Lutte contre les Dénis de Service Réseau

rstack.org

Renaud BIDOU

renaudb@radware.com

La preuve par l'exemple

- Basé sur une histoire réelle
  - Tentative d'extorsion
  - En russie
- Objectif
  - Analyser les attaques
  - Etudier les solutions possibles
    - Au niveau de l'infrastructure finale
    - Au niveau des opérateurs
  - Etudier les faiblesses conceptuelles

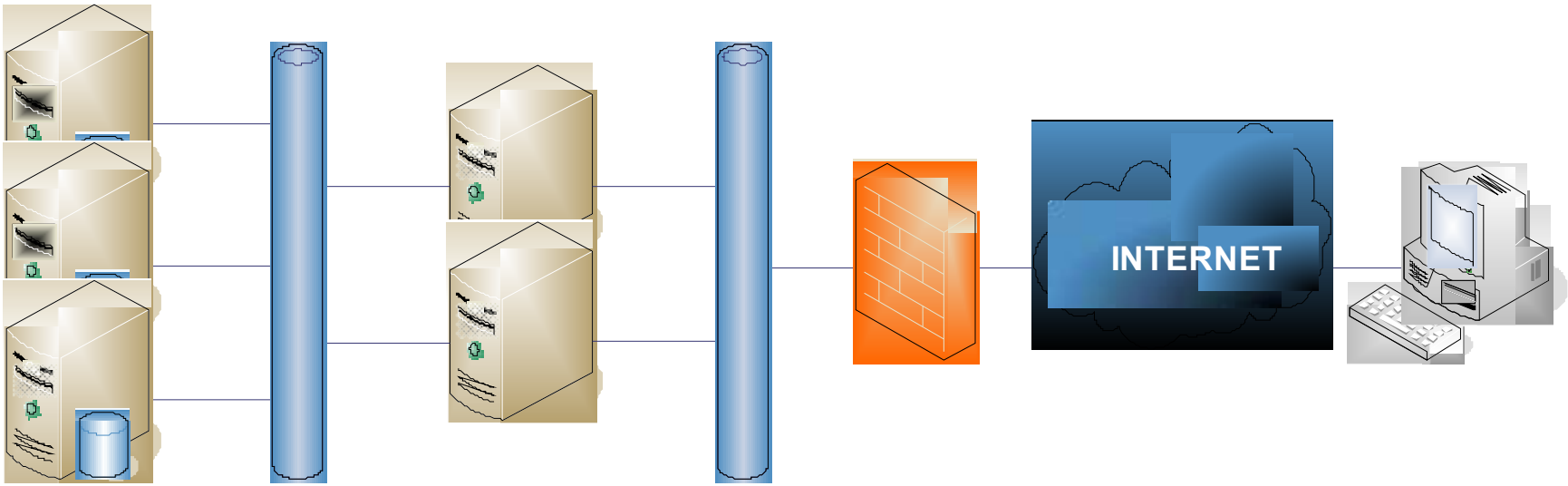
Première Vague

Contexte

La Cible

- Entreprise
  - Russe, basée à Moscou
  - Etablissement financier
  - Effectue des transactions de change
- Exposition aux DoS
  - Application propriétaire
  - Utilisée par les clients pour les transactions
  - 100% des opérations effectuées en ligne
- La cible idéale

Contexte | Architecture



Bases de données

Frontaux

Filtrage

Client
Logiciel Spécifique

Opérations des frontaux

- Aspects fonctionnels
  - Authentifient l'utilisateur
  - Récupèrent les requêtes
  - Formattent et transmettent aux bases de données
- Aspects techniques
  - Au-dessus de TCP, écoute sur un port élevé
  - Toutes les transactions sont chiffrées
    - Chiffrement propriétaire ☺
  - Paranoïa = Aucune autre information disponible

Extorsion

Extorsion | L'alerte

- t0
  - Dysfonctionnements identifiés
    - Serveurs frontaux "freezés"
    - Plus aucune connexion possible
- t0 + 15 mn
  - Trafic réseau analysé entre Internet et les frontaux
    - sources distinctes
    - 1 cible et 1 port destination
    - uniquement des SYNs (150.000 par seconde)
  - ➢ SYNFlood

Extorsion | Le chantage

- t0 + 30 mn
  - Contact via ICQ
  - X M$ de dollards doivent être versés
    - Avant t1 = t0 + 36h
  - Mode de transaction non dévoilé
- t0 + 60 mn
  - SYNFlood stoppé

Analyse

Caractéristiques d'un SYNFlood

- Petits paquets
  - 64 octets
  - Excellent ratio BP / Impact
  - 1 Mbps = 2.000 pps
- Spoofée
  - Aucun besoin de recevoir le SYN/ACK
  - Rend le traçage difficile

Analyse | Impacts d'un SYNFlood

- Sur la cible
  - Saturation de la TCB
    - Abandon des connexions existantes
    - Refus de nouvelles connexions
  - Effet de bord : saturation de la CPU
- Sur l'infrastructure
  - Dépassement de la capacité de traitement des paquets
  - Crash, reboot, freeze etc.

Protection de la cible – 1

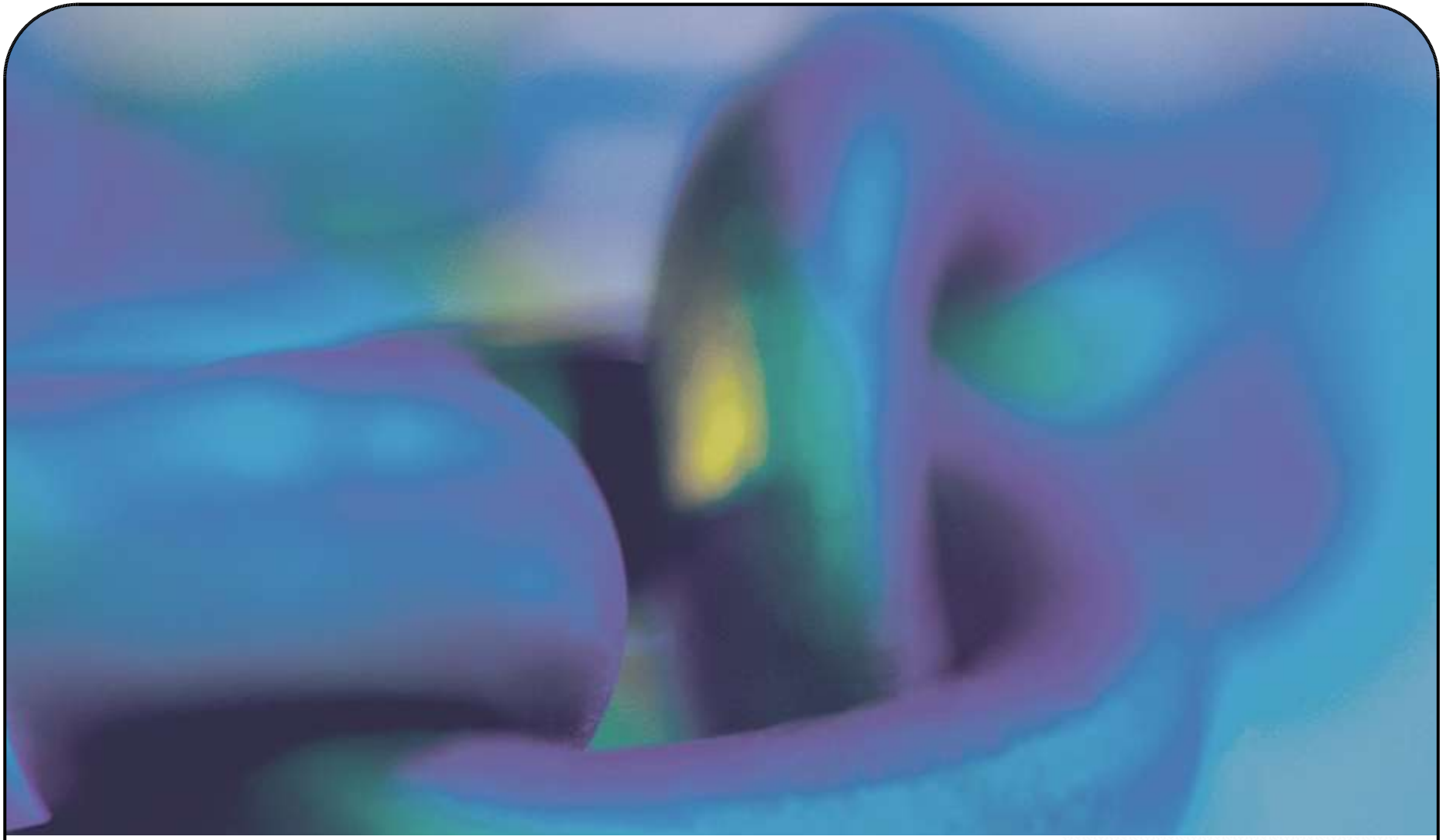
- Mise en place de SYNCookies
  - Session TCP établie en amont du serveur
  - Numéro de séquence du SYN/ACK obtenu par calcul
    - SYN_ACK_SEQ = f(net_params.time)
  - Transfert de ressources : TCB => CPU
  - Numéro de séquence ne doit pas être déviné
    - f() : fonction de hashage
    - SYN_ACK_SEQ = f(seed.net_params.time)
- Mise en place
  - Au plus prêt de la ressource à protéger
  - Derrière le firewall
  - Spécifique cible / port

Protection de la cible – 2

- Protection contre le spoofing
  - uRPF (*Unicast Reverse Path Forwarding*)
    - Strict : Bloque les paquets si le réseau source n'est pas dans la FIB (*Forwarding Information Base*) correspondant à l'interface entrante.
    - Loose : Bloque les paquets si la source n'est pas dans la RIB (*Routing Information Base*) du routeur (RFC 1918, reserved address etc.)
  - VRF (*Virtual Routing and Forwarding*)
    - Fournit à uRPF une table de routage par session eBGP.
- Mise en place
  - uRPF strict : Customer / ISP edge
  - uRPF loose : ISP / ISP edge
  - uRPF strict + VRF : ISP / ISP edge

Analyse | Protection de l'infrastructure

- Couper un bras pour sauver le corps
  - Mise en place de BHR (Black Hole Routing)
    - Application d'une règle de routage statique d'une adresse (@IP1) vers null0 (express forwarding, pas d'impact de performances)
    - Envoi d'un BGP Send : Next-hop pour la cible = @IP1
  - Aucun paquet ne peut atteindre la cible
    - Le Dénis de Service est un succès
    - L'infrastructure est sauve
      - Pas d'impact sur l'opérateur
      - Les autres systèmes du client restent opérationnels



Deuxième Vague

Prelude

Before the tempest

rstack.org

A quoi s'attendre ?

- Contraintes
  - BHR pas acceptable
  - Aucune solution ne peut être mise en place en 36h chez un opérateur
- Analyse
  - Sources spoofées
    - ACL non applicables
  - Port cible non privilégié
    - Connaissance de l'application par l'attaquant
    - Il peut être en possession du logiciel client
  - ➢ Attaques applicatives probables

Où mettre la protection ?

- Opérateur absent
  - Pas de retour
  - Aucune réactivité
- Sur plate-forme cible
  - Protection du firewall
    - Protection en amont pour éviter un crash dû à un nombre élevé de PPS
  - Application obscure et probablement mal développée
    - Protection en aval pour les éventuelles attaques applicatives

Attaque phase I

Phase I | SYNFlood - Le retour

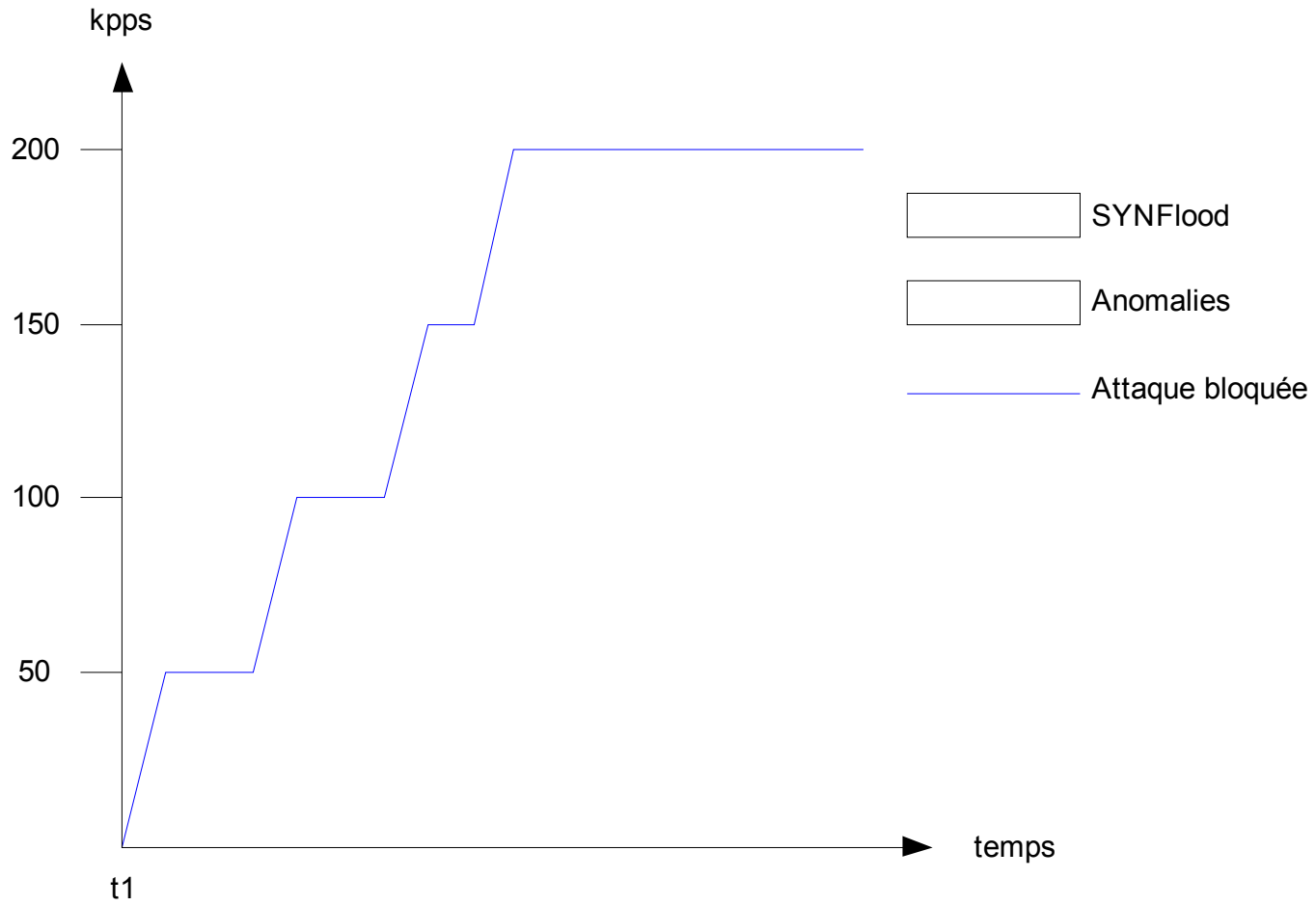
- Mode opératoire
  - t1 = t0 + 36h
    - SYNFlood identique à celui en t0
    - Puissance accrue par palliers
      - t1 + 5 mn = 50.000 pps
      - t1 + 10 mn = 100.000 pps
      - t1 + 15 mn = 150.000 pps
    - ✓ Bloqué par SYNCookies
  - ➢ Probablement un botnet activé séquenciellement

Phase I | Anomalies protocolaires

- Nouvelle attaque
  - t1 + 20 mn
    - Xmas Tree avec numéros de séquence à 0
  - Volume accru ~ 200.000 pps
    - Limite supportée par les routeurs
  - Nouveau type de trafic
    - t1 + 20 mn = SYN 150 kpps / Anomalies 50 kpps
    - t1 + 25 mn = SYN 100 kpps / Anomalies 100 kpps
    - t1 + 30 mn = SYN 50 kpps / Anomalies 150 kpps
  - ✓ Bloqué
  - ➢ A priori 4 botnets, reconfigurables avec une capacité maximale de 200.000 pps

Analyse Phase I

Phase I | Schéma de l'attaque



Phase I | Attaques par anomalies

- Basées sur des bugs ou des exceptions
  - Bugs :
    - Oldies : PoD, land, bo(i)nk, Xmas Tree…
    - Plus récemment : Taille des options
  - Exceptions
    - Valeurs limites ou incohérentes
    - Flags TCP, protocole, numéros de séquences etc.
  - Un seul paquet n'a plus d'impact
    - Mais le traitement de plusieurs kpps monte la CPU à 100%
    - Et merci pour le flag PUSH!

Phase I | Caractéristiques des attaques

- Caractéristiques des attaques
  1. Volume important
  2. Trafic “sortant de l’ordinaire”
  3. Sources spoofées
- Blocage par firewalls
  - Principe du blocage
    - Les attaques sont souvent effectuées en dehors de sessions TCP établies
    - Elles pourraient être bloquées par des firewalls stateful
  - Problématique
    - Petits paquets (en général TCP sans data ~ 70 octets)
    - Nombre important de paquets
    - ➢ Gros problèmes de performances

Phase I | Identification des attaques

- Modes de détection
  - Signatures paquet par paquet
    - ✗ Trop de signatures possibles * trop de paquets
  - Signature par échantillonnage
    - Analyse de 1 paquet sur n
    - Activation uniquement de la signature qui correspond
  - Heuristique
    - Détection de trafic sortant d'un format normal
    - Définition de signatures dynamiques

Phase I | Implémentation

- Mise en place
  - En ligne
    - Nécessite de nombreux équipements
    - Besoin de performances
      - Gestion de l'ensemble du trafic
      - Effectue Détection + Blocage
  - Architecture en 2 blocs
    1. Ecoute du trafic et détection d'anomalies
    2. Redirection du trafic suspect en fonction de la cible vers une « machine à laver »
    - Le système de blocage ne traite que du trafic suspect

➢ Les techniques d'anti-spoofing sont également efficaces

Attaque phase II

Phase II | Niveau applicatif

- t1 + 35 mn
  - Etablissement de connexions légitimes
  - Mode opératoire
    - Etablissement de sessions TCP complètes
    - 1er paquet de data contient un payload aléatoire
- Impact
  - A 5.000 sessions/s (20.000 pps)

GEL DES CONNEXIONS

Phase II | Réaction

- Besoin de reconnaître le trafic légitime
  - Seule information disponible
    - Les deux premiers octets du premier paquet sont “00 01”
  - Mise en place d’un filtre *stateful*
    - Transfert de la session au serveur après le 4è paquet
    - SYN / SYN-ACK / ACK / ACK(00 01)

✓ t1 + 45 mn : Attaque bloquée

Phase II | Trahison

- Nouvelle attaque applicative
  - SYN
  - SYN/ACK
  - ACK
  - ACK (00 01 + données aléatoires)
- Impact
  - Dès la première session

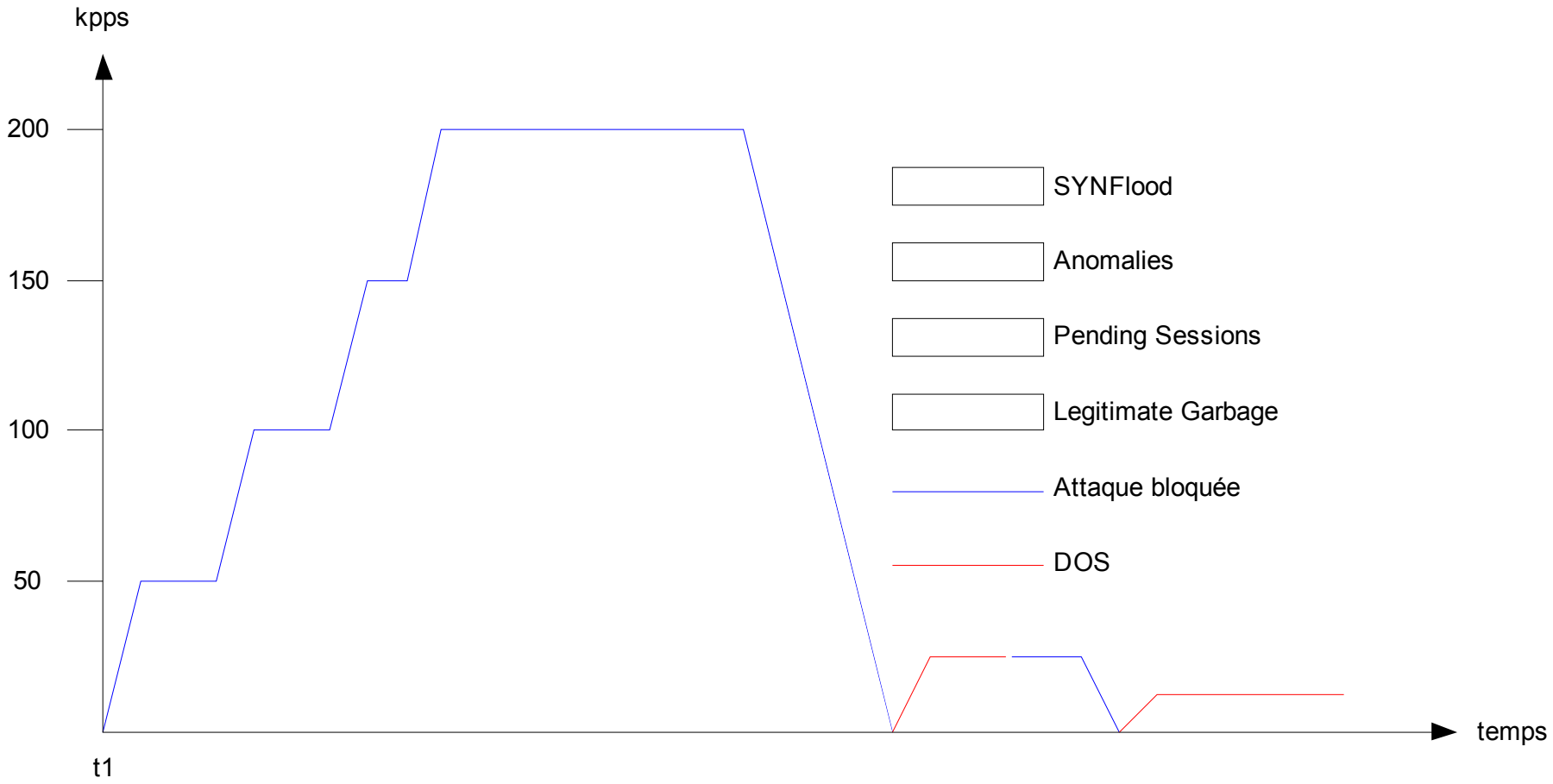
CRASH DE L'APPLICATION

Phase II | La dernière chance

- Comprendre l'application
  - Pour bloquer le trafic au contenu illégitime
  - A défaut de la redévelopper
- Le problème
  - Paranoïa de la cible
    - Ne veut fournir aucune information
  - Ne comprend plus son application
    - Développée en Sibérie par des prisonniers politiques
    - Ne veut pas y rémédier

Analyse Phase II

Phase II | Schéma de l'attaque



Phase II | Bad / Pending Sessions

- Principe
  - Ouverture de sessions légitimes sans fermeture
  - Atteint les limites de connexions
    - Imposées par le serveur
    - Possibles en fonction des ressources
  - SYNFlood de niveau 7
- Dans le cas étudié
  - Le premier paquet de données de répond pas aux critères de l'application
  - Celle-ci attend l'arrivée d'un paquet "correct" et considère la session toujours ouverte

Phase II | Bad / Pending Sessions

- Protection
  - Limitation du nombre de sessions par source
    - Egalement efficace contre les attaques de botnets effectuant des milliers de connexions valides et légitimes
      - Dans ce cas permet de protéger le lien montant au niveau du site final
    - ➢ Attention aux méta-proxy !
  - Fonctionnement “à la SYNCookies”
    - Nécessite un réel Stateful de niveau 7
    - Le moteur de Stateful doit être hautement configurable

Phase II | L'attaque finale

- Crash de l'application
  - Les données erronées envoyées à l'application on été traités
  - Elles ont à priori provoqué un crash
    - Buffer Overflow ?
  - Aucun accès à l'application ni au système
    - Pas de code
    - Pas de dumps
    - ➢ Rien ne peut être fait à ce niveau pour protéger l'application

Phase II | La “solution”

- Identification des adresses sources
  - Les sessions TCP ont révélé les sources
    - ~500 sources identifiées
  - Logs + scripts
    - = Filtrage des sources sur le firewall
- Mauvaise solution
  - Trop d’ACL sur le firewall
    - Groupes d’adresses comprenant des adresses non-compromises
    - Impact de performances
  - Solution temporaire
    - Prochaine attaque à partir d’un nouveau BotNet sera de nouveau efficace

Conclusion

Conclusion | Analyse de la source

- Les Botnets
  - A priori 4 : Montée en puissance des attaques par pallier
  - Les agents sont :
    - Soit des relais de commandes
      - L'ensemble des attaques auraient pu être lancées par hping3, uploadé sur les systèmes
    - Soit une application spécifique codée par l'attaquant
    - Dans tous les cas l'attaque était réfléchie
- L'attaquant
  - Connaissait le fonctionnement de l'application
  - A essayé d'autres attaques avant pour ne pas révéler cette information

Conclusion | Nous avons été chanceux

- Application restreinte
  - Nombre de clients limité
    - Peu de risque de blacklister les clients légitimes en prenant des tranches d'adresses IP larges
  - L'attaquant n'a pas insisté
- Reculer pour mieux sauter
  - « Security by obscurity »
  - OK
    - quand l'application est bien programmée
    - Jusqu'à un certain point
- *Security is a process, not a product*
  - Un produit ne peut pas tout faire à lui tout seul!

A propos de la RSTACK