

Affaire suivie par :
CERTA

NOTE D'INFORMATION DU CERTA

Objet : Détection des outils d'attaque distribuée

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-004>

Gestion du document

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

1 Introduction

RID (Remote Intrusion Detection) est un outil permettant de tester les hôtes d'un réseau à la recherche de machines infectées.

C'est un programme C développé par David Brumley de l'Université de Stanford et disponible à l'adresse :
<http://www.theorygroup.com/Tools/RID/>

2 Avantages et inconvénients

Il existe d'autres outils disponibles avec des fonctionnalités similaires : *gag*, *dds*, *Zombie Zapper*,... *RID* possède la particularité de lire un fichier ASCII de configuration (*rid.conf*) qui permet donc de rajouter très facilement de nouveaux tests correspondant à l'évolution de la menace.

Toutefois l'ensemble de ces outils simulent le comportement d'un agent ou d'un serveur pour obtenir une réponse caractéristique d'une machine infectée. Il y a donc un risque de ne pas détecter un outil distribué dont les sources ont été modifiées et qui ne présente donc pas la même signature. Cependant, en cas de découverte d'une nouvelle version d'un outil, *RID* permet de tester rapidement l'ensemble d'un réseau en modifiant le fichier de configuration en conséquence.

3 Installation

3.1 Prérequis

De manière à garantir la meilleure portabilité, l'outil est livré sous forme de code source C compilable sous Unix (testé au CERTA sous Linux). Il nécessite par ailleurs la librairie de capture de paquets *libpcap* qui doit être installée préalablement. Cette dernière est également utilisée par *tcpdump* et donc présente conjointement, et est également disponible à l'adresse :

<http://ee.lbl.gov>

De plus, la compilation de l'analyseur du fichier de configuration nécessite *lex* et *yacc* (ou *flex* et *bison* en version libre).

3.2 Compilation - utilisation

La procédure est classique :

```
./configure
```

```
make
```

```
./rid <adresses ip> le fichier rid.conf étant présent dans le même répertoire.
```

Où <adresses ip> représente le réseau à tester sous la forme *adresse/taille_du_masque*¹.

Exemples :

```
./rid 192.168.10.12 teste uniquement la machine d'adresse 192.168.10.12
```

```
./rid 192.168.0.0/16 teste les adresses comprises entre 192.168.0.1 et 192.168.255.254.
```

Remarques :

- l'outil testant des ports "exotiques", son efficacité peut être grandement diminuée si il est utilisé à travers un firewall,
- les outils distribués étant dérivés les uns des autres, il peut y avoir des fausses alarmes dues à des comportements similaires (alertes *AgentTrinoo* et *WinTrinoo* pour la même machine alors que les 2 outils fonctionnent sous des OS différents). Cependant toute alerte doit donner lieu à une analyse de l'hôte concerné.

4 Options

La ligne de commande accepte un certain nombre d'options, énumérées ci-dessous :

-f fichier_liste_ip contenant la liste des machines à tester (1 par ligne),

-t durée temps d'attente des réponses (30s par défaut),

-b nombre maximum de machines testées simultanément (défaut 128),

-s taille nombre d'octets capturés en écoute (défaut 1500),

-c fichier_config (défaut *rid.conf*)

-n nombre nombre de pings pour tester les hôtes "vivants" (3),

-v mode verbeux,

-h aide.

5 Syntaxe du fichier de configuration

Le fichier *rid.conf* inclu est une suggestion du CERTA, tout retour d'expérience des utilisateurs permettra de l'enrichir.

Il est composé de plusieurs entrées encadrées par les mots clés "begin" et "end", chacune correspondant à une menace testée :

```
begin <IDENTIFIANT>
```

```
send <PROTOCOLE><OPTIONS>
```

```
recv <PROTOCOLE><OPTIONS> nmatch = <nombre>
```

```
end <IDENTIFIANT>
```

```
PROTOCOLE=: tcp | udp | icmp
```

```
OPTIONS =: ICMP_OPTIONS | UDP_OPTIONS | TCP_OPTIONS
```

```
ICMP_OPTIONS =: seq=<sequence> | id=<id> | type=<type ICMP> | code=<code ICMP> | data="<chaîne>"
```

```
UDP_OPTIONS =: sport=<port> | dport =<port> | data="<chaîne>"
```

```
TCP_OPTIONS=: sport=<port> | dport =<port> | data="<chaîne>"
```

send décrit le paquet ip envoyé,

¹ RID n'accepte pas de tester plus de 254 machines à la fois (x.x.x.x/16) ce qui encombre déjà beaucoup un réseau...

recv décrit le paquet ip attendu, nmatch représentant le nombre d'options qui doivent être vérifiées pour générer une alarme.

Exemple :

```
start WinTrinoo  
send udp dport=34555 data="png []..Ks 144"  
recv udp sport=35555 data="PONG" nmatch=1  
end WinTrinoo
```

Un paquet udp contenant la chaîne "png []..Ks 144" est envoyé vers le port 34555 de chaque hôte, une alerte *WinTrinoo* est générée si une réponse udp revient depuis le port 35555 ou contient la chaîne "PONG".

Gestion détaillée du document

29 mai 2000

version initiale.