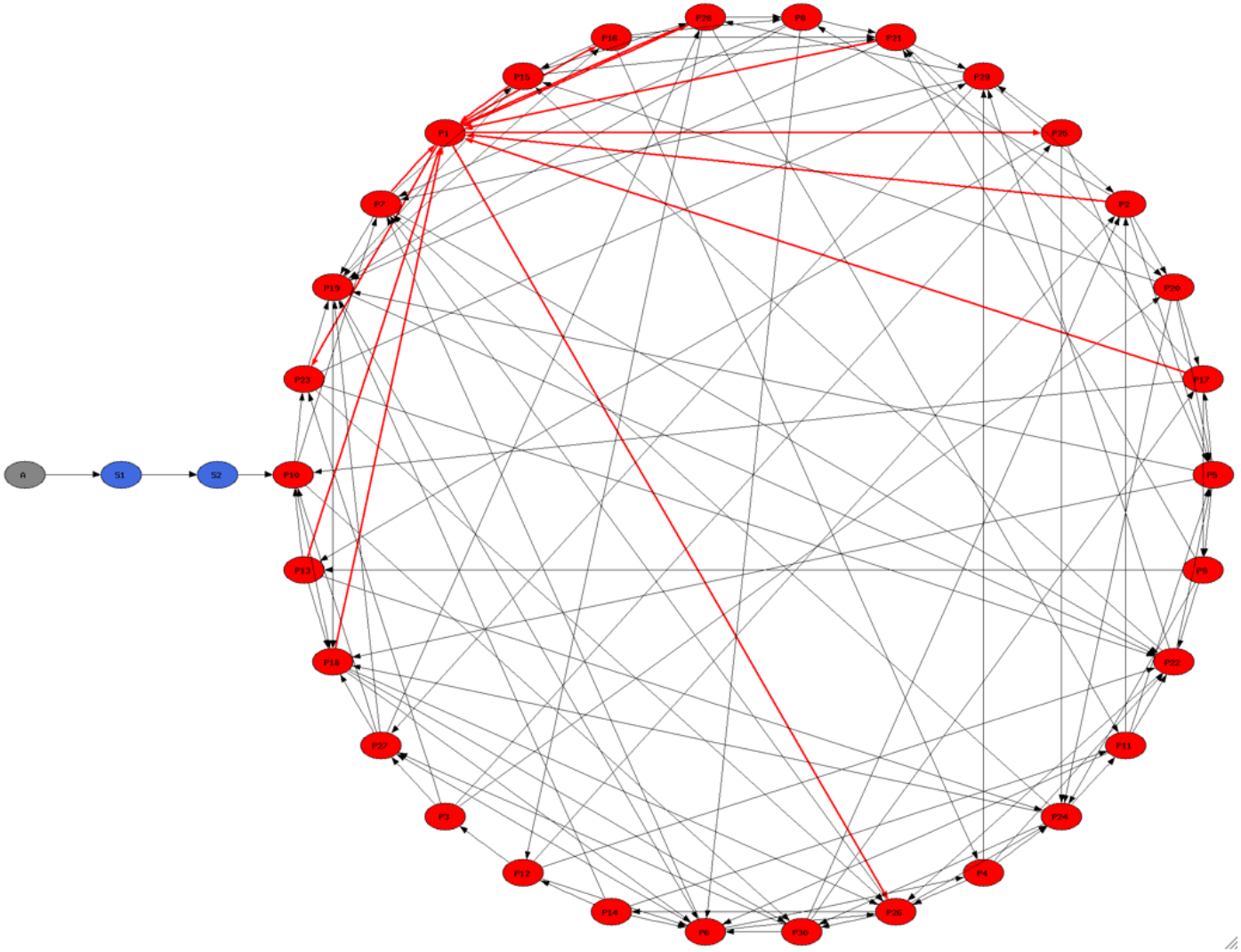
Structure - Handler/Agent



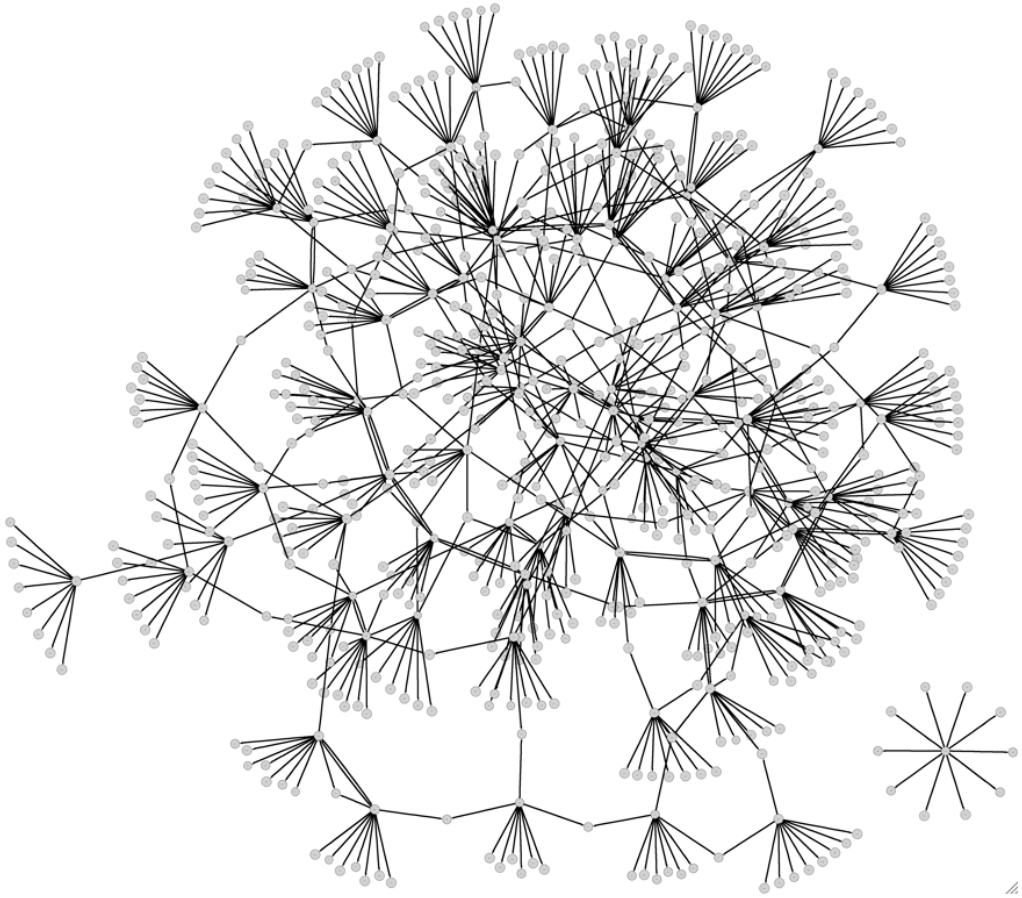
Structure - IRC botnet



Structure - Peer-to-Peer



Sample P2P botnet structure



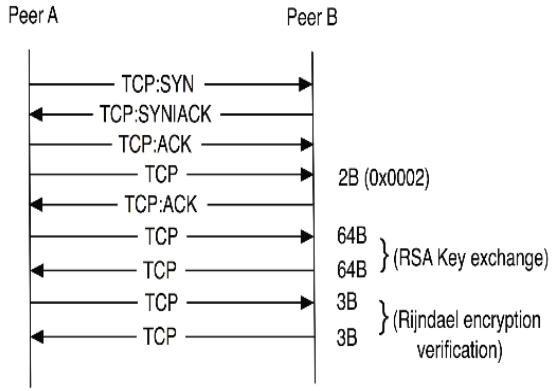
Nugache - a P2P bot

- Nugache escaped scrutiny for a long time
- Designed to lay low
- Crypto was “good enough” to thwart casual observers from peering into packets
- Correlation non trivial

Nugache Bot - Features

- Successful P2P C&C
  - Original on port 8/tcp (June 2006)
  - Since then on random high-numbered ports
  - *All C&C* over P2P channel, including updates
  - Advanced use of crypto
- Feature-rich, OO shell command set
- Exposes only a *limited subset* of peers to traffic analysis
- Observed uses: DDoS, scanning, exfiltration of sensitive data, and more

Nugache public-key crypto



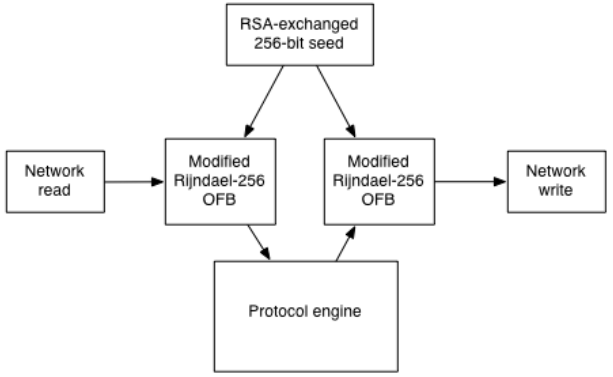
- RSA-style key exchange
  - 512-1024 bit public key modulus
  - Fixed exponent e= 2^{16}+1
  - Typically (n,e) exchanged, here only n = pq is transferred

00 02	nonce (21 bytes)	check (4 bytes)	session key (32 bytes)	TestN (4 bytes)

  - Public key embedded in binary

Nugache symmetric crypto

- Rijndael-256 cipher in OFB mode (AES-like)
  - Used Rijndael-only block size (256 bit)
  - One session key, applied to each I/O stream



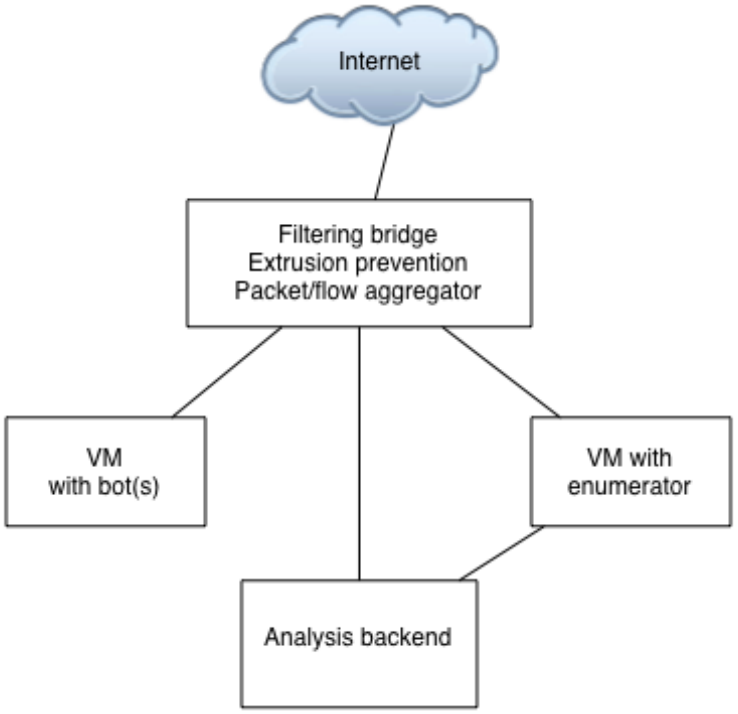
Feature Comparison

	Storm	Nugache
Primary C2	“Pull” from C2 servers	P2P
Initial Peer List Seeding	Text file	Built-in, or pre-loaded into Registry (HKCU)
Use of Crypto for C2 Comms	MD4 hash to conceal file names, 320bit shared key	512-1024bit RSA key exchange, Rijndael-256-OFB session keys
Use of DNS in C2	None/ ”Fast Flux” to hide C2 servers	None
Connectivity	Hundreds	~1 dozen
Updates	By command (not at all?)	Automatic
Listening port	Random high-numbered	Random high-numbered
Architecture	Multipartite	Monolithic
Detection	Visible on host; eDonkey traffic detectable by signature	Visible on host; P2P traffic not easily detected by signature

Analysis of the Storm and Nugache Trojans: P2P is here, Sam Stover, Dave Dittrich, John Hernandez, and Sven Dietrich, USENIX ;login: vol. 32, no. 6, December 2007, pp. 18-27

Data collection & analysis

- How data was collected
  - Honeywall
    - Relevant packets only
  - tcpdump
    - Full packet capture
  - Custom tools
- How data was processed
  - Wireshark
    - Payload
  - Flow analysis tools
    - Flows/headers
  - Custom tools



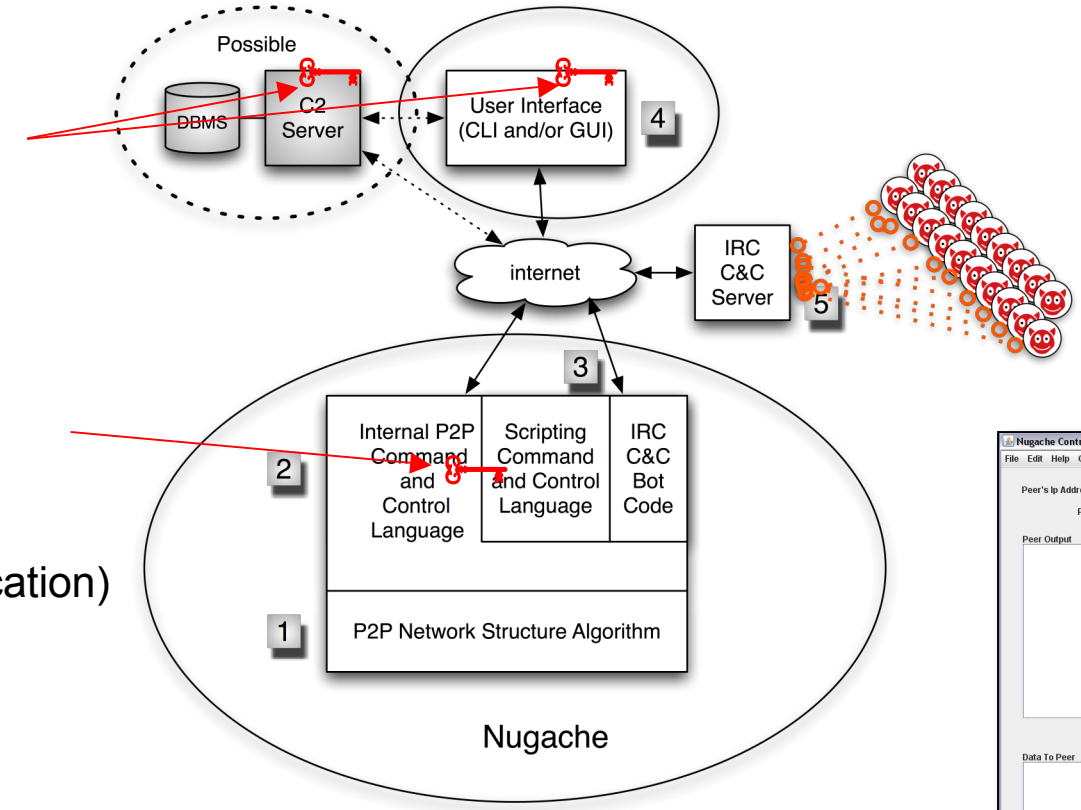
Command propagation

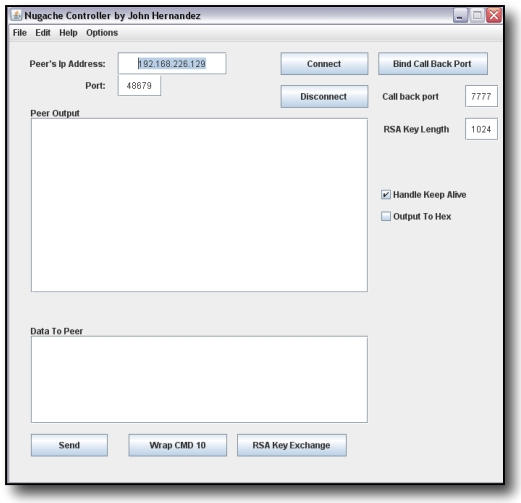
- How do commands travel?
  - Simple commands (hi/bye/peer list)
    - Numeric, hardly any text, easy to forge
  - Extended commands
    - Text-based, signed, hard to forge
    - Capable of probabilistic, wildcard, UUID-based delegation
  - Updates
    - Binary transfers over P2P channel
    - Signed as well

Command and Control Relationships

4096 bit private (signing) key

4096 bit public (signature verification) key





Encryption/Signing of Commands

- Command block

```
10|3|22|AIM.Spread(10);|583E37091E7B8B7AFD8662485EFE7BF08DA3ACB63F611D46A329DD1CF86DB42A7FCD6616F2CAAEA728701F
A8B71C782FAA48911DBF069E0091D914415320DD8624CFA982D23761C082BC88C037621473B3E518E6DD8C2E659E1A02EF2DA63C86C788
53EF7868A3B92E4F7F8FB7DFB8BD9AF9F7ACB1A3AE7761A4E008B070B02793965FB4F1DBFA447FD871058F4893511BECAB27AB81B532B6
143159EAB065798220A46864B0927E0A8D6F8C7C6D0FA396D23C33010F61054381531D5D166A4F08517064A2EECBA2A7671E35257247CB
2D7F07DFBC6ADC3C7A3D2EB9DAEB29DAE79B9157657A70620DFF4CA7E827598EEF9D3CDC2B191DF6A2C0B1D33F95228C368BD2BE2A7E59
D3223B898EA975C991057BEE002B9DA34A97F697F12832B7CBFEB786B438092FEEA362A10EC9E19602DD9D3F557014DC06DDD34EE5FD14
C2D7DB16D9BE05E31DA862341F92464761443DE1E4865A867E5179998C386115890DBDDE3FF61B3E34B184C20B7B3155DF4D6C6C973B36
FBAEC7A6A5D82A0C7C89ACB41BFB60497A3D3578701DEC096580098B756F9633D019109430E3804E127B4C924AAB0197171AB554642ED2
E7B04019C70C5DF26BC4D2CA18C30AC990EBAFAA302C91F998C5556D513448FA357AA036A08B9F43A9A2A5EE1C2301F68A70BA8431ED80
8190D1AABCDE180F58E4DB13C17FECF47B8131304F96EA337E64FC5A|D027C867
```

- Signature block decrypted with embedded 4096-bit public key

```
01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00
B55A34436A2A6AD7DA837A21BA58E5CC
```

- Validation

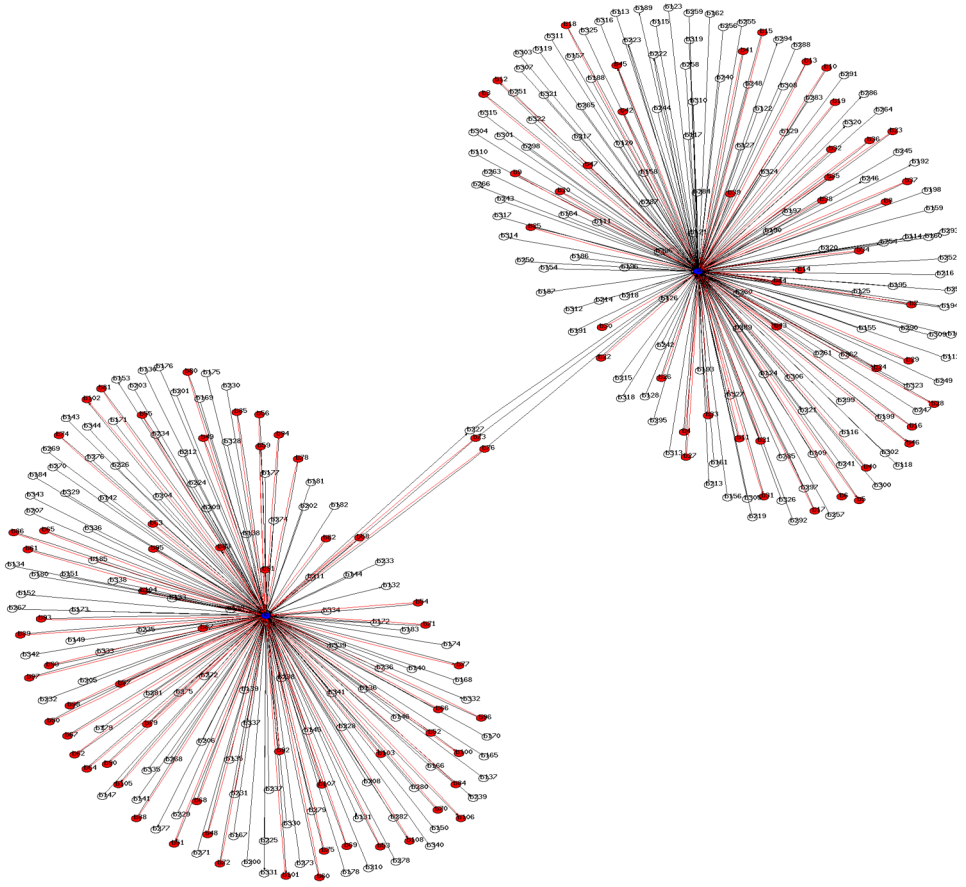
```
$ echo -n "AIM.Spread(10);D027C8673" | md5sum
b55a34436a2a6ad7da837a21ba58e5cc  -
```

Decentralized

- Truly P2P
  - Past June 2006, no hybrid behavior
- No high valence nodes by design
  - No more than 10 connections in set
  - No more than 100 known peers
- No need for a “super botnet”

Connection graph for 7 days for 2 bots

- Peers and friends
  - Are some peers more equal than others?
  - Different versions



Detection

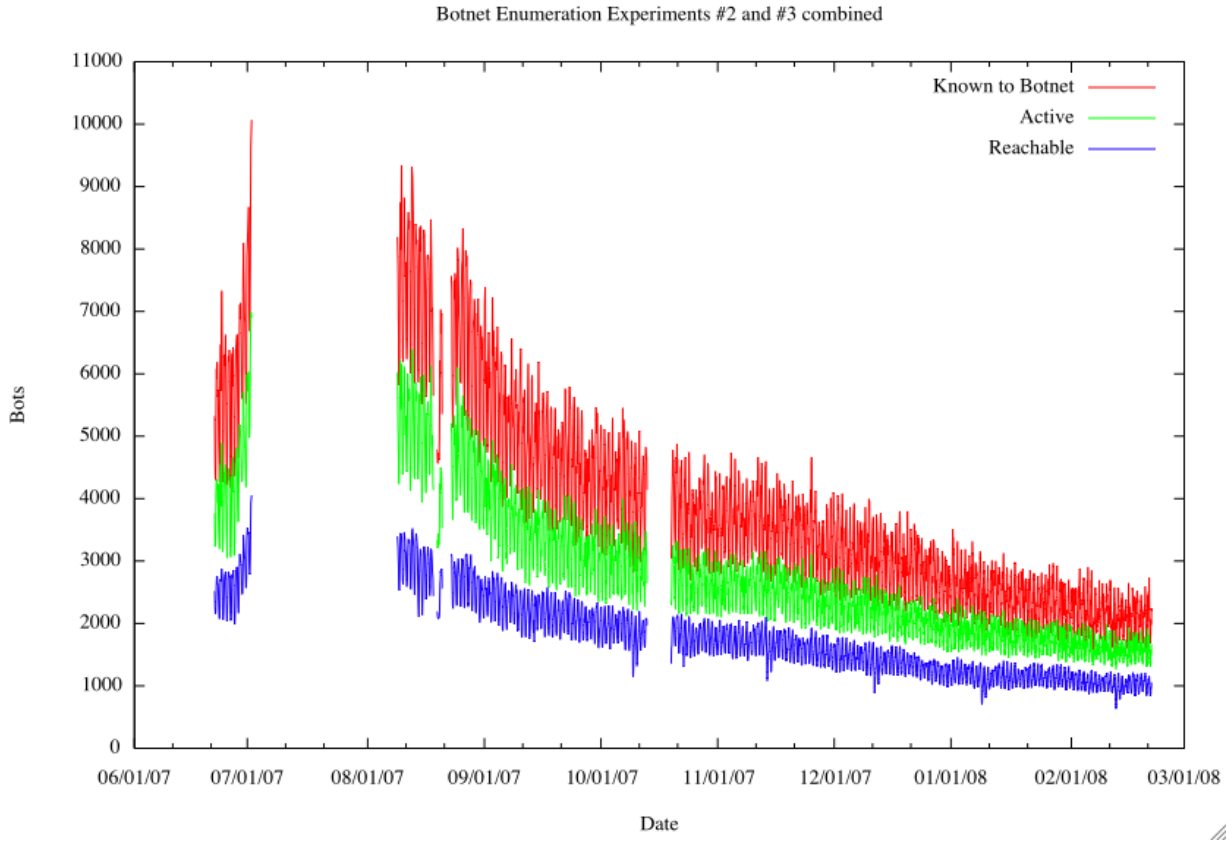
- Network
  - Random high-numbered ports
  - Minimal strings in key exchange
  - Visible by its secondary activity/ies
    - Scanning
    - Attacks
    - Keylog activity
    - Other
- Host
  - Running as “benign” process
  - Tainted the registry keys

Remaining Questions

- Resilient aspects to be inherited by others?
- How can we apply this to *Bot XYZ* arriving next week?
- We only had limited network views - how would it have looked from multiple vantage points?
- What if we hadn't been stealthy?
- Not fully reverse-engineered client
  - What does botmaster console look like?
  - We have our own "console"

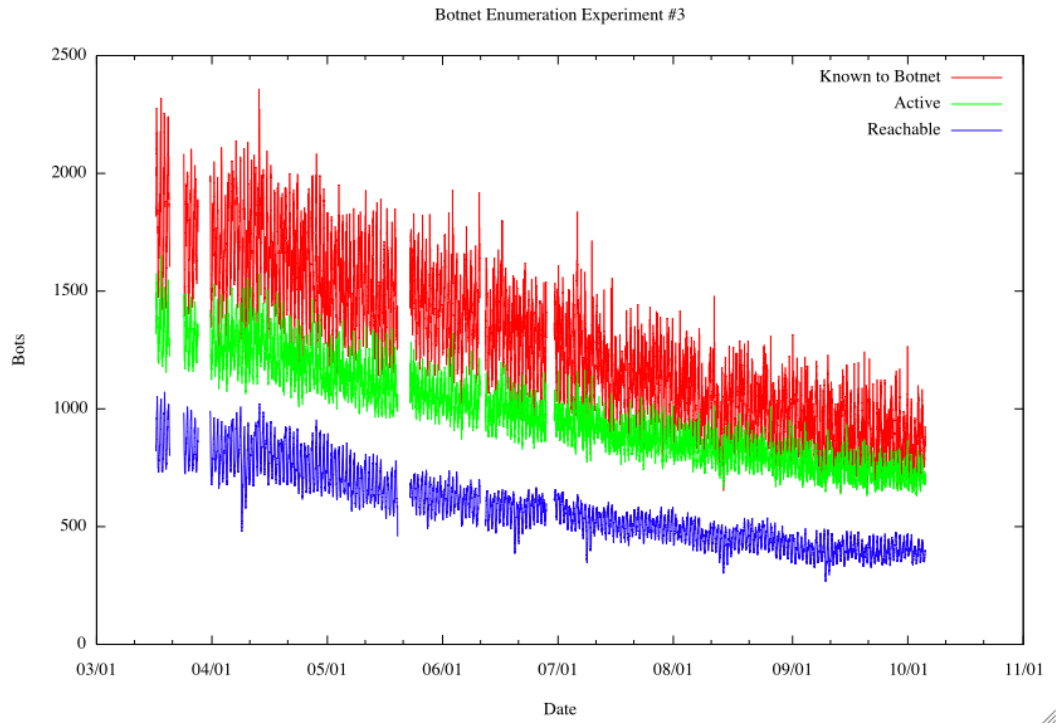
Enumeration

- Shown here: June 2007-March 2008



Enumeration

- Shown here: March-October 2008



Patch Tuesday Effect

Microsoft serves light fare on Patch Tuesday

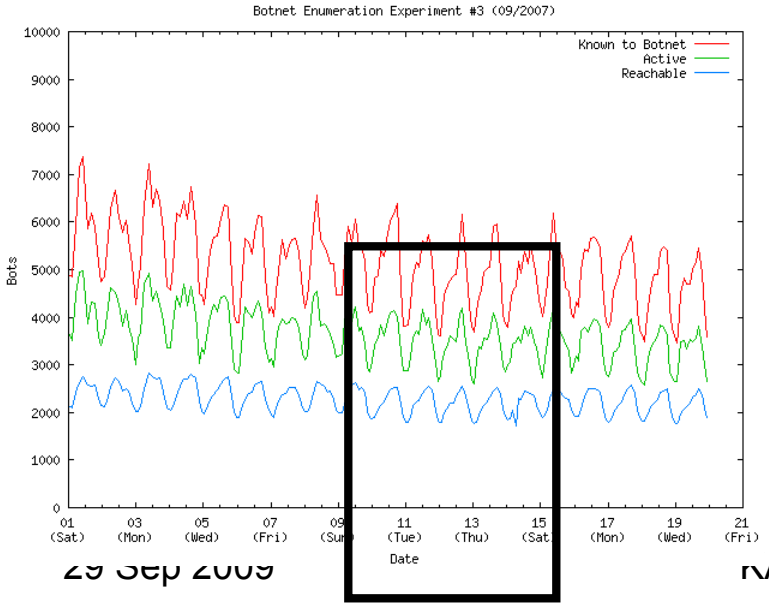
No critical patches for most Windows users

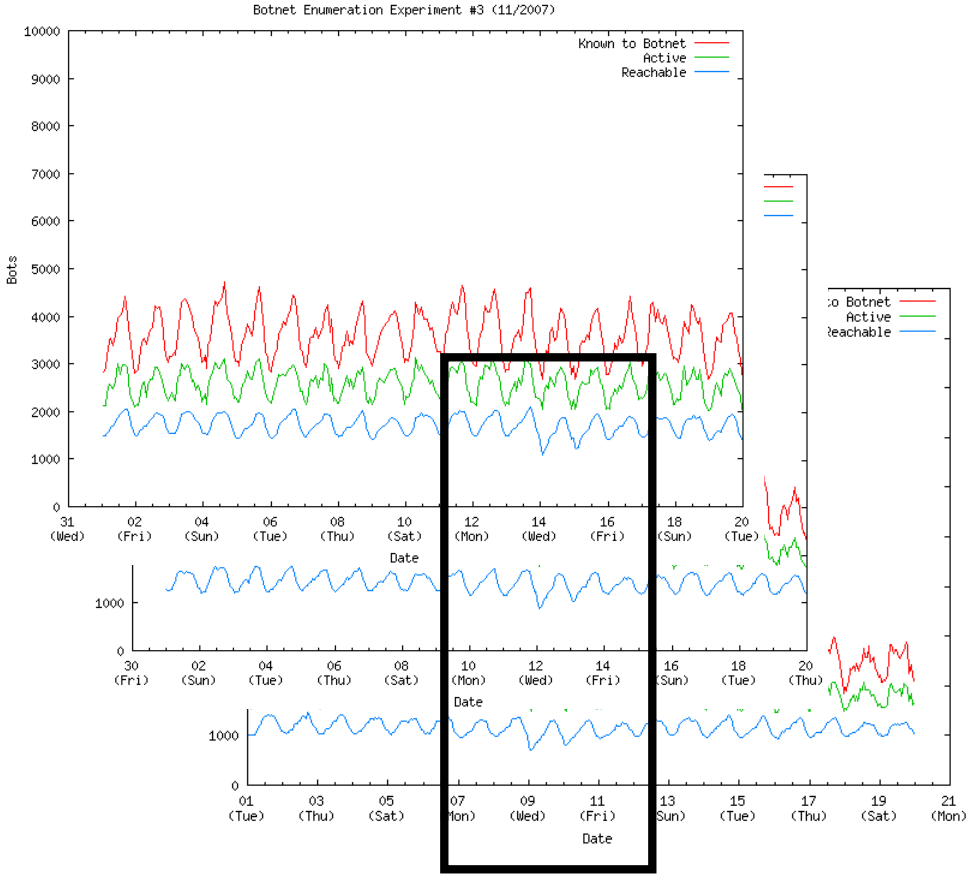
By Dan Goodin in San Francisco → More by this author
Published Tuesday 11th September 2007 22:00 GMT
Green Computing - Where do you stand?

Microsoft served comparatively modest fare for its monthly patch release on Tuesday, issuing only four security-related updates, only one of which carried its top severity rating of critical. It plugged a hole in a Windows 2000 component, while the other updates fixed vulnerabilities rated as important in instant messenger programs, Visual Studio .Net and Windows services for Unix found on several different versions of the Windows operating system.

In a rare event, the typical Windows user is likely to have just one patch to install. It addresses a vulnerability in the MSN Instant Messenger and Windows Live Messenger that could allow an attacker to take over a machine by tricking a victim into clicking on a specially crafted chat request. Despite MSN Messenger being installed on every copy of Windows, Microsoft rated the flaw important, presumably because it can't be exploited without the user taking action first.

Some users may have no patches to install, as was the case with this reporter. That's because the vulnerability doesn't affect Windows Live Messenger version 8.1, which was installed on the machine. A spokeswoman says other versions of Windows Live Messenger don't use Windows Update to install new updates. Instead, the client prompts the user to install a new version, she said. Windows Update still encouraged us to run Windows Malicious Software Removal Tool, as it does every month.





Encryption/Signing of Executable

- `HKCU/Software/GNU/upsn` key decrypted w/embedded 4096-bit public key:

```
01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFF0018cac164032d1a74bb678295801adcc5
```

- Validation
  - `HKCU/Software/GNU/uphs` key
    `"uphs"=hex:18,ca,c1,64,03,2d,1a,74,bb,67,82,95,80,1a,dc,c5`
  - MD5 hash of version 14 binary
    `18cac164032d1a74bb678295801adcc5`

Questions?

- Contact info:

Dr. Sven Dietrich
Computer Science Department
Stevens Institute of Technology
Hoboken, NJ 07030, USA
spock@cs.stevens.edu