

Affaire suivie par :  
CERTA

## NOTE D'INFORMATION DU CERTA

### Objet : Évolution des outils de déni de service distribué

---

Conditions d'utilisation de ce document : <http://www.certa.ssi.gouv.fr/certa/apropos.html>  
Dernière version de ce document : <http://www.certa.ssi.gouv.fr/site/CERTA-2000-INF-003>

---

### Gestion du document

Date de la première version	29 mai 2000
Date de la dernière version	-

## 1 Introduction

Après les actions du mois de février 2000 à l'encontre de divers sites américains, les retombées médiatiques des manifestations "agressives" des outils de déni de service distribué semblent avoir décliné. Cependant, des réseaux d'agents et de serveurs sont régulièrement démantelés par les CERTs coopérant au sein du FIRST. Cela confirme que l'activité d'infection ne s'est jamais tarie et que des stations sont toujours activement compromises.

Par ailleurs, les premières versions ont inspiré d'autres pirates et de nouveaux outils continuent à apparaître.

La présente note a pour objet de décrire sommairement deux nouveaux outils d'attaque récemment apparus, trois programmes de détection et de donner des premiers éléments de conduite à tenir en cas de compromission.

## 2 Description

L'architecture d'un réseau de déni de service distribué reste identique à celle décrite dans la précédente note du CERTA (réf. 1), avec des agents pilotés par un ou plusieurs serveurs, ce(s) dernier(s) étant également accédé(s) à distance par un client dédié ou via telnet.

A ce jour, seuls les systèmes Unix peuvent être infectés par les deux outils décrits ci-dessous.

### 2.1 Shaft

*Shaft* est dérivé de *Trinoo*, les serveurs et agents communiquent via le protocole udp :

```
                20432/tcp                18753/udp
Client -----> Serveur -----> Agent
                <-----
                20433/udp
```

En l'absence de "rootkit", *netstat* donne :

#### Serveur

---

Secrétariat général de la défense et de la sécurité nationale – ANSSI – COSSI – CERTA  
51, bd de La Tour-Maubourg    **Tél.:** 01 71 75 84 50    **Web:** <http://www.certa.ssi.gouv.fr>  
75700 Paris 07 SP                **Fax:** 01 71 75 84 70    **Mél :** [certa-svp@certa.ssi.gouv.fr](mailto:certa-svp@certa.ssi.gouv.fr)

Protocole	Adresse locale
tcp	0.0.0.0:20432
udp	0.0.0.0:20433

## Client

Protocole	Adresse locale
udp	0.0.0.0:18753

Cependant les agents auraient la capacité de changer le serveur (modification de l'adresse IP) et le port auquel ils s'adressent (mais pas leur port d'écoute) sur ordre du serveur. Cela rend la détection des serveurs beaucoup plus aléatoire.

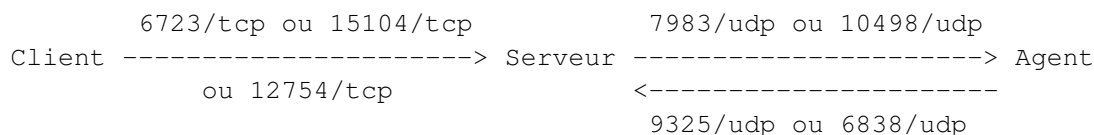
Lors d'une attaque, les agents envoient des paquets TCP, UDP, ICMP ou un mélange des trois, séquentiellement à chacune des cibles désignées. L'adresse source n'est falsifiée que lorsqu'un agent s'exécute avec les droits de root.

## 2.2 Mstream

*mstream* est outil de déni de service distribué peu sophistiqué basé sur le code du programme de déni de service *stream.c* apparu il y a quelques mois. Sa principale caractéristique est l'efficacité de l'attaque qui, par conséquent, nécessite moins d'agents que les outils semblables, pour un résultat équivalent.

Par ailleurs, le serveur prévient les clients déjà connectés de toute activité sur ses ports d'écoute (connection, authentification réussie ou non, nouvel agent). Il est donc important d'être en mesure d'isoler très rapidement du réseau (voir paragraphe 4) un hôte infecté par un serveur.

Plusieurs versions ont déjà été découvertes, d'où plusieurs ports caractéristiques possibles :



En l'absence de "rootkit", *netstat* donne :

## Serveur

Protocole	Adresse locale	
tcp	0.0.0.0:x	où x=6723, 15104 ou 12754
udp	0.0.0.0:x	où x=9325 ou 6838

## Client

Protocole	Adresse locale	
udp	0.0.0.0:x	où x=7983 ou 10498

Le serveur stocke la liste des clients identifiés dans un fichier nommé ".sr" par défaut.

Durant une attaque, les agents envoient des paquets TCP avec le drapeau ACK (accusé de réception) activé et une adresse source aléatoire. Lorsque le port destinataire de la cible est ouvert, le paquet nécessite une analyse plus poussée de la part de la pile IP de la victime qu'avec les leurres usuels avant d'être rejeté, et peut en outre induire un trafic supplémentaire dû à la notification du refus.

## 3 Outils de détection

Tout système de routage possédant une capacité de journalisation des connections peut être utilisé pour réaliser une détection sommaire de compromission. On peut mettre en place, sur un garde-barrière ou un routeur, les règles de journalisation suivantes (sans pour autant bloquer les paquets correspondants) concernant les ports sources ou destinations **internes** suivants :

**TCP :**

- 27665/tcp, serveur *Trinoo*,
- 27444/tcp, agent *Trinoo*,
- 16660/tcp, serveur *Stacheldraht*,
- 61111/tcp et 65512/tcp, serveur *Stacheldraht V4*,
- 65000/tcp, agent *Stacheldraht*,
- 65512/tcp et 65513/tcp, agent *Stacheldraht V4*,
- 20432/tcp, serveur *Shaft*,
- 6723/tcp, 15104/tcp et 12754/tcp, serveur *Mstream*.

#### UDP :

- 31335/udp, serveur *Trinoo*,
- 27444/udp, agent *Trinoo*,
- 34555/udp, agent *WinTrinoo* (*Trinoo* sous Windows),
- 20433/udp, serveur *Shaft*,
- 18753/udp, agent *Shaft*,
- 9325/udp et 6838/udp, serveur *Mstream*,
- 7983/udp et 10498/udp, agent *Mstream*.

Ces ports n'étant pas privilégiés, ils peuvent être attribués par le système d'exploitation à n'importe quel programme ouvrant une socket. Cependant leur apparition régulière dans les fichiers journaux peut être révélatrice d'une machine compromise. Il convient alors d'investiguer l'hôte correspondant pour déterminer si le port est utilisé par un programme légitime ou non.

Il existe également des outils dédiés permettant une analyse plus systématique, décrits ci-dessous.

### 3.1 Remote Intrusion Detector (RID)

*RID* est un programme de génération et capture de paquets IP. Il permet donc de détecter des programmes ayant une réponse réseau caractéristique pour une sollicitation donnée (réf. 5).

Après analyse des sources des versions 1.10 puis 1.11 (la version 1.11 est une modification de la 1.10 après suggestions du CERTA) et utilisation dans l'environnement du CERTA, il apparaît que ce logiciel ne semble faire aucune action nuisible et semble répondre à ce qui est décrit dans sa documentation.

Il est joint sous la forme d'une archive contenant les sources, une version exécutable pour Linux sur Intel et une version modifiée du fichier *rid.conf* de la version 1.11 en vue de détecter *mstream* accompagnée d'un mode d'emploi en français (réf. 2).

(voir les hachages md5 correspondants en annexe A).

### 3.2 Nessus

Nessus est un scanner de vulnérabilités réseau développé sous l'égide de Renaud Deraison résidant en France. Il fonctionne selon une architecture client/serveur. Le serveur, réalisant les tests, tourne sous Linux, Solaris, OpenBSD et FreeBSD. Il existe des clients permettant de le piloter depuis Windows, Linux, Solaris, FreeBSD ainsi qu'une version Java (réf. 6).

Les vulnérabilités sont testées à travers des scripts dont certains recherchent la présence d'outils distribués en envoyant des stimuli semblables à ceux de *RID*.

Les sources n'ont pas fait l'objet d'une analyse, mais le produit est utilisé dans l'environnement du CERTA sans effets nuisibles apparents.

### 3.3 find\_ddos

*find\_ddos* est un outil réalisé par le FBI. Il analyse les programmes de la station sur laquelle il s'exécute à la recherche de signatures caractéristiques des outils de déni de service distribué (réf. 7).

Le FBI ne distribue que des versions compilées (pour Solaris sur Sparc et Intel et Linux sur Intel) mais en revanche aucunes sources. Il n'est donc pas possible de connaître facilement le fonctionnement exact du programme et donc de se prononcer sur son innocuité.

Cependant, il a été utilisé avec succès, à plusieurs reprises, par des clients d'un autre CERT français sur des machines préalablement déclarées compromises (**en isolant l'hôte du réseau**).

## 4 Conduite à tenir en cas de détection

Il est essentiel de préserver l'ensemble des traces disponibles sur une machine (disque(s) et mémoire) et dans son environnement réseau. Il convient donc de ne pas intervenir directement sur la machine concernée jusqu'à la collecte. Les actions à entreprendre immédiatement (en consignant tout ce qui est entrepris) :

- supprimer l'accès réseau : au mieux en interdisant toute connexion depuis et vers la machine en configurant le routeur ou le garde-barrière le plus proche (configurer les journaux pour enregistrer le plus d'informations possible sur ces accès nouvellement bloqués, idéalement des "dumps" complets), au pire en débranchant le câble réseau,
- limiter l'accès physique au matériel,
- contacter le CERTA,
- en cas d'intervention obligatoire sur la machine : utiliser des programmes externes compilés statiquement (*ls*, *ps*, *netstat*,... sont très probablement compromis), noter toutes les commandes utilisées (utiliser *script* par exemple), minimiser les fichiers écrits localement.

## 5 Bibliographie

1. Note CERTA-2000-INF-001 "Le déni de service distribué" du 21 février 2000.
2. Note d'information CERTA-2000-INF-004 "Détection des outils d'attaque distribuée" du 29 mai 2000.
3. Analyse de *Shaft* par S. Dietrich, N. Long et D. Dittrich  
[http://netsec.gsfc.nasa.gov/spock/shaft\\_analysis.txt](http://netsec.gsfc.nasa.gov/spock/shaft_analysis.txt)
4. Analyse de *mstream* par D. Dittrich, G. Weaver, S. Dietrich et N. Long  
<http://staff.washington.edu/dittrich/misc/mstream.analysis.txt>
5. *RID*, détection réseau d'outils distribués par D. Brumley  
<http://theorygroup.com/Tools/RID/>
6. *Nessus*, scanner réseau de vulnérabilités par R. Deraison  
<http://www.nessus.org>
7. *find\_ddos*, détection locale d'outils distribués, réalisé par le FBI  
<http://www.nipc.gov/trinoo.htm>

## Gestion détaillée du document

**29 mai 2000**

version initiale.

## A MD5 des fichiers joints dans l'archive

```
ed63bb3aa16274aa752a07f0ca1b56bb ./source-rid-1.11/CVS/Root
09cb555fcabe0ea63cd66ab2339de7c4 ./source-rid-1.11/CVS/Repository
bdedde1b34287bc6c59c751b6a887470 ./source-rid-1.11/CVS/Entries
3152f046cc4f8c30a88bc9651acba29c ./source-rid-1.11/.h
feac3b0da1b00fc2a0f9cce9debeaa6c ./source-rid-1.11/BUGS
94bd123fc2f7972b1cf2d679dacd5aee ./source-rid-1.11/CHANGES
88923787dd054a3b1c6a7196cd207f09 ./source-rid-1.11/COPYRIGHT
07aa4b71b1229170197f8695382e88f4 ./source-rid-1.11/CREDITS
2bd50652025b5b2f9d49b6eb9aa497e9 ./source-rid-1.11/Makefile.in
efcf547a500eb33a640f271c37a1f8f1 ./source-rid-1.11/README
86829804843dc9788c59930d9154fb98 ./source-rid-1.11/config.guess
9ea90fddfa35f0bcd7214df135afa32a ./source-rid-1.11/config.h.in
9b6423290a5b9e54465d120adc3ccd9b ./source-rid-1.11/config.sub
f74d1e202bfb83a809f6d7610522f140 ./source-rid-1.11/configure
faa519094d7f16c50f1ed0f6ddcd261a ./source-rid-1.11/configure.in
```

```

2269c0beddfc299507a59377127b0501 ./source-rid-1.11/functions.h
3152f046cc4f8c30a88bc9651acba29c ./source-rid-1.11/grammer.h
a1dfc11bef5c6f1b33db869fd51182ac ./source-rid-1.11/grammer.y
ca4c8e08be31723608758c97c4001862 ./source-rid-1.11/install-sh
fbdc6043ec14b7a8d5e3e68d4a65032a ./source-rid-1.11/listen.c
393d298125f8e03c13364afe447b308d ./source-rid-1.11/missing
264cf0e0c373684dfed5a27b4ae39890 ./source-rid-1.11/mkinstalldirs
7a079ac8b7ef1643572e6db205e60dc0 ./source-rid-1.11/myconfig.in
39995303524a9707eceb10bb9129360 ./source-rid-1.11/netconfig.h
29a1ebf1ebf610f8ea7f4d95ebc1fdc9 ./source-rid-1.11/parser.l
bcf94fcc9091197881b726338d7363e8 ./source-rid-1.11/pinger.c
98cdc88fe364a07463efee74036271e9 ./source-rid-1.11/rid.c
c9600224a6493bcabeaf3274e5b1adc0 ./source-rid-1.11/rid.conf
45e65e0d9d03c1f92ba845dcf7c40d1b ./source-rid-1.11/sender.c
ed63bb3aa16274aa752a07f0ca1b56bb ./source-rid-1.11/.deps/CVS/Root
b34a842f241e8e1ecc14ceee72e02096 ./source-rid-1.11/.deps/CVS/Repository
22da040614547628c3b2a933f9796021 ./source-rid-1.11/.deps/CVS/Entries
6a5c95e0ed4cbb36ac268bf7fe644b38 ./source-rid-1.11/.deps/grammer.P
471252c2e6a715f95eae0a105861c397 ./source-rid-1.11/.deps/listen.P
a105d37353fea53d000de4cff4f462fb ./source-rid-1.11/.deps/parser.P
b49fef1bc376b57ee16ca5520362615f ./source-rid-1.11/.deps/pinger.P
ba30677a015c38eb29ad51c9b700690c ./source-rid-1.11/.deps/rid.P
fef080fa416ec170450b6640fae3e333 ./source-rid-1.11/.deps/sender.P
a8de6b2a512c5007a018e5109802cf7e ./source-rid-1.11/.directory
bede090e900d2b5cb928d0d2d92097db ./binaire/rid
c9600224a6493bcabeaf3274e5b1adc0 ./binaire/rid.conf
d77ca6390f05ad7e805c63b0106a0537 ./rid.tar.gz

```