A Survey of Contemporary Chinese DDoS Malware

Jeff Edwards
Dr. Jose Nazario

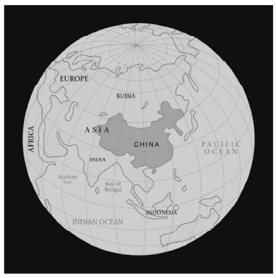
Arbor Networks
Ann Arbor, Michigan, USA

Agenda

- **What is "Chinese DDoS" malware?**
- **Summary of High-Level Findings**
- **Research Methods**
- **The "Typical" CN-DDoS malware family**
- **Command & Control Mechanisms**
- **DDoS Attack Capabilities**
  - DDoS techniques supported by attack engines
  - Attack Duration
  - Attack Method of choice
  - Target Preferences
  - Code Sharing
- **Conclusions**

What is CN-DDoS Malware?

- **We define "CN-DDoS" Malware as:**
  - CnCs hosted primarily in CN IP space; AND
  - Contains substantial DDoS capability



- **This definition excludes most:**
  - Infostealers
  - Spambots
  - Generic RATs
  - Basic Droppers

We make no claims regarding identity of malware authors, botnet operators, etc.

High-Level Summary of Findings

- **Large amount of diversity in CN-DDoS space**
  - Over 40 different families and counting

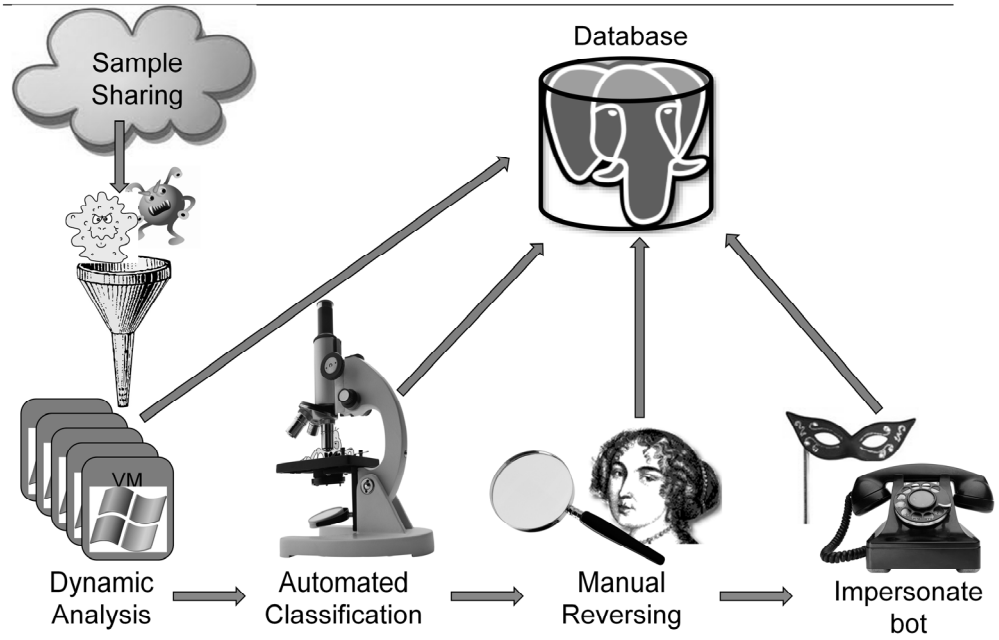
- **CN-DDoS malware is relatively unsophisticated**
  - Fairly rudimentary DDoS attack traffic
  - Very weak or non-existent encryption of comms
  - Little or no stealthiness

- **Code components "Mixed-and-matched"**
  - Attack engines, comms modules, etc.
  - Cross-pollination across families
  - Source code freely exchanged

Research Methods

- **Our Focus: understanding the characteristics of network traffic generated by malware families**
  - Command and control comms
  - DDoS flooding traffic
- **Secondary characteristics only interesting to the extent they aid automated classification:**
  - Installation/hiding mechanisms
  - Registry changes
  - Mutexes
  - Strings in binaries or process memory dumps
- **Reverse comms to write "fake bot" monitors**
  - Check in regularly with identified CnCs
  - Log DDoS attacks (target and attack type)

Research Methods



Page 6

The “Typical” CN-DDoS malware family

- **Written in straight C++ (Visual Studio)**
  - No Delphi, Visual Basic, C#, etc.
  - Very easy to analyze and reverse
- **Basic installation as a Windows Service**
- **Phones home via raw TCP connection to weird port**
  - Uses WinSock2: `socket()`, `connect()`, `send()`, `recv()`
- **CnC uses `3322.org` or `8866.org` domain**
- **CnC hosted somewhere on `CHINANET`**
- **CnC orders attack against single victim at a time**
  - Attack mode varies: HTTP, UDP, ICMP floods, etc.
  - Target is a web site (TCP port 80)
  - Victim usually hosts Chinese content
- **Attack lasts about two hours**

Command & Control Mechanisms

- **Pretty Simplistic Phone Home**
  - Usually contacts CnC via basic TCP socket
  - Binary C struct or **|**-delimited buffer
  - Sometimes HTTP request
- **Simple CnC Response**
  - Issues attack instructions
  - If HTTP-based CnC, then `.INI` response
- **Crude obfuscation (if any at all)**
  - Simple byte-wise substitution tables
  - No true encryption found so far

C&C Obfuscation: Example

```
0x0000:  00 00 00 10 a8 95 9b aa   95 91 de de de de de de      ........  ........
0x0010:  de de de de de de de de   de de de de de de de de      ........  ........
0x0020:  de de de de de de de de   de de de de de de de de      ........  ........
0x0030:  de de de de de de de de   de de de de de de de de      ........  ........
0x0040:  dc dc dc dc cc c9 c8 91   9c dc dc dc dc dc dc dc      ........  ........
0x0050:  de de de de de de de de   de de de de de de de de      ........  ........
0x0060:  de de de de a7 75 70 7a   6f 87 8b a6 ae de de de      .....upz  o.......
0x0070:  de de de de de de de de   de de de de de de de de      ........  ........
0x0080:  de de de de a8 75 8e cc   ce cd ce ce c6 cd de de      .....u..  ........
0x0090:  de de de de de de de de   de de de de de de de de      ........  ........
0x00a0:  de de de de 70 de de de   00 00 00 00 00 00 00 00      ....p...  ........
0x00b0:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00      ........  ........
0x00c0:  00 00 00 00 00 00 00 00   00 b0 fd 7f 00 00 00 00      ........  ........
0x00d0:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00      ........  ........
0x00e0:  b0 fc fl 00 00 00 00 00   ff ff ff ff 18 ee 90 7c      ........  .......|
0x00f0:  00 8e 91 7c ff ff ff ff   fa 8d 91 7c 25 d6 90 7c      ...|....  ...|%..|
0x0100:  cf ea 90 7c                                           ...|
```

Family **`darkshell`**: Obfuscated phone home

C&C Obfuscation: Example

```
0x0000:  00 00 00 10 56 49 43 54   49 4d 00 00 00 00 00 00      ....VICT  IM......
0x0010:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00      ........  ........
0x0020:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00      ........  ........
0x0030:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00      ........  ........
0x0040:  00 00 00 00 32 35 36 4d   42 00 00 00 00 00 00 00      ....256M  B.......
0x0050:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00      ........  ........
0x0060:  00 00 00 00 57 69 6e 64   6f 77 73 58 50 00 00 00      ....Wind  owsXP...
0x0070:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00      ........  ........
0x0080:  00 00 00 00 56 69 70 32   30 31 30 30 38 31 00 00      ....Vip2  010081..
0x0090:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00      ........  ........
0x00a0:  00 00 00 00 6e 00 00 00   00 00 00 00 00 00 00 00      ....n...  ........
0x00b0:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00      ........  ........
0x00c0:  00 00 00 00 00 00 00 00   00 b0 fd 7f 00 00 00 00      ........  ........
0x00d0:  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00      ........  ........
0x00e0:  b0 fc fl 00 00 00 00 00   ff ff ff ff 18 ee 90 7c      ........  .......|
0x00f0:  00 8e 91 7c ff ff ff ff   fa 8d 91 7c 25 d6 90 7c      ...|....  ...|%..|
0x0100:  cf ea 90 7c                                           ...|
```

Family **`darkshell`**: De-obfuscated phone home

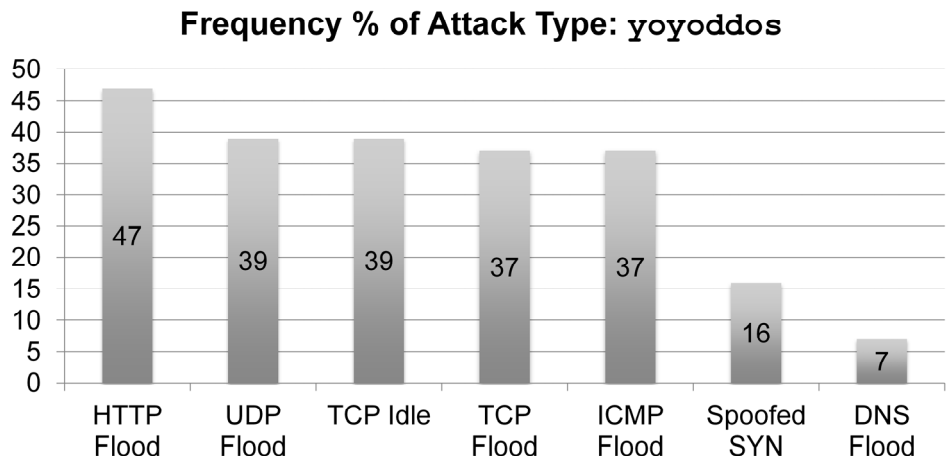
DDoS Attack Engines

Type of Attack	Families	%
WinSock2-based HTTP Flood	36	92%
UDP Flood	35	90%
TCP Flood	32	82%
ICMP Flood	31	79%
Spoofed SYN Flood	30	77%
TCP Connection Exhaustion	18	46%
TCP Idle Attack	12	31%
WinInet-based HTTP Flood	10	26%
Internet Explorer HTTP Flood	8	21%
Spoofed RST Flood	5	13%
URLMON-based HTTP Flood	2	5%
DNS Flood	2	3%

"Slow HTTP" Attacks noticeably absent

Observed CN-DDoS Attack Statistics

- **HTTP Flood attacks are most commonly observed**



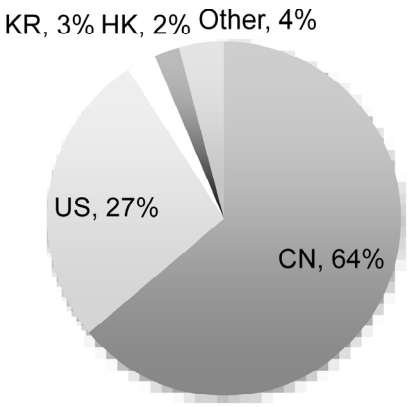
Observed CN-DDoS Attack Statistics

- **Median Attack Duration was 2 hours**
- **`darkshell` botnets engaged in longer attacks**
- **`chcod` botnets engaged in shorter attacks**

Family	Mean	Median	Min	Max
yoyoddos	10 hrs	2 hours	1 hour	45 days
avzhan	15 hrs	7 hours	1 hour	6 days
jkddos	30 hrs	6 hours	1 hour	30 days
darkshell	38 hrs	8 hours	2 hours	23 days
heartbreaker	13 hrs	2 hours	1 hour	9 days
chcod	8 hrs	4 hours	1 hour	2 days
TOTAL	**12 hrs**	**2 hours**	**1 hour**	**45 days**

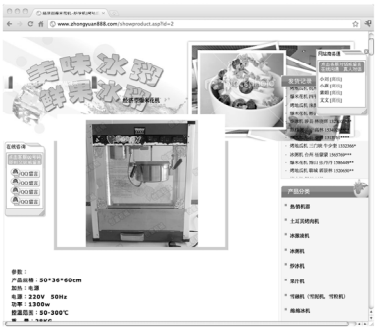
Observed CN-DDoS Attack Statistics

- Targets are most frequently hosted in China
- Targets were hosted in 24 different countries
- Almost 2400 unique targets attacked in 166 days



CN-DDoS Targets: `darkshell` Family

Chinese manufacturers of industrial food processing equipment



Page 15

CN-DDoS Targets: `jkddos` Family

- **International holding companies**
- **Global investment firms**
- **Mining-related investment projects**
  - In particular related to gold mining
- **Prominent NYC-based commercial real estate group**
  - Attacked on six different occasions in October 2010
  - Longest attack lasted approximately 33 hours

Code Sharing across CN-DDoS Attack Engines

- **Numerous examples of "cross-pollination"**
- **Same DDoS attack code shows up in multiple families**
  - Sometimes verbatim copied
- **Observed in most attack types:**
  - WinSock2 HTTP Flooding templates
  - Malformed HTTP Floods
  - TCP Flood payloads
  - Spoofed SYN Floods
  - UDP Floods
  - ICMP Floods

Code Sharing across CN-DDoS Attack Engines

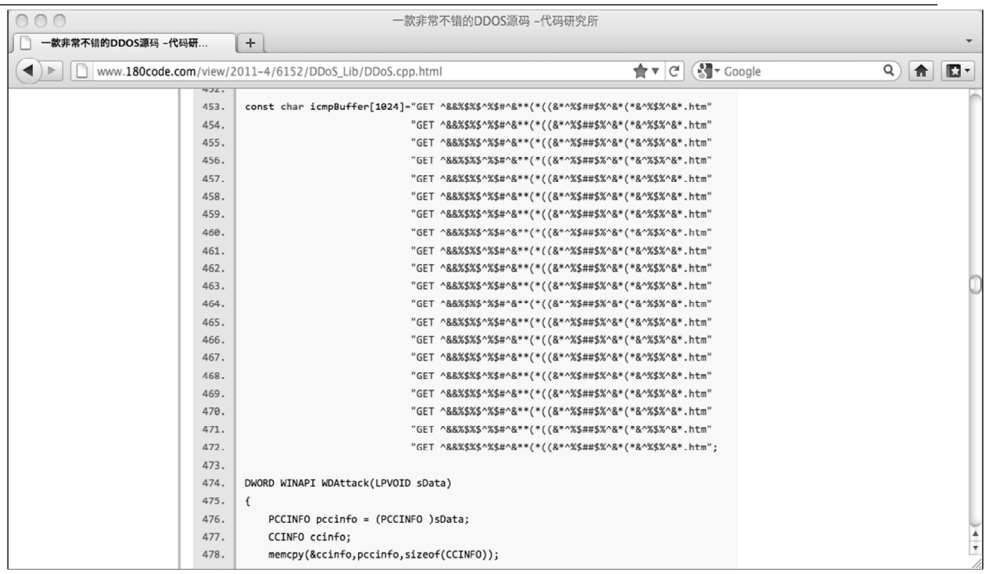
A `GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm`

B `GET ^*%%RTG*(&^%FTGYHJIJ%^&*()*&*^&%RDFG(JKJH.asp`

C `GET *(&*^TGH*JIHG^&*(&^%*(*)OK)(*&^%$EDRGF%&^.html`

- **Family `avzhan` TCP Flood**
  - Floods with 980-byte TCP requests containing 20 copies of **A**
- **Family `krbook` TCP Flood Type I**
  - Floods with 4096-byte TCP requests filled with copies of **A**
- **Family `rincux_hx` Attack code `tcp`**
  - Floods with 3944-byte TCP payloads filled with **A**, **B**, and **C**
- **Family `hackfans` HTTP Flood Types II and III**
  - WinInet-based HTTP floods to URLs formed with either **B** or **C**
- **Family `skingddos` Attack code `tcp`**
  - Sends 3944-byte UDP datagrams filled with **A**, **B**, and **C**

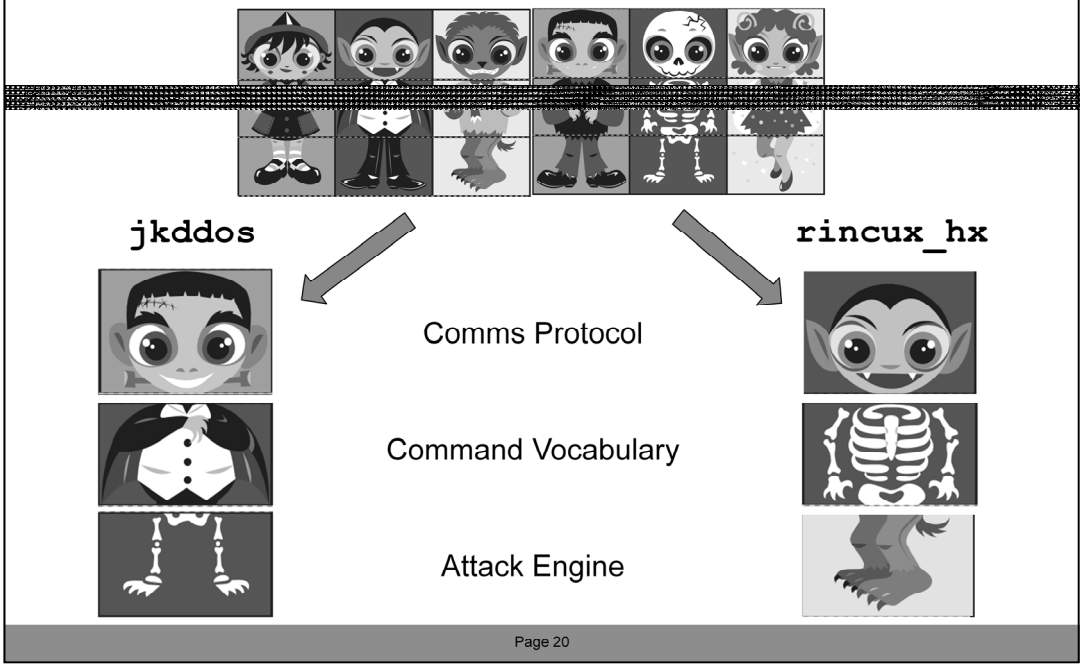
Code Sharing across CN-DDoS Attack Engines




```
453.  const char icmpBuffer[1024]="GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
454.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
455.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
456.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
457.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
458.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
459.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
460.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
461.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
462.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
463.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
464.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
465.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
466.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
467.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
468.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
469.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
470.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
471.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm"
472.                              "GET ^&&%$%$^%$#^&**(*((&*^%$##$%^&*(*&^%$%^&*.htm";
473.
474.  DWORD WINAPI WDAttack(LPVOID sData)
475.  {
476.      PCCINFO pccinfo = (PCCINFO )sData;
477.      CCINFO ccinfo;
478.      memcpy(&ccinfo,pccinfo,sizeof(CCINFO));
```

From: `DDoS_Lib/DDoS.cpp`, "*DDOS a very good source*"

Code Sharing across CN-DDoS Attack Engines



Page 20

Conclusions

- **The CN-DDoS families are relatively unsophisticated**
- **Attack traffic fairly rudimentary**
  - Lots of volumetric flooding attacks
  - Hard-coded HTTP flooding templates
  - No “Slow HTTP” capabilities yet
- **Little or no comms obfuscation**
  - Simple byte-wise substitution tables
  - No true encryption
- **Little or no stealthiness**
  - Very easy to analyze
  - Very little rootkit technology
  - Egregious typos in Windows service names, etc.
  - Sloppiness everywhere

Conclusions

- **Most CN-DDoS families appear to be assembled from reusable components**

- **Mix-and-Match components:**
  - Communications Protocols
  - Command Grammars
  - DDoS Routines
    - HTTP request templates
    - TCP/UPD Flood payloads
    - SYN/ICMP Flood routines
    - Entire Attack Engines

Questions?

Jeff Edwards
Dr. Jose Nazario
Arbor Networks – ASERT
asert@arbor.net