Nethemba
all about security

# Modern ways to design fully distributed, decentralized and stealthy worms

Nethemba s.r.o.

Norbert Szetei, CEH
norbert.szetei@nethemba.com

# Basic terminology

- computer virus
- computer worm
    - active
    - passive
- botnet
- command and control (rendezvous point)
- exploits

# Formal Definition

- A Formal Definition of Computer Worms and Some Related Results by Fred Cohen
- definition of the worm with Turing Machine model
- detection is undecidable because of the halting problem
- heuristic methods are often only possible ways for detection

# Goals

**Optimalization of the following conditions and principles:**

- communication
- self-replication
- polymorphism

# Communication

- decentralization, P2P, DHT, DOLR
- dynamic environment → redundancy property necessary
- compromising of one node should not lead to gaining control over the whole network
- digital signatures, encryption

# Self-Replication

- optimal propagation can be mutual exclusive conditions
    - extremely rapid spreading
    - stealthy
- appropriate environment for spreading (web application, OS layer)
- analyzing an epidemic model

# Polymorphism

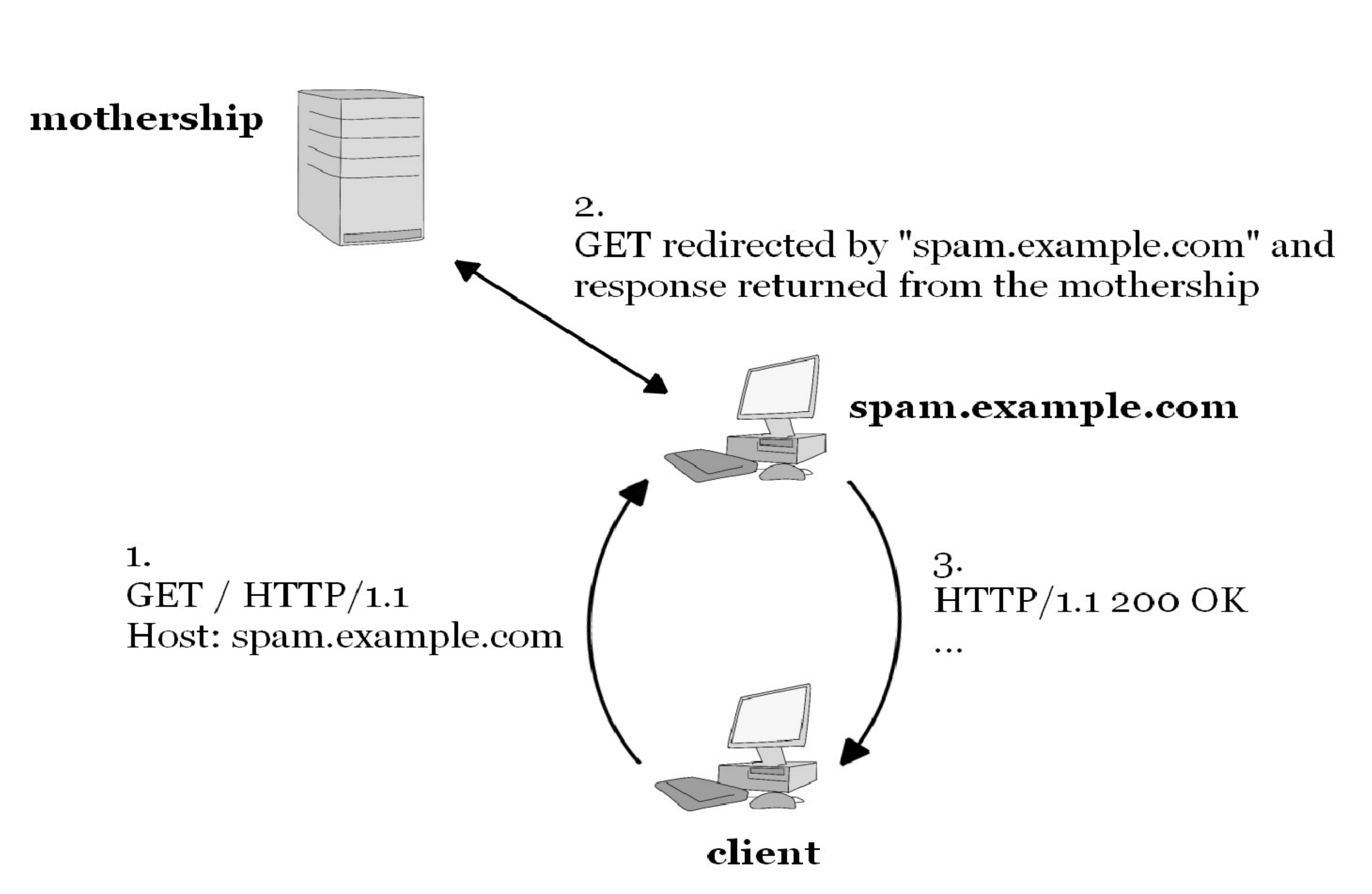
- server-side polymorphism
- IPS/IDS evasion techniques
- avoiding AV systems detection
- hiding the true purpose of wormnet

# Existing worms analysis

- several dangerous worms lacks of clever architecture (Morris's worm, Love Letter)
- we choose the representative worms from distinct types
    - Waledac (Waled, Waledpak)
    - Warhol (SQL Slammer)
    - Conficker (Downup, Downadup, Kido)

# Waledac

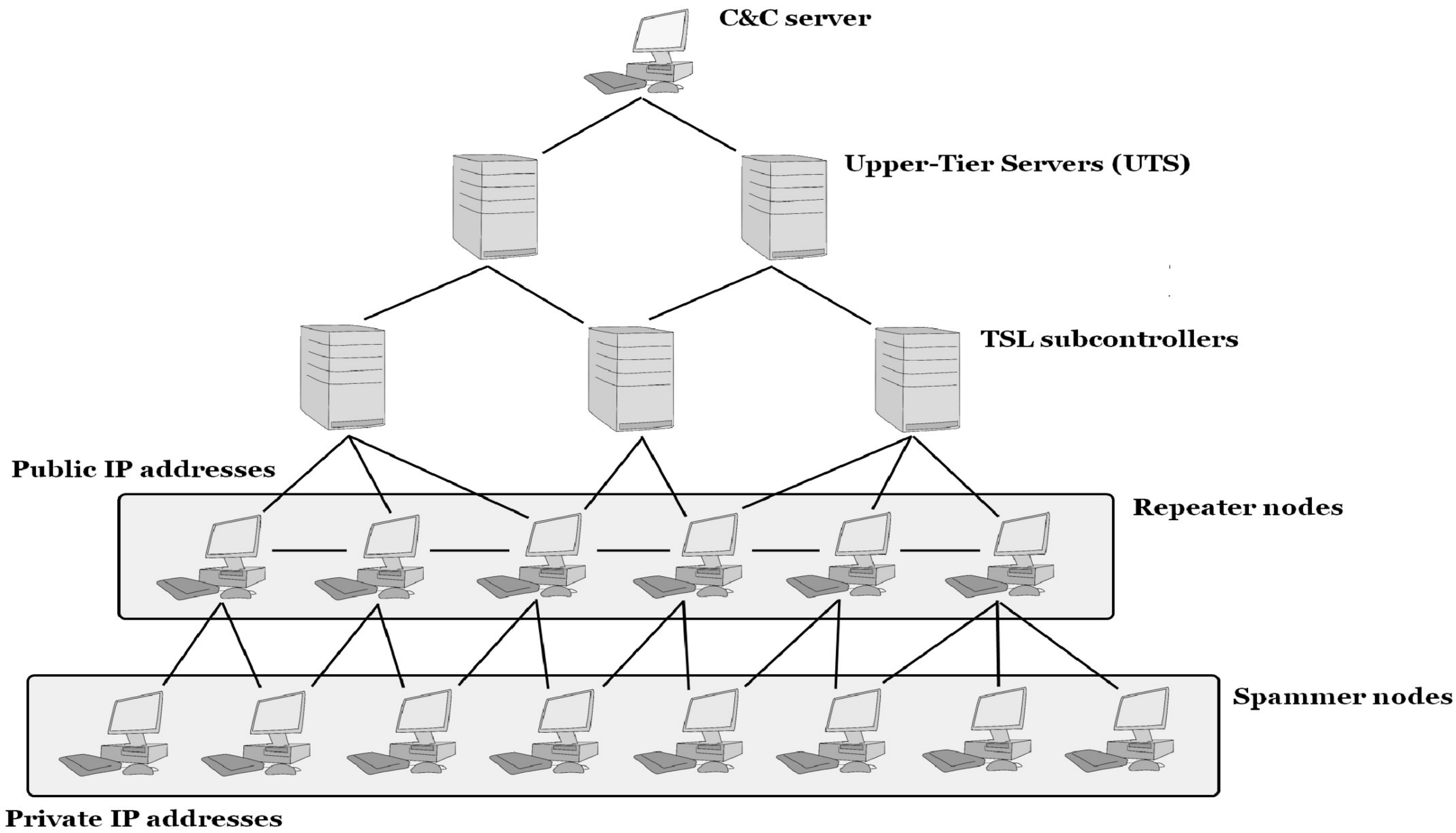
- came to attention at late of 2008
- similar to famous storm botnet
- passive worm, replication using emails pointing to Fast-Flux network
- layered P2P network, encrypted XML over HTTP
- taken down in March 2010 by Microsoft




**The client makes connection to web server, that is compromised in Fast-Flux network**

# Architecture

- spammers (private range)
- repeaters (public range), hidden by Double Flux
- TSL (nginx)
- upper tier servers (UTS), C&C
- encryptions for different purpose AES, RSA, SHA1 and BZIP2 (with base64)


C&C server
Upper-Tier Servers (UTS)
TSL subcontrollers
Public IP addresses
Repeater nodes
Spammer nodes
Private IP addresses

# Communication

- bootstrap binary with hardcoded IP addresses and URL
- locating the repeater node, sending encrypted public certificate
- propagation to the C&C, obtained the communication key (always the same)
- repeaters maintain the active nodes list (500-1000) digitally signed with timestamps

# Defense mechanism

- fixed location on disc
- Fast Flux domains can be blocked
- repeaters node depends on “X-Request-Kind-Code: nodes” string, IPS/IDS can match it easily
- “sinkhole” attack with TSL takeover

# Warhol Worm

- spreading as fast as possible
  a) scan machines for specific ports
  b) determine the runing service
  c) infect the target
  d) copy itself to new target
  e) repeat the process for each infected nodes
- spreading is exponentially fast

# Warhol Worm

- the best results using initial “hit list” (~5000 Ips)
- saturation (sigmoid function)
- SQL Slammer (January 2003), 75 000 victims in 10 minutes
- similar to the behavior of biological viruses (Kermack-McKendrick epidemic model)

# Simple epidemic model

- I(t) – infected machines at time t
- S(t) – susceptible machines at time t
- N – number of machines
- λ – average scan rate

$$\frac{\partial I(t)}{\partial t} = \lambda \frac{S(t)}{N} I(t) \qquad I(t) = \frac{i_0 N}{i_0 + (N - i_0) e^{-\lambda t}}$$

# Conficker

- firstly detected in November 2008
- the largest computer infection after SQL Slammer in 2003 (in 206 countries)
- A-E variants (named by The Conficker Working Group)
- MS08-067 buffer overflow
- installation of Waledac Botnet (Virus)

# Domain Generation Algorithm

- Conficker A generates 250 domain names
- when a valid IP address is found, it tries to download the binary from host (RP)
- validate digital signature of binary
- Conficker C enhanced generation to 50,000 domains, 500 was queried
- current time as seed, synchronization with adobe.com, facebook.com, seznam.cz, ...

# B, C variants

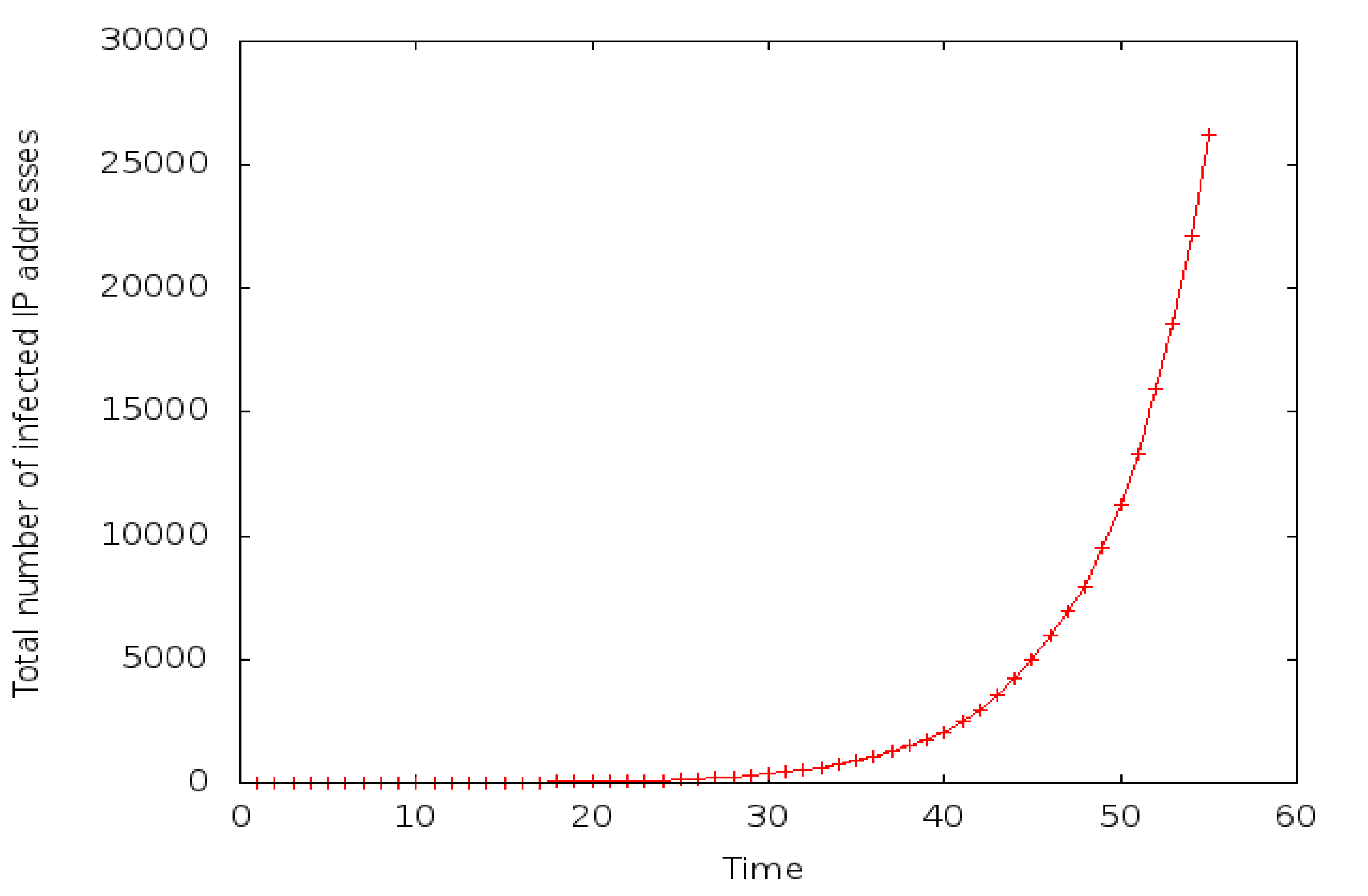
- Conficker B enhanced with dictionary attacks, patching MS08-067
- Conficker C leaves just 15% of code from A or B samples
- P2P network (download, upload functionality)
- RSA, RC4, MD-6 (the last one only a few weeks after releasing)

# Mitigation

- disabling security products, deleting Windows Restore points, aware of rootkits
- NMAP detection plugin (March 2009)
- Windows Malicious Software Removal Tool with patch to prevent reinfection

# Worm proposal

- **Short-time worm**
- NMAP Congruential Linear Generator
- difficult to do simulations because ISP complaints
- “harvesting” data, fixing issues caused by another worm




# Worm spreading according Apache 2.2.3 signature banner

# Stealthy worm

- according to the weaknesses in Waledac, we recommend to use an RSA key with the size at least 2048 bits for digital signature checks
- AV/IDS/IPS companies and worm creators' army race
- scripting language support (LUA interpreter has around 200KB)
- Worm.Win32.Leave

# Social networks as C&C I.

- worm's instructions can be stored to the Twitter account, we suppose that his name will be revealed and publicly known
- the botmaster publishes his instructions including the computed digital signature and valid timestamp through this account
- worm will check these instructions every few minutes and verify timestamp and the digital signature.

# Social networks as C&C II.

- after the worm reveals that account is blocked, he can use hash of this account
- the output is consequently converted into a set of allowable characters with cardinality n (radix or the number of displayable characters) that are used as a new user nickname during the registration
- the worm will try to follow all other instructions from this hashed account

# Example

- substr(MD5(mothership), 0, 10) = f052705d0a
- f052705d0a(16) = D66AHPFU(36)
- worm tries to connect to URL http://twitter.com/D66AHPFU
- substr(MD5(D66AHPFU), 0, 10) = bf0c1de19c
- bf0c1de19c(16) = AGY9JKWS(36)
- worm tries to connect to URL http://twitter.com/AGY9JKWS …

# Antivirus Bypassing

- Metasploit Framework Encoders
- Necessary to modify templates
- **PolyPack: An automated Online Packing Service for Optimal Antivirus Evasion**
- Virus Total uploader

# Code Armoring I.

- necessary to generate a (fixed) key that is derived from the environment
- unlike conventional methods, dynamically generated malware is not necessary on the disk, while it is prepared to run
- disk may contain any other application where instructions are ”transformed” in our malware (checksum verification)

# Code Armoring II.

- hash function provides a string of hexadecimal characters that are used for a substring that already represents the required instructions for binary (opcode)
- in order to get the desired output we have to generate salt appropriately and we can distribute the calculation over several computers
- detection possible using heuristics

# Example

- $ objdump -xs ./example

```
40043c:   48 83 ec 08             sub    $0x8,%rsp
400440:   48 8b 05 99 0b 20 00    mov    0x200b99(%rip),%rax
400447:   48 85 c0                test   %rax,%rax
40044a:   74 02                   je     40044e

40044c:   ff d0                   callq  *%rax
40044e:   48 83 c4 08             add    $0x8,%rsp
400452:   c3                      retq
```

- $ ./armour -k user -i 4885c0 -c 2

  (00001) MD5 of [userls']=[ba530ef1**4885c0**30225071efd4e1f405] at position: **09-14**
  (00002) MD5 of [user+RM]=[23013e**4885c0**75bc98d1a880b4f9a555] at position: 07-12

# Conclusion

- usage of distributed P2P architecture increases the worm's potential damage
- strong cryptography and accepting only digitally signed commands minimize the worm's exposure
- communication cannot be easily blocked
- it is relatively easy to hide bot master

# References

- Karamatlı E., Modern Botnets: A Survey and Future Directions
- Jon Oberheide, PolyPack: An Automated Online Packing Service for Optimal Antivirus Evasion
- Kennedy D., Metasploit: The Penetration Tester's Guide
- Mikko Hypponen, The Conficker Mystery
- http://mtc.sri.com/Conficker/
- Stuart Staniford, Vern Paxson, Nicholas Weaver, How to 0wn the Internet in Your Spare Time
- Sinclair G., The Waledac Protocol: The How and Why
- John Aycock, Anti-disassembly using Cryptographic Hash Functions

# Any questions?

# Thank you for listening

Norbert Szetei, CEH