Supervision de sécurité et détection d'intrusion

Attaques contre les IDS

Guillaume HIET
guillaume.hiet@centralesupelec.fr

Equipe CIDRe, **CentraleSupélec**-Inria-IRISA(CNRS)

Novembre 2017

Notes

Typologie des attaques

Des attaques classiques

- Attaques « classiques » contre un service réseau :
  - Un IDS réalise des traitements sur des données réseau (en grande partie maîtrisables par l'attaquant)
  - Décoder des paquets réseaux est une tâche complexe
  - Exemples : débordement de tampon, injection de code (si utilisation d'un langage interprété), etc.
  - Peuvent conduire à un DOS ou une prise de contrôle de l'IDS (modification de la configuration, rebond sur le réseau surveillé, etc.)
- Attaques contre les interfaces d'administration :
  - Souvent des interfaces Web, pas toujours correctement sécurisées
  - Sensibles au injections SQL, XSS, etc.
  - Attaques contre l'authentification (mot de passe faible, absence de chiffrement, etc.)
  - Souvent dues à une mauvaise utilisation (configuration)

  $\rightarrow$ Solution : utilisation d'un réseau dédié à l'administration

Notes

Typologie des attaques

Des attaques plus spécifiques

- DOS par consommation excessive des ressources (CPU, mémoire)
  - Les NIDS réalisent des opérations couteuses (différence firewall/routeur)
  - Certains algorithmes peuvent entraîner une consommation CPU mémoire excessive sur des cas « problématiques »
  - 2 stratégies :
    1. Augmenter le traffic (DDOS, etc.) pour saturer l'IDS : peu discret...
    2. Générer du trafic correspondant aux cas problématiques pour saturer l'IDS : plus discret (et plus facile à mettre en œuvre) $\rightarrow$ attaques en complexité algorithmique
  - Attention au dimensionnement (souvent, pas de prise en compte du « pire des cas »)
- Evasion
  - Leurer le décodage protocolaire
  - Exploiter les limites du moteur de détection (Ex : polymorphisme, mimicry)
  - Leurer l'analyste de sécurité (Ex : inondation de faux +)
- Attaques contre le réseau de supervision

Notes

Evasions protocolaires : principes

- Un NIDS doit interpréter les paquets **exactement** de la même manière que chaque cible surveillée
- En pratique, impossible car :
  - il n'est pas placé au même endroit dans le réseau
  - les paquets sont interprétés différemment suivants les implémentations des piles réseau (ambiguïtés des spécifications)
  - la sonde ne dispose (généralement) pas des informations de contexte concernant les cibles
  - la simulation de toutes les possibilités est impossible (ressources limitées)
- Souvent, vérifications « lâches » pour limiter la consommation des ressources (CPU, mémoire)
- Le paquets observé sera-t-il reçu ? Comment sera-t-il interprété ?
- Objectif de l'attaque : exploiter les incohérences entre l'interprétation de l'IDS et celle de la cible
  - injecter des paquets que l'IDS rejettera mais pas la cible
  - injecter des paquets que l'IDS acceptera mais pas la cible (insertion)

Notes

Evasion protocolaire : exemple (TTL)

Notes

Pourquoi est-ce possible ?

- L'IDS effectue un contrôle inapproprié (trop laxiste ou trop restrictif)
  - « Facile » à patcher (il suffit de corriger le contrôle)...
  - ... mais attention au surcoût entraîné !
- L'attaquant exploite une ambiguité du protocole
  - Plus difficile à prévenir
  - L'IDS doit prévoir le comportement de la cible (qui peut varier)
- Quelques exemples d'ambiguités (cf Ptacek & Newsham 98)

Info Needed	**Ambiguity**
Network Topology	IP TTL field may not be large enough for the number of hops to the destination
Network Topology	Packet may be too large for a downstream link to handle without fragmentation
Target Config.	Destination may be configured to drop source-routed packets
Target OS	Destination may time partially received fragments out differently depending on its OS
Target OS	Destination may reassemble overlapping fragments differently depending on its OS
Target OS	Destination host may not accept TCP packets bearing certain options
Target OS	Destination may implement PAWS and silently drop packets with old timestamps
Target OS	Destination may resolve conflicting TCP segments differently depending on its OS
Target OS	Destination may not check sequence numbers on RST messages

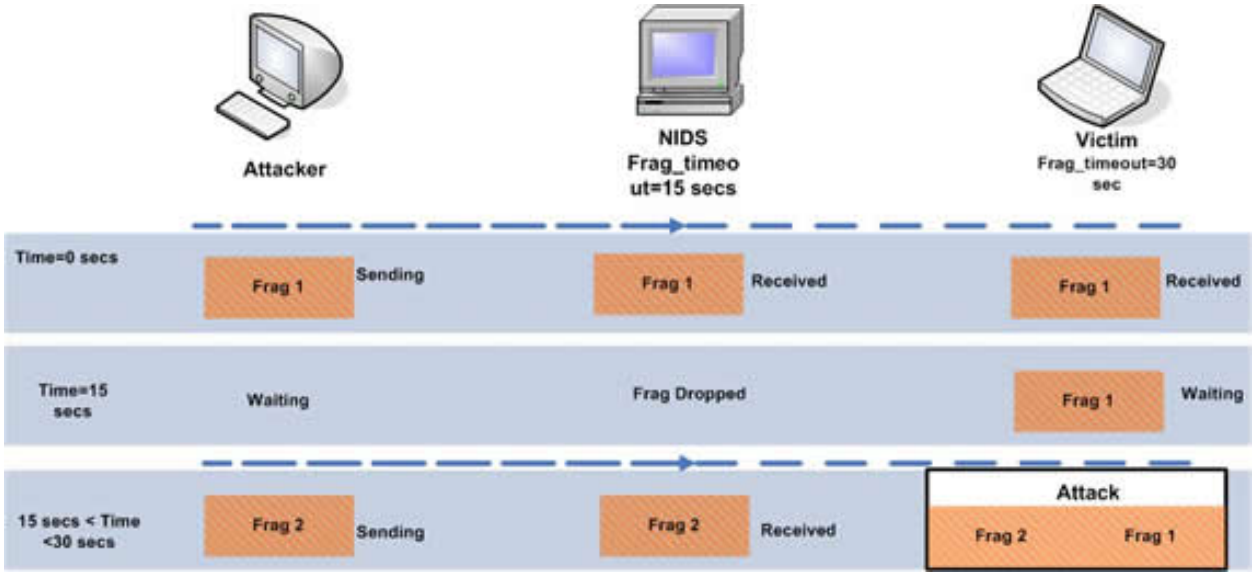
Notes

Evasion protocolaire : IP

- Insérer des paquets avec des champs d'en-tête incorrects (qui seront refusés par les piles TCP/IP)
  - Souvent acceptés par les IDS (vérifications laches)...
  - Mais rejetés par les pare-feux, routeurs, etc.
  - Effets de bords sur l'interprétation du paquets (exemple : size)
  - Exemple : version, checksum
- Ambiguïtés liées à la topologie
  - Modifier champs TTL si l'IDS n'est pas sur le même segment que la cible
  - Augmenter taille paquets + bit DF
- Ambiguïtés liées à l'OS
  - Source-routing
  - IP Timestamp
  - Ecouter les réponses ICMP ? Attention à la surcharge...
  - Fragmentation IP : émission dans le désordre, timeout, recouvrement, etc.
- Attention à l'effet des équipements réseaux situés en aval ou en amont
- Mais aussi : attaques sur la couche physique (MAC)

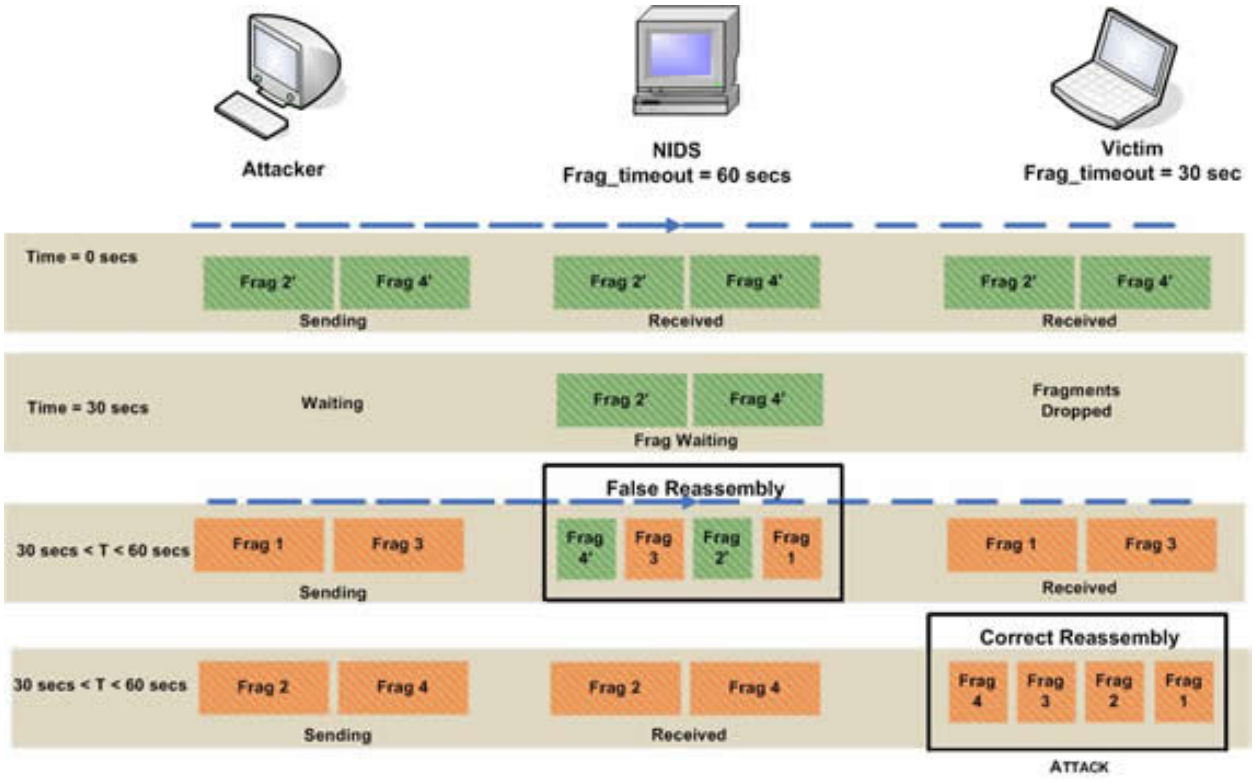
Notes

Exemple : Fragmentation Timeout



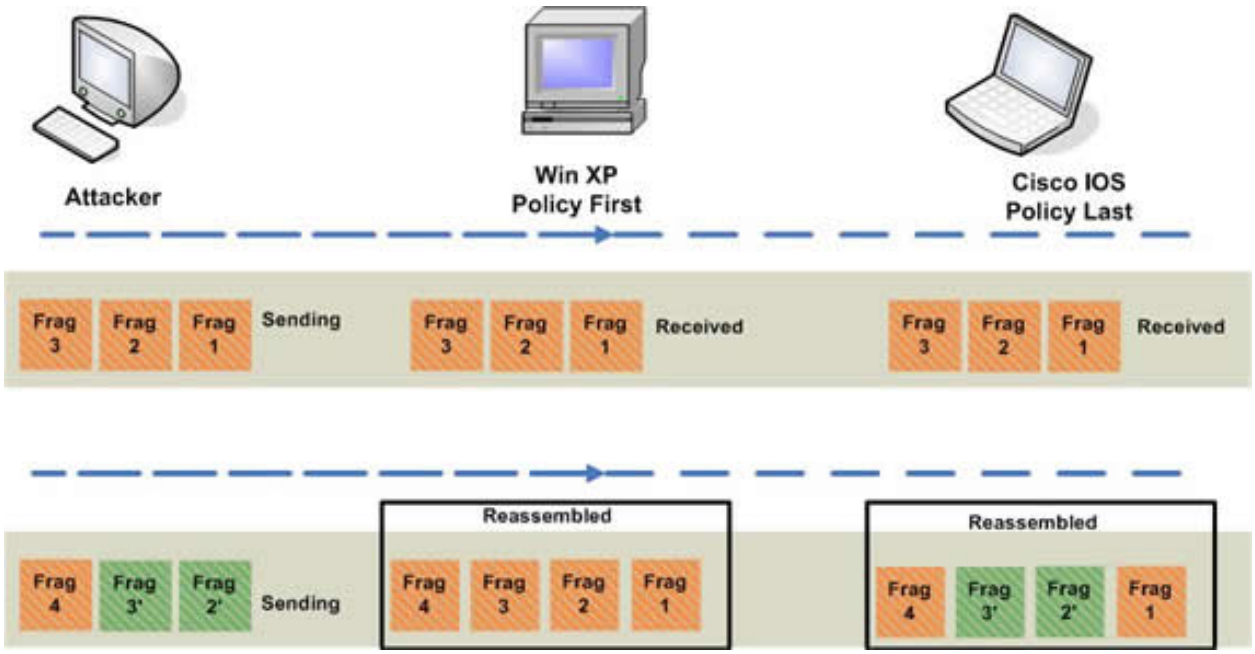
Notes

Exemple : Fragmentation Timeout



Notes

Exemple : Overlapping fragments



Notes

Evasion protocolaire : TCP

- Attaques similaires à celles concernant IP :
  - Insérer des paquets avec des champs d'en-tête incorrects (checksum, etc.)
  - Ambiguïtés : data + ack, data + syn, gestion des options (complexe), etc.
  - Différentes version du protocole, exemple : PAWS
- Reconstruction du flux TCP, suivi d'états : TCP Control Block
  - Création de la TCB
  - Reconstruction du flux (payload)
  - Destruction de la TCB
- Objectif de l'attaquant : désynchroniser l'IDS
- Faut-il s'appuyer strictement sur le « Three-Way Handshake » ?
- Rôles des autres équipement réseau pour éviter le *spoofing*
- L'IDS (passif) ne participe pas à la connexion : il ne peut émettre d'acquittement, etc. $\rightarrow$ Avantage du mode *inline*
- Problèmes liées à la reconstruction du flux : émission dans le désordre, recouvrement, timeout, longueur du flux, etc.
- Gestion (correct) des paquets FIN et RST, gestion des timeout

Notes

Pour aller plus loin...

Autres protocoles

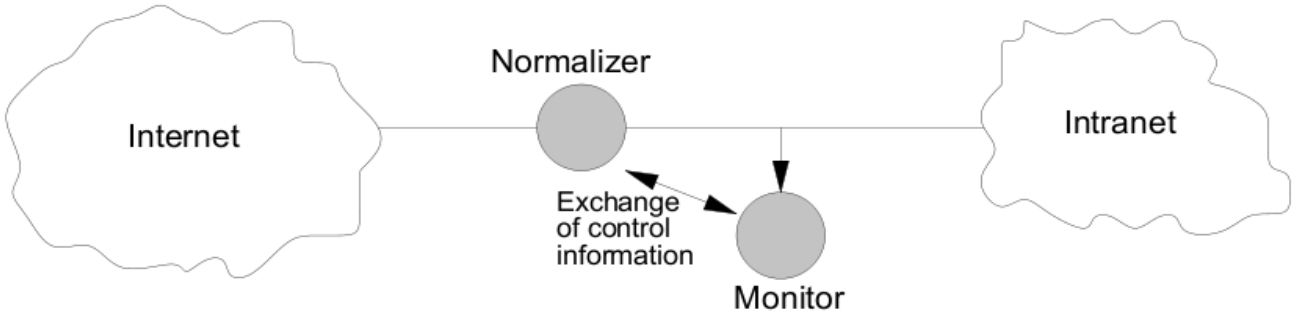
- Typiquement, couche applicative (N5)
- Exemples : HTTP, SMB, RPC, SMTP, FTP
  - fragmentation dans RPC
  - négociation des options FTP
  - encodage des URL, etc.
- Différents protocoles à supporter, différentes versions, implémentations, etc.
- Exemple : utilisation d'unicode
  - utf-16le, utf-16be, utf32-le, utf32-be, utf-7, utf-8, etc.
  - 125 manière d'encoder "A" pour UN *caracter set* !

Quelques outils

- Frageroute : `http://monkey.org/~dugsong/fragroute/`
- Wisker/Nikto : `http://www.cirt.net/nikto2`
- Scapy : `http://www.secdev.org/projects/scapy/`

Notes

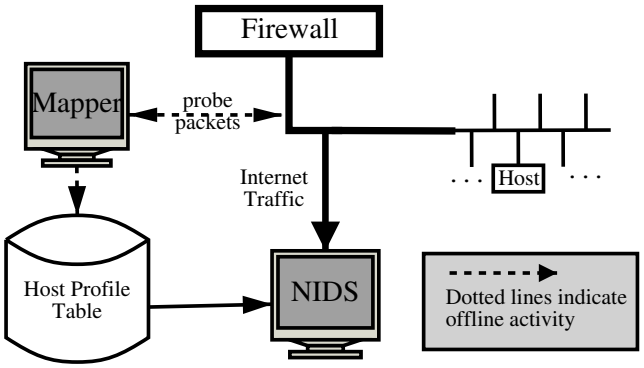
Contre-mesure : normalisation



- Positionné en amont, il normalise le trafic pour lever les ambiguïtés
- Compromis niveau normalisation vs. consommation ressource (maintient et suivi d'états)
- Respect de la sémantique du protocole (ex : TTL)
- Allège la charge de l'IDS
- Peut être implémenté dans un pare-feu ou un IPS (mais attention à la charge consommée...)

Notes

Contre-mesure : paramétrage de sondes



- Principe : fournir à la sonde les informations manquantes pour lever les ambiguïtés
  - Topologie du réseau surveillé
  - Configuration des cibles (OS, etc.)
- Limites
  - Fiabilité des informations collectée (passivement ou activement)
  - Interférence avec le réseau surveillé, stabilité
  - Surcharge de l'IDS

Notes

Bilan sur l'évasion protocolaire

- Il est en théorie facile de leurrer le décodage protocolaire des IDS :
  - Ressources limitées
  - Décodage des connexions de l'ensemble des cibles surveillées
  - Différences protocoles, versions, implémentations, etc.
  - Ambiguïtés (topologie, variation entre les implémentations, etc.)
- En pratique, cela est devenu plus difficile
  - Les implémentation des IDS ont évolué (exemple Snort : frag3, stream5, configuration par machine)
  - Les éléments réseaux placés sur le chemin de l'attaquant normalisent en partie le trafic
- Un problème toujours d'actualité
  - Cf blog de Julie Novak : `http://www.packetstan.com/`
  - Protocoles applicatifs, données interprétées côté client (javascript)
- Compromis ressources consommées vs. précision du décodage
- Combiner les approches : normalisation, utilisation des informations de contexte
- Spécialiser les sondes, recourir au HIDS (ou au proxy type WAF)

Notes

Bibliographie

[CW03] Scott A. Crosby and Dan S. Wallach, Denial of service via algorithmic complexity attacks, SSYM'03 : Proceedings of the 12th conference on USENIX Security Symposium (Berkeley, CA, USA), USENIX Association, 2003, pp. 3–3.

[HPK01] Mark Handley, Vern Paxson, and Christian Kreibich, Network intrusion detection : evasion, traffic normalization, and end-to-end protocol semantics, Proceedings of the 10th conference on USENIX Security Symposium - Volume 10 (Berkeley, CA, USA), USENIX Association, 2001, pp. 9–9.

[KKMR05] Christopher Kruegel, Engin Kirda, Darren Mutz, and William Robertson, Automating mimicry attacks using static binary analysis, Proceedings of the 14th conference on USENIX Security Symposium, vol. 14, 2005, p. 11.

Notes

Bibliographie (cont.)

[KL06] Oleg Kolesnikov and Wenke Lee, Advanced Polymorphic Worms : Evading IDS by Blending in with Normal T USENIX Security Symposium, 2006.

[PN98] Thomas H. Ptacek and Timothy N. Newsham, Insertion, evasion, and denial of service : Eluding network intrusion detection, http ://www.securityfocus.com/data/library/ids.ps, January 1998.

[PSJ07] Chetan Parampalli, R. Sekar, and Rob Johnson, A practical mimicry attack against powerful system-call monitors, Tech. Report SECLAB07-01, Secure Systems Laboratory, Stony Brook University, 2007.

Notes

Bibliographie (cont.)

[SEJ06] Randy Smith, Cristian Estan, and Somesh Jha, Backtracking algorithmic complexity attacks against a nids, ACSAC '06 : Proceedings of the 22nd Annual Computer Security Applications Conference (Washington, DC, USA), IEEE Computer Society, 2006, pp. 89–98.

[SP03] Umesh Shankar and Vern Paxson, Active mapping : Resisting nids evasion without altering traffic, SP '03 : Proceedings of the 2003 IEEE Symposium on Security and Privacy (Washington, DC, USA), IEEE Computer Society, 2003, p. 44.

[WS02] David Wagner and Paolo Soto, Mimicry attacks on host-based intrusion detection systems, CCS '02 : Proceedings of the 9th ACM conference on Computer and communications security (New York, NY, USA), ACM, 2002, pp. 255–264.

Notes