Cracking the Encrypted C&C Protocol of the ZeroAccess Botnet

@ VirusBulletin 2012, Dallas

Presented by:
John Morris

Agenda

Cracking the Encrypted C&C protocol of the ZeroAccess Botnet.

1. ZeroAccess 1 vs 2
2. The Infection process
3. Decoding the C&C
4. Structure and Size of BotNet
5. The Impact
6. Q&A

For a copy of the paper visit http://www.kindsight.net/securitylabs

About Kindsight

Majority-owned subsidiary of Alcatel-Lucent

- Founded in 2007
- Offices in Mountain View, CA and Ottawa, ON

Security analytics platform that Service Providers embed in their network:

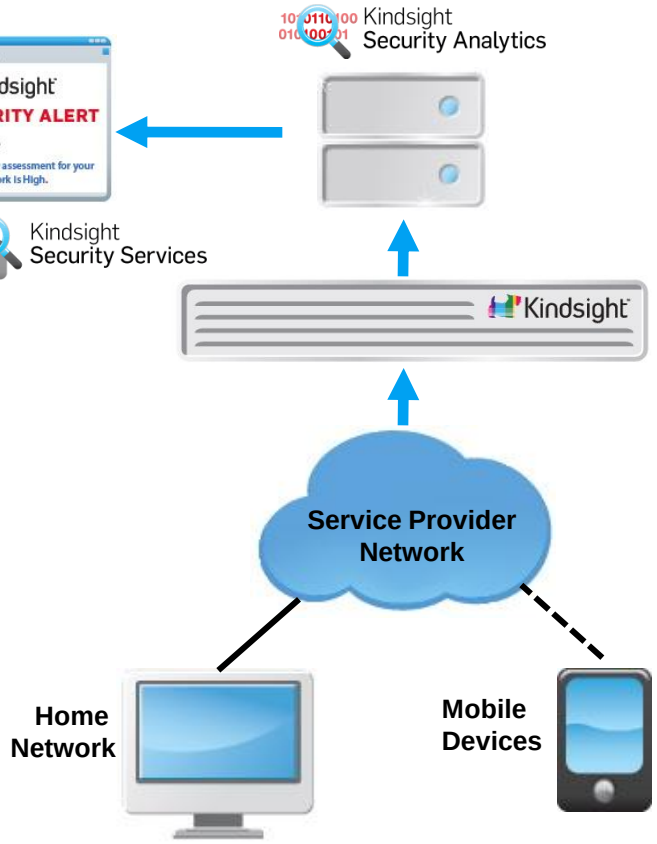
- Security Analytics
- Security Services
- Mobile Security

Key components:

- Snort based malware detection sensors placed at strategic points in the service provider network
- Security Analytics Engine in ISP data center
- Remediation portal for subscribers

Kindsight Security Labs

- Research team with expertise in network-based malware and signatures



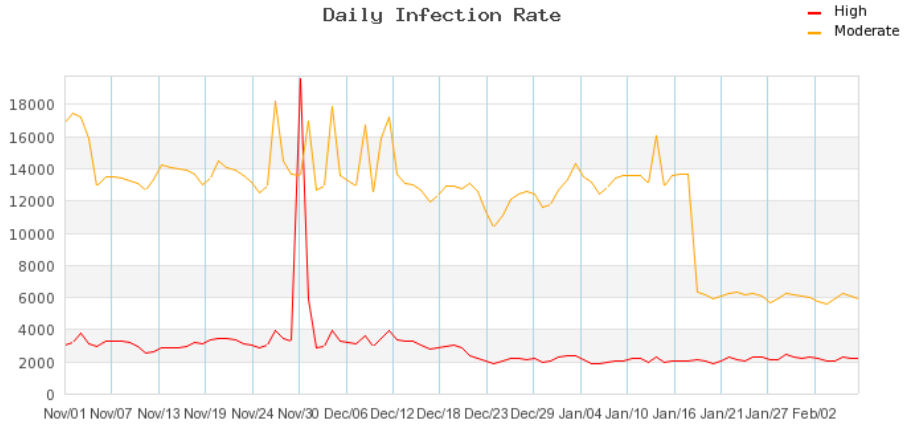
Detected Strange Traffic Pattern

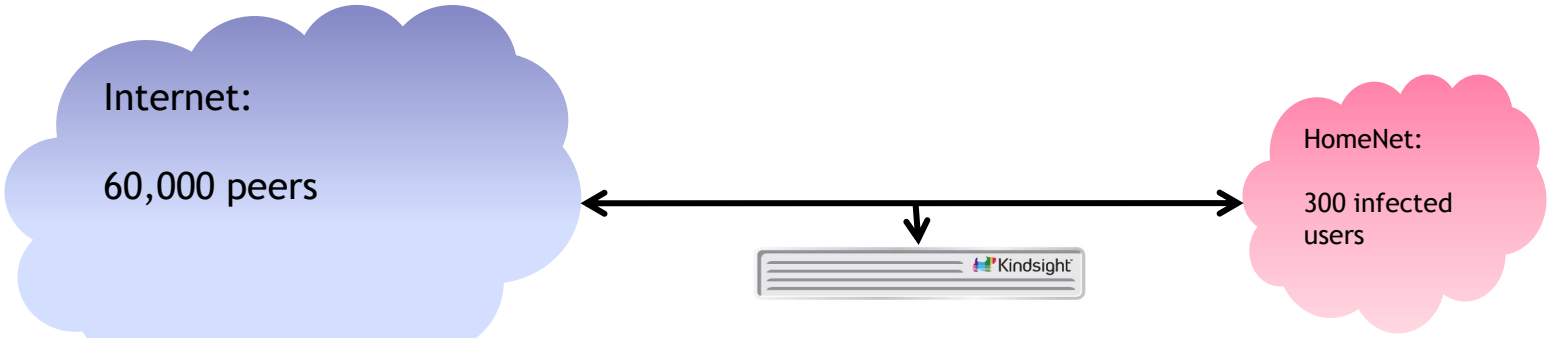
New signature deployed in sensors.

New infection event seen.

300 infected computers were talking to ov

Looks like a p2p botnet?






```
alert tcp $HOME_NET any -> $EXTERNAL_NET any
(msg:"Win32.Botnet.ZeroAccess - Runtime Detection";
flow:established,to_server;  dsize:20;  content:"|E5 AA C0 31|";  depth:4;
content:"|5B 74 08 4D 9B 39 C1|";  distance:5;  within:7;
reference:url,www.abuse.ch/?p=3499;
reference:url,www.symantec.com/security_response/writeup.jsp?docid=2
011-071314-0410-99&tabid=2; classtype:botnet;  sid:2013911;  rev:3; )
```

ZeroAccess At A Glance

	ZeroAcess 1	ZeroAcess 2
Purpose:	Both appear designed to distribute programs via a resilient P2P network and execute them. Most notable payloads perform click-fraud and BitCoin mining.	
Infection Path:	Blackhole, drive-by download sites, Trojans (+social engineering)	
Aliases:	Sirefef	
Released:	Sept 2011	April 2012
Protocols:	Custom P2P,TCP	Custom P2P, UDP & TCP
Ports:	21810, 21860, 22292, 25700, 34354	16464, 16465, 16470, 16471
Encryption:	RC4 w/static key	XOR w/static key & RC4 w/dynamic key
Protection:	Kernel-Mode rootkit	User-Mode / Hidden Files

ZeroAccess 2 - Infection Process

When executed (on Windows7), it sets itself up two directories using a device specific fake CLSID as a directory name:

```
C:\Windows\Installer\{CLSID}
C:\Users\UserName\AppData\Local\{CLSID}
```

It drops a copy of two files into these directories:

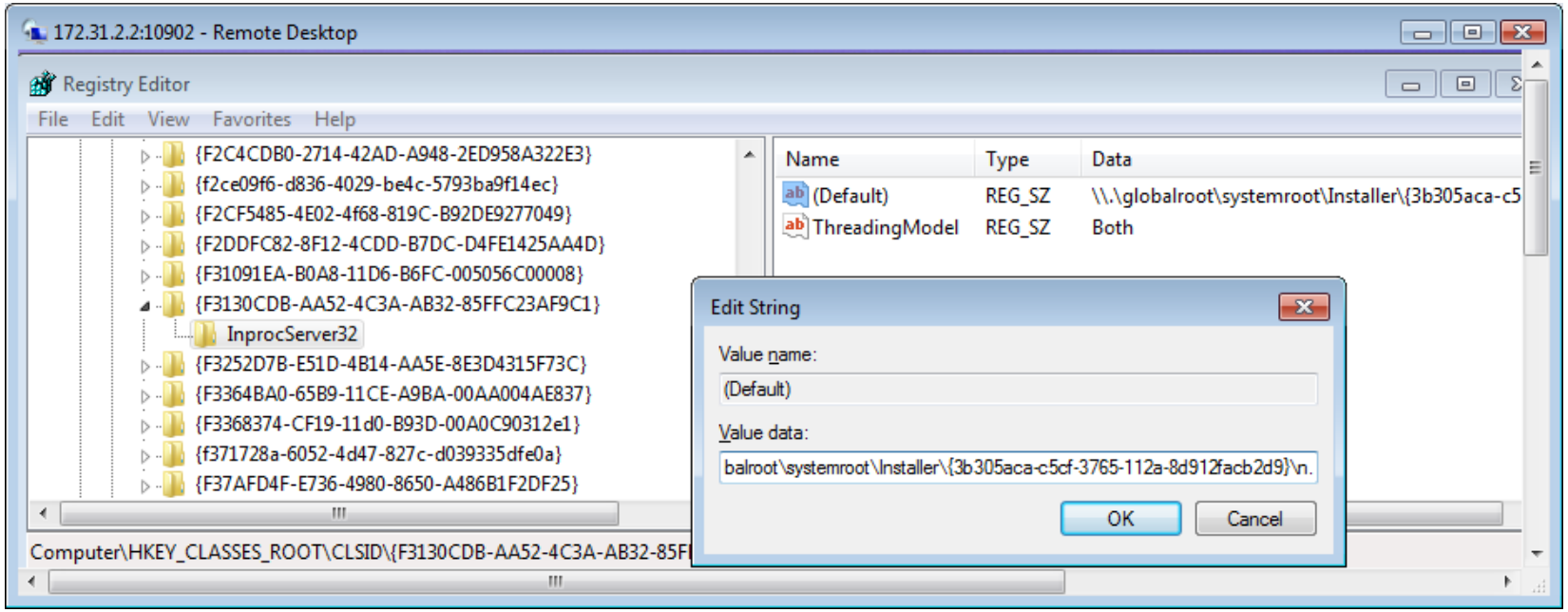
```
n - the malware executable
@ - the list of 256 peer IP addresses
```

Subdirectories U and L are also created. U will contain additional downloaded malware. L is a directory for temporary files.

No Kernel-Mode components required

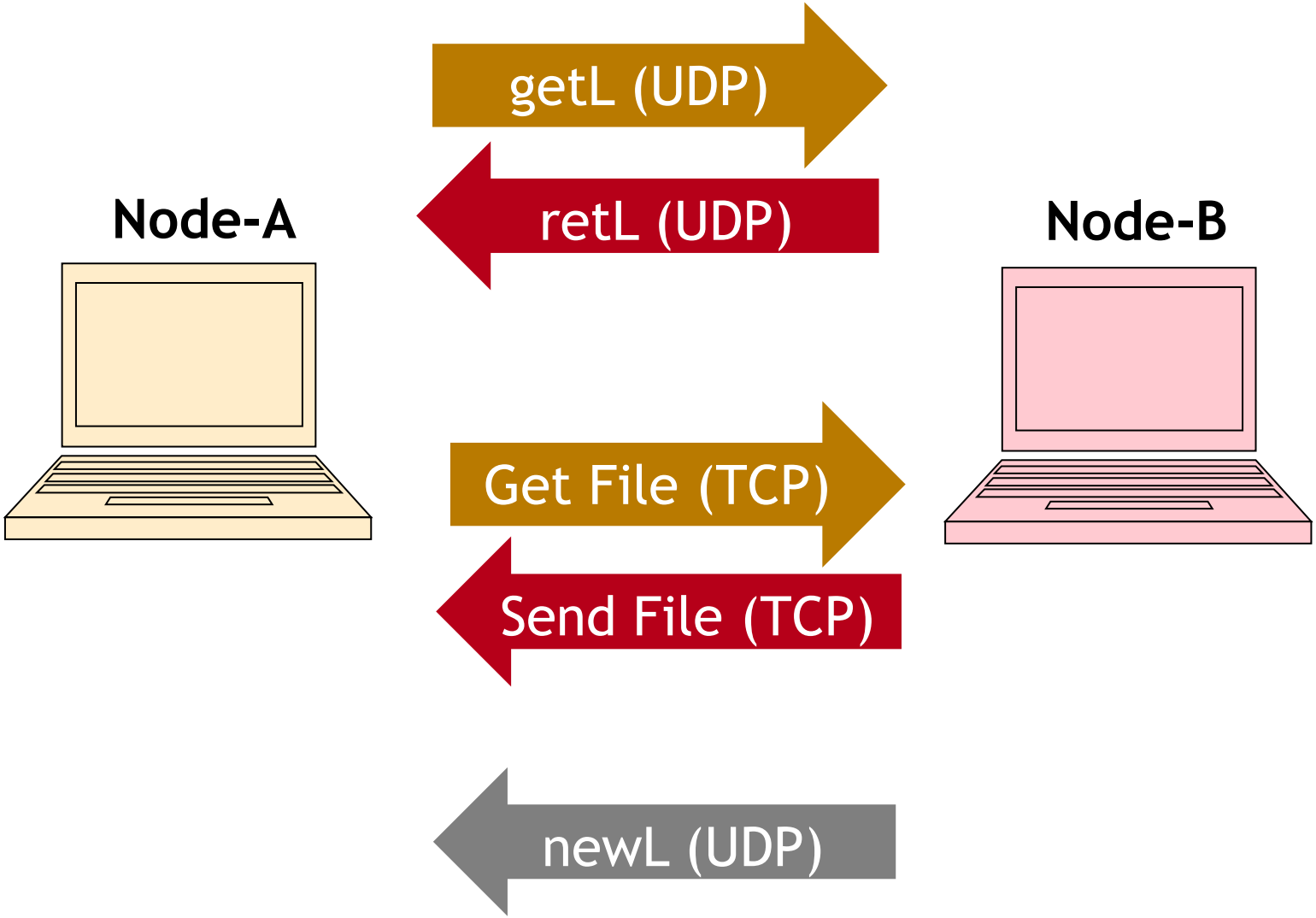
ZeroAccess 2 - Infection Process (cont'd)

The malware ensures it is reloaded on boot by modifying the CLSID entries in the registry for wbemess.dll and shell32.dll to load the malware instead of the normal DLLs.



The malware in the \Installer directory is associated with the wbemess.ddl registry entry. It attaches itself to an svchosts process and is active in the p2p communication.

ZeroAccess2 - P2P Command and Control



ZeroAccess2 - UDP Encryption

UDP command packets contain several 32bit (4 byte) fields, best thought of as unsigned integers (little-endian).

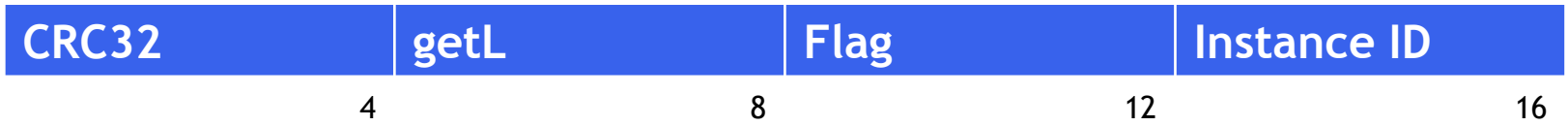
All UDP command packets are obfuscated using a rudimentary XOR scheme.

Sample encryption/decryption algorithm written in C:

```
void XORcrypt(unsigned char *buff, int bytes)  {
    int i;
    unsigned int *num;
    unsigned int key=0x66747032;            // Encryption key 'ftp2'

    for (i=0; i<bytes; i+=4) {              // loop through buffer
        num=(unsigned int *) &buff[i];      //  - 4 bytes at a time
        *num ^= key;                        // XOR to (de-)obfuscate
        key = key<<1 | key>>31;             // Rotate key left 1
    }
}
```

getL Request (UDP)



The Bot will send a getL command to one peer in its list every minute.

The getL command packet is always16 bytes in size, divided into four 4-byte/32 bit integer fields

The CRC32 field is a 32-bit CRC of the entire unencrypted16 byte request. The field has a value of zero for the purposes of the calculation.

The flag field has always been seen to have the value of zero.

The 32 bit value at the end of the packet appears to be a unique identifier of the Bot instance sending request.

retL command / getL response (UDP)

CRC32	retL	Flag
N1	← *Number of peers*	
IP address (1)	Time(1)	
…	…	
IP address (n1)	Time(n1)	
N2	← *Number of files*	
FileName(1)	TimeStamp(1)	FileSize(1)
128 byte signature		
…	…	…
FileName(n2)	TimeStamp(n2)	FileSize(n2)
128 byte signature		

Header

Peer List

Normally limited to 16 IP addresses

File List

Typically ~3 files

Get File command (TCP)

When processing a ‘retL’ response, if the bot sees a new or updated file it will send a Get File request to the peer.

TCP is used for file transfers

The file name is stored as a 32 bit binary value and used as an 8 character hex string.

File names observed included “800000cb”, “00000001” and “80000000”.

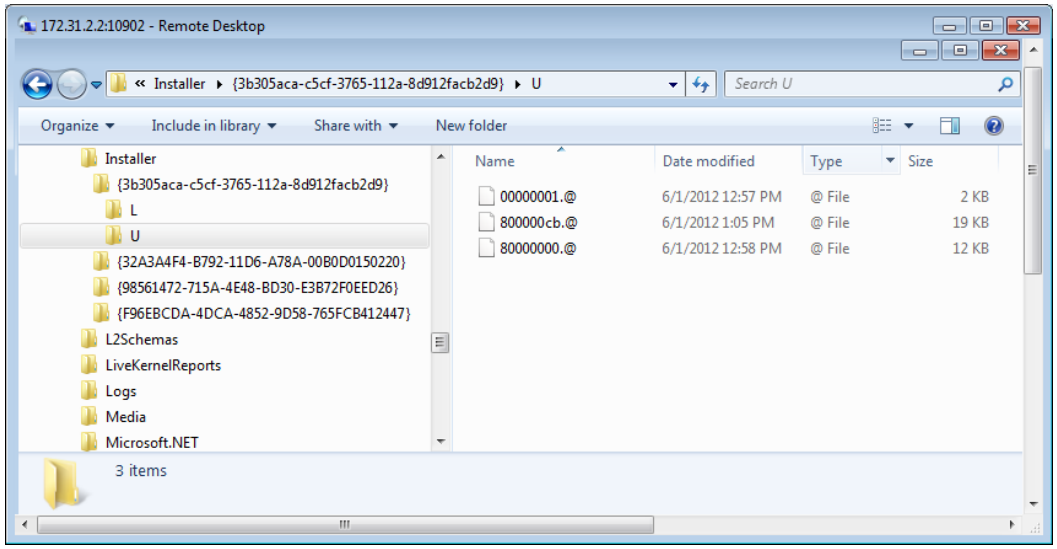
Get File requests are sent unencrypted!

Send File / Get File reply (TCP)

The download file is encrypted with RC4.

- The encryption key is the MD5 of the 12 byte Get File request itself.
- This ensures that each file is encrypted with a different key for each version of a file.

Before running and storing the file, the Bot will check it against the 128 byte signature it received in the retL file list.



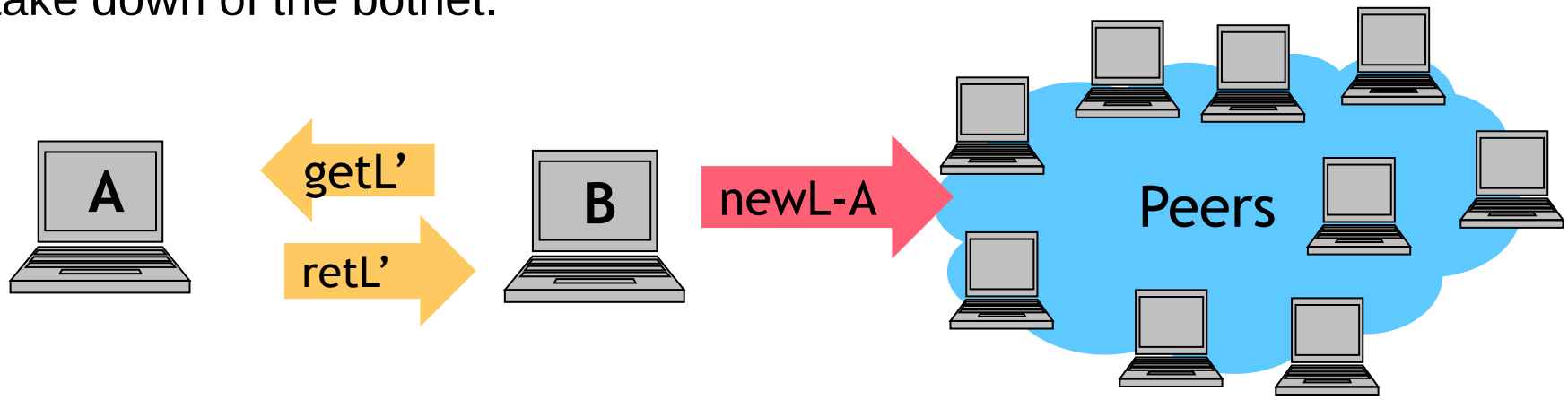
The downloaded files are saved in the ‘U’ directory.

newL Command (UDP)

CRC32	newL	Flag	IP-address
4	8	12	16

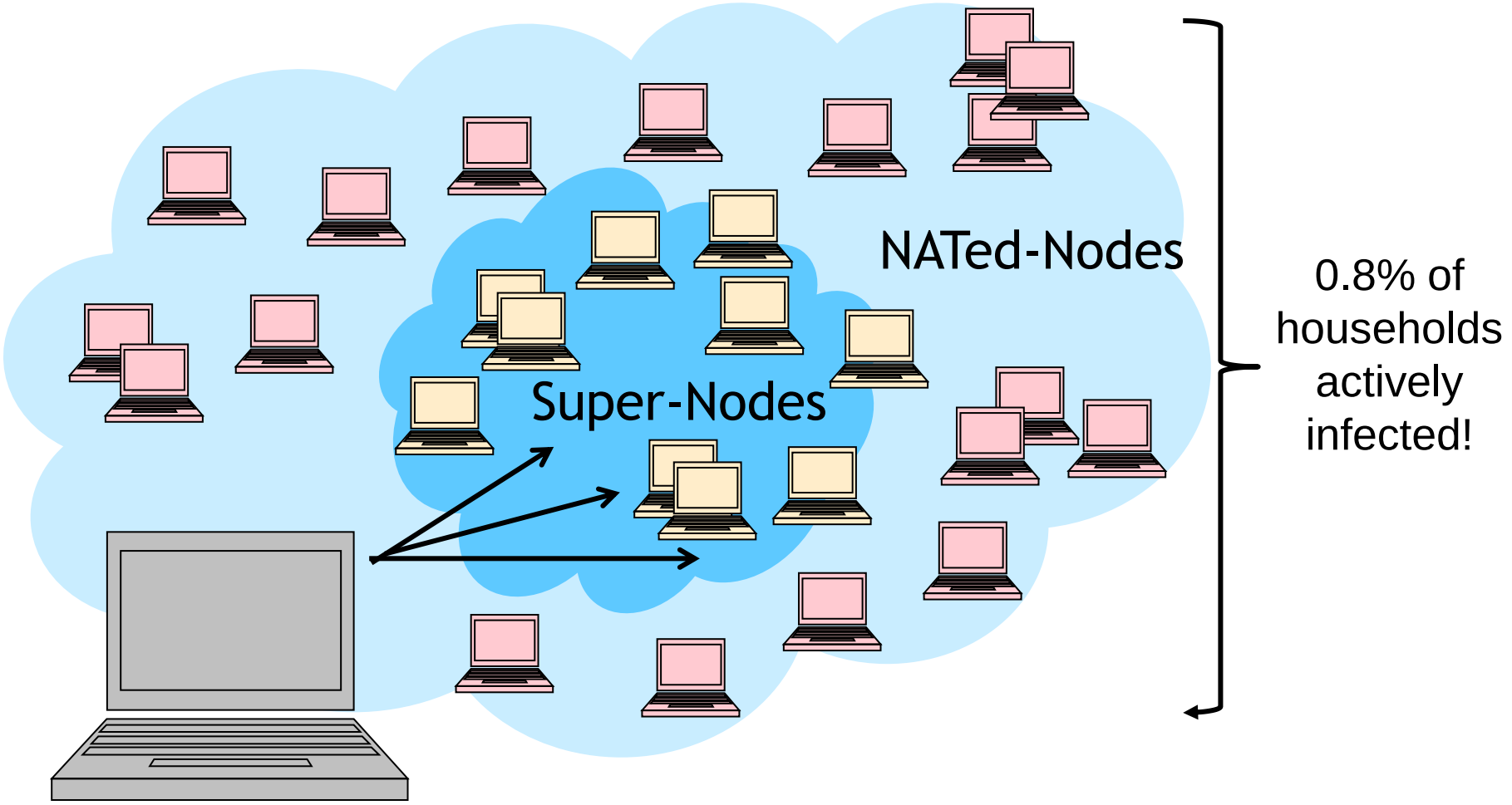
Command has not been seen used “in the wild”.

It has been suggested [Tan&Yang] that this command allows one node (B) to broadcast to its peers a request to add the specified peer (A) to their peer-list. It is also hypothesized this might be used to help thwart a take down of the botnet.



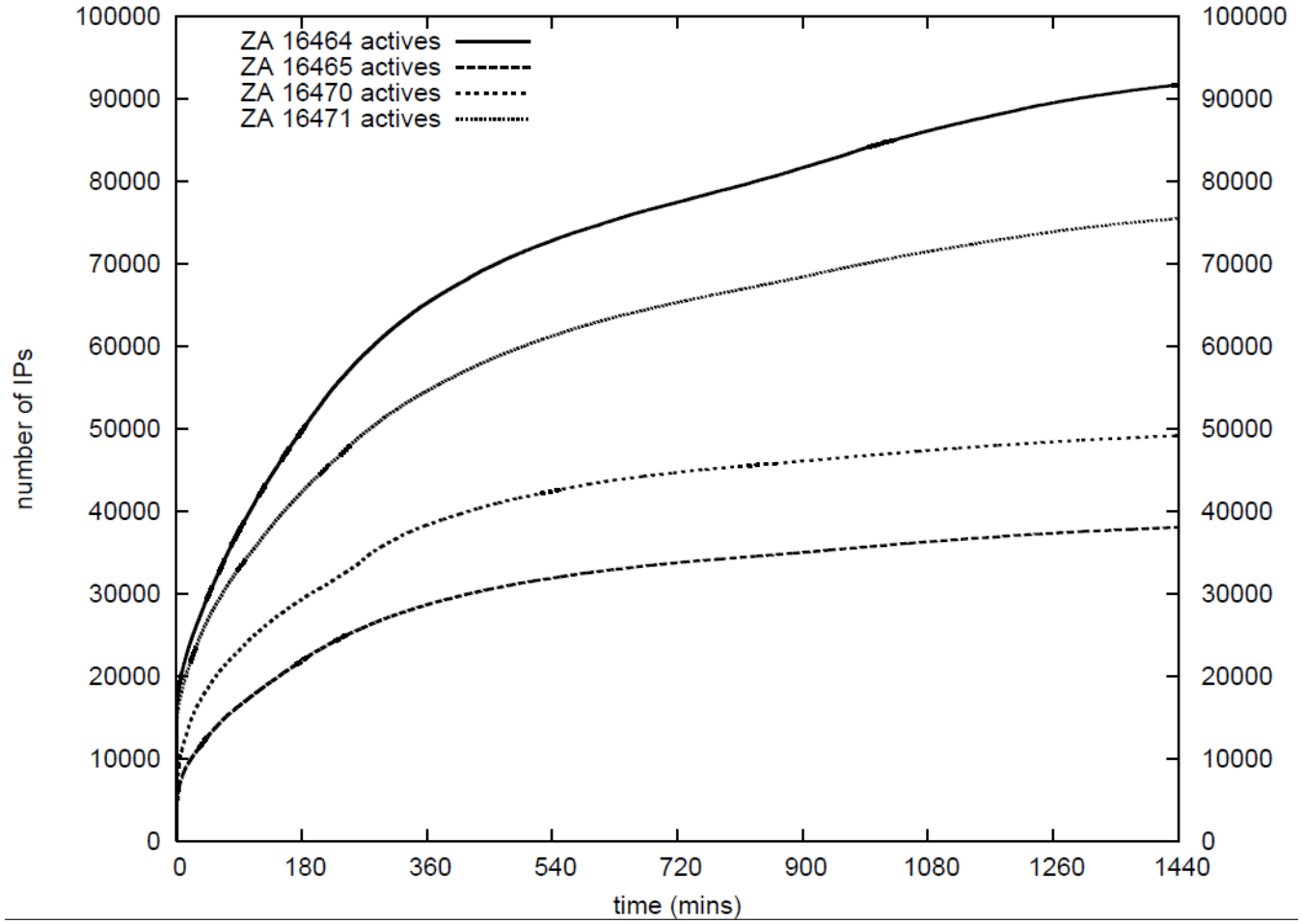
[Tan&Yang] *ZeroAccess Detailed Analysis, Virus Bulletin, Aug 2012, Tan&Yang, Fortinet* Canada

Structure of the BotNet.



On a daily basis, there are more then 200K Super-Nodes active

Crawler Results for four ZA2 Botnet Ports



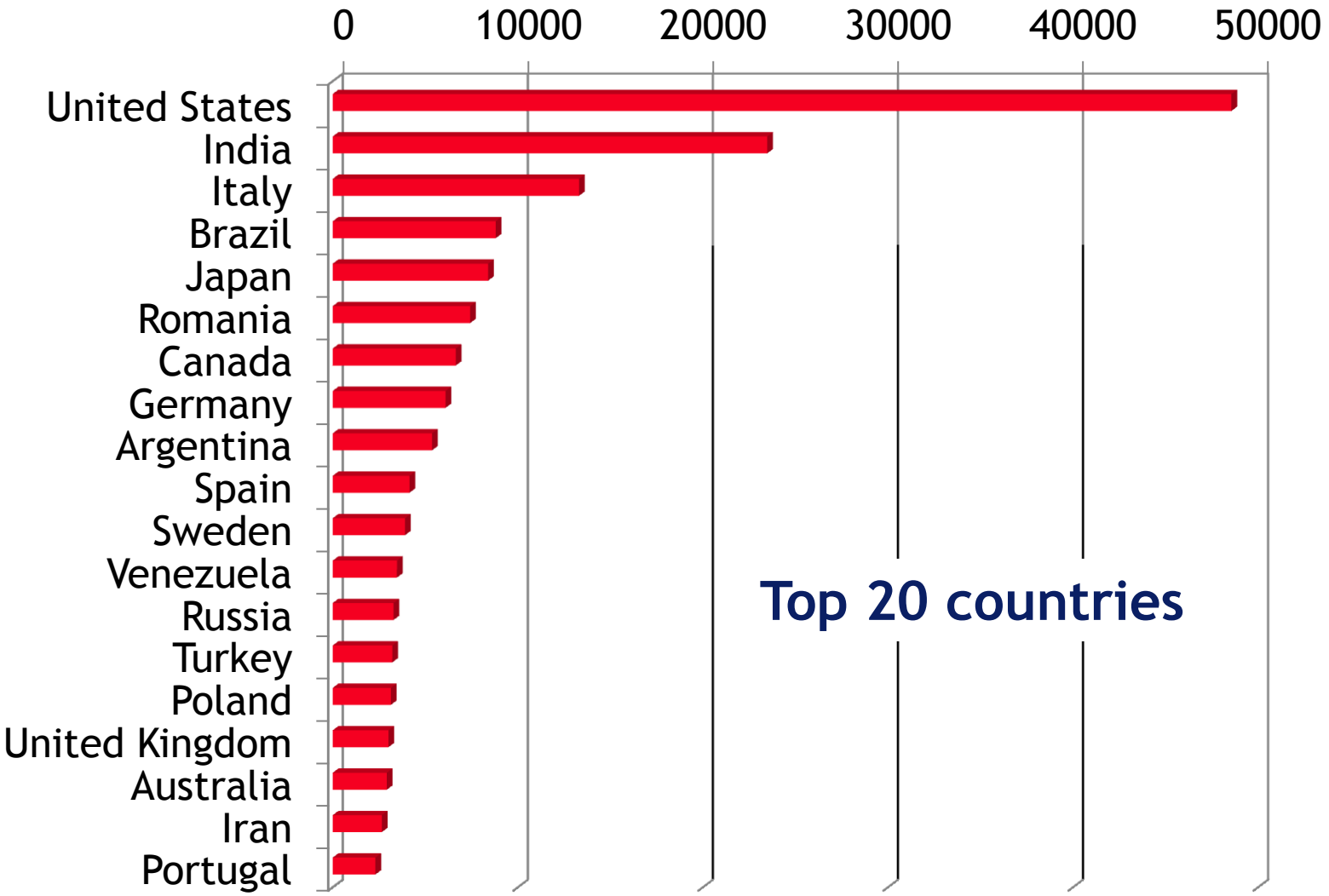
"Covers only non-NATed peers"

"The more popular networks serve 32 bit binaries, whereas the less popular serve 64 bit binaries."

Source:
Christian Rossow
VU University, Amsterdam
http://christian-rossow.de

Crawler results align with observed network traffic.

Where are the ZeroAccess-2 SuperNodes?



Estimating BotNet's overall size

North American Service Provider Networks monitored by Kindsight solutions see ~ 0.8% infection rates (one out of every 125 households are infected).

In 2009, the US alone had ~70Million households (Cable+Phone companies) with broadband connections.

Extrapolating the infection rate to entire broadband subscriber base, it is estimated that there are ~560K ZeroAcess actively infected machines in the US.

We know that there are ~48.5K SuperNodes active in the US on a daily basis. That gives us a ratio of 11.5 infections per 1 super-node.

Overall size of BotNet is estimated to be ~2.3 Million nodes

Making Money / Costing Money

Although nodes will come and go, the size of the BotNet's core remains large and relatively stable size wise day to day.

With 200K-1M nodes dedicated to performing Click-Fraud and BitCoin mining, it has been estimated [Wyke] that the operators could be earning more then $2 million per month.

ClickFraud, in addition to be being costly to the advertising industry, also generates considerable network traffic, which impacts service provider networks and consumers.

- ZeroAccess's ClickFraud trojan consumes about 0.1 MBits/second when averaged over a long period.
- For an individual user this adds up to 32GigBytes per month!!

[Wyke] *The ZeroAccess BotNet , Sept 2012, James Wyke, Sophos*

References

Kindsight's updated paper on which this presentation is based:

- http://www.kindsight.net/sites/default/files/Kindsight_Malware_Analysis-New_CC_protocol_ZeroAccess-final2.pdf
- Reference sample MD5: c71d6136d7549559ebddf65a48dd6a06

Other notable papers covering ZeroAcess 2:

- The ZeroAccess BotNet – Mining and Fraud for Massive Financial Gain
  - James Wyke, Sophos
  - http://www.sophos.com/en-us/why-sophos/our-people/technical-papers/zeroaccess-botnet.aspx
- ZeroAccess Detailed Analysis
  - VirusBulletin, August 2012
  - by Neo Tan and Kyle Yang, Fortinet Canada

Q & A

Kindsight™

Thank You

www.kindsight.net

John Morris

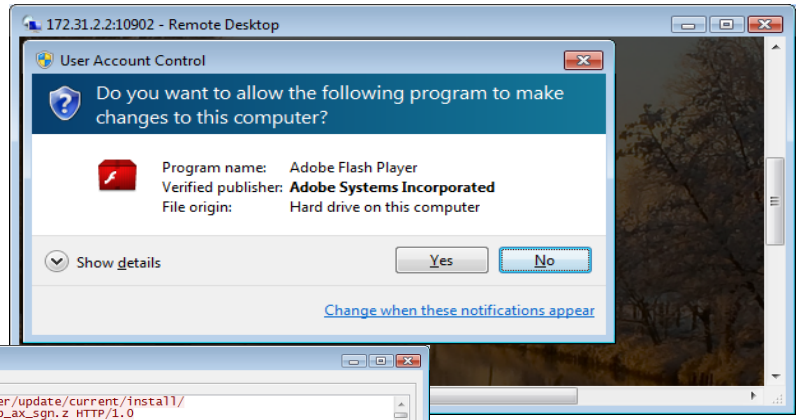
john.morris@kindsight.net

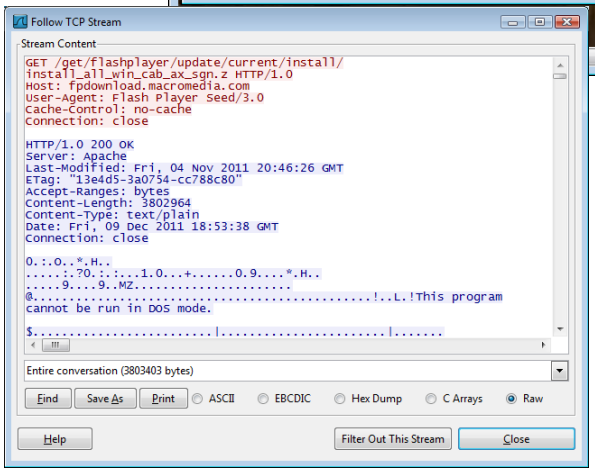
Backup Slides

ZeroAcess 1 - Infection Highlight

Version 1 used a hidden rootkit environment that sneakily gained Administrative access

- Downloads a legitimate copy of the Adobe Flash Player from fpdownload.marcomedia.com.
- It drops this and an infected copy of msimg32.dll into a temp directory.
- It then executes the Adobe installation program. This caused the system's UAC to ask for permissions to go ahead with the installation.
- The infected msimg32.dll is loaded with Adobe install process. *
- This takes advantage of the permissions that it has been granted to create a hidden root-kit environment.





These permissions allowed it to:

1. Create a hidden partition on the hard drive.
2. Install a shadow copy of the malware described in phase 1 into this partition.
3. Attach this copy to the "system process" (pid 4) and uses port 22292. It appeared to run independently from visible version but uses same exact C&C.
4. Add a device service to run a process called 3147163332.exe. This appears to be a watch dog process.

* Reported to Adobe and now fixed

Also see: "McAfee: ZeroAccess Rootkit Launched by Signed Installers"