Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
PREMIER MINISTRE

S . G . D . S . N
Agence nationale de la sécurité des systèmes d'information
CERT-FR

Paris, le 12 septembre 2016
N° CERTFR-2016-ACT-037

Affaire suivie par :
CERT-FR

BULLETIN D'ACTUALITÉ

Objet : Bulletin d'actualité CERTFR-2016-ACT-037

1 - Techniques d'évasion utilisées à l'encontre d'équipements de type NIDS

Introduction

Un système de détection d'intrusion réseau (Network-based Intrusion Detection System [NIDS]) permet d'observer et de journaliser le trafic transitant sur un réseau afin de détecter de potentielles intrusions d'attaquants sur celui-ci. Un NIDS est passif sur un réseau, il ne fait qu'observer sans interagir au contraire d'un IPS (Intrusion Prevention System) ou d'un pare-feu. Son but n'est pas de stopper une attaque lorsqu'il en détecte une, mais de faire remonter une alerte aux personnes en charge de la supervision.
Pour ce faire, un NIDS possède une base de signatures et/ou de comportements anormaux. Il observe le trafic, effectue une analyse sur les paquets qu'il reçoit puis s'il détecte une anomalie (paquets correspondants à une signature, comportement anormal d'une connexion TCP, scan de ports...), fait remonter une alerte et la journalise.
Un NIDS doit être robuste aux évasions et aux insertions, c'est-à-dire qu'il ne doit pas ignorer des paquets ou des segments qu'un système dont il assure la supervision accepterait ou inversement.
L'insertion consiste à faire accepter par un NIDS un paquet ou un segment que le système cible rejette. Le but étant, pour un attaquant, de mettre les deux systèmes dans un état différent pour mener son attaque. Le plus souvent, l'implémentation d'un NIDS fait l'erreur d'analyser un paquet comme valide, suppose que le système cible a accepté le paquet et l'accepte à son tour.
L'évasion consiste à faire l'inverse de l'insertion : faire accepter par le système cible un paquet que le NIDS ignore. Cela arrive souvent lorsque l'implémentation d'un protocole par le NIDS est plus restrictive que celle d'un client ou d'un serveur. Par exemple parce qu'un NIDS respecterait correctement une RFC contrairement à un serveur qui serait plus tolérant, voire laxiste, sur la syntaxe d'un protocole.
Les techniques d'évasion réseau (sur IP et TCP principalement) ont fait l'objet d'une publication pour la première fois en 1997 [1], mais le sujet est resté et reste actif, notamment à cause du nombre croissant de protocoles applicatifs qui sont apparus dans les années qui ont suivi.

Exemples d'attaques

Il existe de nombreuses techniques d'évasion, que ce soit sur la couche réseau, transport ou application du modèle OSI. Nous en donnons ici quelques-unes, mais elles ne représentent qu'une portion de ce qu'il est possible de faire.

attaques reposant sur la connaissance du réseau

Un NIDS seul n'ayant pas connaissance de la cartographie du réseau qu'il surveille, il est vulnérable à des attaques reposant sur l'envoi de paquets n'atteignant jamais les machines du parc, mais l'atteignant lui. En IPv4, les champs TTL (Time To Live) ou DF (Don't Fragment) peuvent être utilisés dans ce but. Par exemple, en fixant le champ TTL d'un paquet assez bas pour que les routeurs entre un attaquant et sa cible le rejettent, mais assez

haut pour qu'il atteigne un NIDS, une insertion est possible.

attaques reposant sur des différences d'implémentation

La grande majorité des techniques d'évasion reposent sur les différences d'implémentation des protocoles conséquentes aux ambiguïtés des standards définissant ceux-ci. Par exemple, les RFC d'IP et de TCP ne donnent pas de règle à suivre lorsque des paquets ou des segments reçus se recouvrent (c'est-à-dire qu'ils ont le même offset pour IP et les mêmes numéros de séquence pour TCP) (fragmentation overlap et segmentation overlap), chaque implémentation est libre de prendre le premier ou le dernier reçu. Si la politique adoptée par un NIDS est différente de celle de la cible d'un attaquant, les paquets ou les segments reçus par les deux ne sont pas identiques et des évasions ou des insertions sont alors possibles.
La gestion des timers en est un autre exemple. Pour plusieurs fonctionnalités, dont la fragmentation IP, un timer est utilisé par les implémentations du protocole. Lors de la réception de fragments par une pile TCP/IP, tant que tous les fragments n'ont pas été reçus, l'automate ne peut reconstituer les paquets. Pour éviter que cet état soit bloquant, un timer est enclenché dès qu'un fragment est reçu. La durée de ce timer n'est pas spécifiée par la RFC d'IP, si ces durées diffèrent entre un NIDS et les systèmes qu'il supervise alors il est possible de jouer sur celles-ci, par exemple en attendant que le timer d'un NIDS expire, pour provoquer insertions et évasions.

Contre mesures

Nombre des techniques d'évasion existantes sont difficilement contrables par un NIDS seul, celui-ci étant passif, il ne peut pas contrer les attaques reposant sur la connaissance de la cartographie du réseau par lui-même. De plus, tant qu'il existera des ambiguïtés dans les standards des protocoles, il existera des différences dans les implémentations de ceux-ci et donc des évasions en découleront. Néanmoins :

– Il convient de mettre régulièrement à jour le ou les NIDS d'un parc informatique pour que les correctifs de sécurité de ceux-ci soient appliqués.

– Posséder plusieurs NIDS différents permet de réduire le risque d'évasion, les faiblesses d'un IDS étant potentiellement comblées par un autre.

– L'utilisation d'équipements actifs comme un normaliseur de trafic est une solution pour lever les ambiguïtés pesant sur un NIDS mais pose des contraintes de performance.

Références

1. Thomas H. Ptacek, Timothy N. Newsham, Insertion, evasion and denial of service: Eluding network Intrusion Detection, 1997

2 - Rappel des avis émis

Dans la période du 05 au 11 septembre 2016, le CERT-FR a émis les publications suivantes :

– CERTFR-2016-ALE-006 : Campagne de messages électroniques non sollicités de type Zepto
– CERTFR-2016-AVI-292 : Multiples vulnérabilités dans Apache OpenOffice
– CERTFR-2016-AVI-293 : Multiples vulnérabilités dans Google Chrome
– CERTFR-2016-AVI-294 : Multiples vulnérabilités dans les produits Apple
– CERTFR-2016-AVI-295 : Multiples vulnérabilités dans les pare-feux Cisco
– CERTFR-2016-AVI-296 : Multiples vulnérabilités dans SCADA Siemens SIPROTEC 4 et SIPROTEC Compact
– CERTFR-2016-AVI-297 : Multiples vulnérabilités dans Google Android (Nexus)
– CERTFR-2016-AVI-298 : Multiples vulnérabilités dans WordPress
– CERTFR-2016-AVI-299 : Multiples vulnérabilités dans Wireshark
– CERTFR-2016-AVI-300 : Multiples vulnérabilités dans les produits Juniper
– CERTFR-2016-AVI-301 : Multiples vulnérabilités dans Xen

- CERTFR-2016-AVI-302 : Multiples vulnérabilités dans Asterisk
- CERTFR-2016-AVI-303 : Multiples vulnérabilités dans les produits Citrix

Gestion détaillée du document

12 septembre 2016 version initiale.