Projet client

Cesi

2021

Documentation sur la partie cybersecurité

Differents types d'attaques

Injections SQL

SQL (Structured Query Language) est un langage qui nous permet d'interagir avec des bases de données . Les applications Web modernes utilisent des bases de données pour gérer les données et afficher un contenu dynamique aux lecteurs. Comme il s'agit d'une attaque courante, essayons d'en apprendre davantage sur ce que c'est, comment cela se produit et comment s'en défendre. Prêts ? Plongeons dedans !

L'injection SQL, ou SQLi, est un type d'attaque sur une application web qui permet à un attaquant d'insérer des instructions SQL malveillantes dans l'application web, pouvant potentiellement accéder à des données sensibles dans la base de données ou détruire ces données.

Il existe plusieurs types d'injection SQL :

- la méthode blind based (associée à sa cousine la time based), qui permet de détourner la requête SQL en cours sur le système et d'injecter des morceaux qui vont retourner caractère par caractère ce que l'attaquant cherche à extraire de la base de données. La méthode blind based, ainsi que la time based, se basent sur la réponse du serveur : si la requête d'origine renvoie bien le même résultat qu'à l'origine (et indique donc que le caractère est valide) ou ne renvoie pas le même résultat (et indique donc que le caractère testé n'est pas le bon). La time based a pour seule différence qu'elle se base sur le temps de réponse du serveur plutôt que sur la réponse en elle-même ;
- la méthode error based, qui permet de détourner la requête SQL en cours sur le système et d'injecter des morceaux qui vont retourner champ par champ ce que l'on cherche à extraire de la base de données. Cette méthode profite d'une faiblesse des systèmes de base de données permettant de détourner un message d'erreur généré par le système de base de données et préalablement volontairement provoquée par l'injection SQL pour lui faire retourner une valeur précise récupérée en base de données ;
- la méthode union based, qui permet de détourner la requête SQL en cours sur le système et d'injecter des morceaux qui vont retourner un ensemble de données directement extraites de la base de données. Cette méthode profite de certaines méthodes afin de détourner entièrement le retour de la requête SQL d'origine afin de lui faire retourner en une seule requête un important volume de données, directement récupéré en base de données. Dans ses exemples les plus violents, il est possible de récupérer des tables entières de base de données en une ou deux requêtes, même si en général cette méthode retourne entre 10 et 100 lignes de la base de données par requête SQL détournée ;
- la méthode Stacked queries, la plus dangereuse de toutes. Profitant d'une erreur de configuration du serveur de base de données, cette méthode permet d'exécuter n'importe quelle requête SQL sur le système ciblé, ce qui ne se limite pas seulement à récupérer des données comme les 3 précédentes. En effet, quand ce type de requête n'est pas désactivé, il suffit d'injecter une autre requête SQL, et elle sera exécutée sans problème, qu'elle aille chercher des données, ou en modifier directement dans la base de données.

Exemple :

Considérons un site web dynamique (programmé en PHP dans cet exemple) qui dispose d'un système permettant aux utilisateurs possédant un nom d'utilisateur et un mot de passe valides de se connecter. Ce site utilise la requête SQL suivante pour identifier un utilisateur :

```
SELECT uid FROM Users WHERE name = '(nom)' AND password = '(mot de passe hashé)';
```

L'utilisateur Dupont souhaite se connecter avec son mot de passe « truc » hashé en MD5. La requête suivante est exécutée :

```
SELECT uid FROM Users WHERE name = 'Dupont' AND password = '45723a2af3788c4ff17f8d1114760e62';
```

ATTAQUER LA REQUETE

Imaginons à présent que le script PHP exécutant cette requête ne vérifie pas les données entrantes pour garantir sa sécurité. Un hacker pourrait alors fournir les informations suivantes :

- Utilisateur : Dupont';--
- Mot de passe : n'importe lequel

La requête devient :

```
SELECT uid FROM Users WHERE name = 'Dupont';--' AND password = '4e383a1918b432a9bb7702f086c56596e';
```

Les caractères -- marquent le début d'un commentaire en SQL. La requête est donc équivalente à :

```
SELECT uid FROM Users WHERE name = 'Dupont';
```

L'attaquant peut alors se connecter sous l'utilisateur Dupont avec n'importe quel mot de passe. Il s'agit d'une injection de SQL réussie, car l'attaquant est parvenu à injecter les caractères qu'il voulait pour modifier le comportement de la requête.

Supposons maintenant que l'attaquant veuille non pas tromper le script SQL sur le nom d'utilisateur, mais sur le mot de passe. Il pourra alors injecter le code suivant :

Utilisateur : Dupont

Mot de passe : ' or 1 --

L'apostrophe indique la fin de la zone de frappe de l'utilisateur, le code « or 1 » demande au script si 1 est vrai, or c'est toujours le cas, et -- indique le début d'un commentaire.

La requête devient alors :

```
SELECT uid FROM Users WHERE name = 'Dupont' AND password = '' or 1 --';
```

Ainsi, le script programmé pour vérifier si ce que l'utilisateur tape est vrai, il verra que 1 est vrai, et l'attaquant sera connecté sous la session Dupont.

SOLUTION

La première solution consiste à échapper les caractères spéciaux contenus dans les chaînes de caractères entrées par l'utilisateur.

En PHP on peut utiliser pour cela la fonction mysqli_real_escape_string, qui transformera la chaîne ' -- en \' --. La requête deviendrait alors :

```
SELECT uid FROM Users WHERE name = 'Dupont\' -- ' AND password =
'4e383a1918b432a9bb7702f086c56596e';
```

L'apostrophe de fin de chaîne ayant été correctement dé-spécialisée en la faisant précéder d'un caractère « \ ».

L'échappement peut aussi se faire (suivant le SGBD utilisé) en doublant les apostrophes.

La marque de commentaire fera alors partie de la chaîne, et finalement le serveur SQL répondra qu'il n'y a aucune entrée dans la base de données correspondant à un utilisateur Dupont' -- avec ce mot de passe.

La fonction addslashes ne suffit pas pour empêcher les injections via les variables numériques, qui ne sont pas encadrées d'apostrophes ou de guillemets dans les requêtes SQL. Exemple avec la requête :

```
SELECT ... FROM ... WHERE numero_badge = $numero AND code_4_chiffre = $code;
```

qui réussit lorsque la variable $numero contient 0 or 1 --. Une précaution est d'utiliser la fonction ctype_digit pour vérifier les variables numériques des requêtes. On peut aussi forcer la transformation de la variable en nombre en la faisant précéder d'un transtypeur, comme (int) si on attend un entier (la chaîne 0 or 1 -- sera alors transformée en l'entier 0 et l'injection SQL échouera).

La fonction addslashes possède elle-même quelques failles sur certaines versions de PHP qui datent. De plus, elle échappe uniquement les caractères « \ », « NULL », « ' » et « " ». Il serait plus approprié d'utiliser la fonction mysqli_real_escape_string qui échappe justement les caractères spéciaux d'une commande SQL (NULL, \x1a, \n , \r , \, ', " et \x00).

PHISHING:

Le Phishing est une technique utilisée par des fraudeurs pour obtenir des renseignements personnels dans le but de perpétrer une usurpation d'identité. La technique consiste à faire croire à la victime qu'elle s'adresse à un tiers de confiance — banque, administration, etc. — afin de lui soutirer des renseignements personnels : mot de passe, numéro de carte de crédit, numéro ou photocopie de la carte d'identité, date de naissance, etc. En effet, le plus souvent, une copie exacte d'un site internet est réalisée dans l'optique de faire croire à la victime qu'elle se trouve sur le site internet officiel où elle pensait se connecter. La victime va ainsi saisir ses codes personnels qui seront récupérés par celui qui a créé le faux site, il aura ainsi accès aux données personnelles de la victime et pourra dérober tout ce que la victime possède sur ce site. L'attaque peut aussi être réalisée par courrier électronique ou autres moyens électroniques. Lorsque cette technique utilise les SMS pour obtenir des renseignements personnels, elle s'appelle SMiShing.

Techniques

Vérification de l'orthographe du nom de domaine

La vérification de l'adresse web dans la barre d'adresse du navigateur web est la première parade. Ainsi, une attaque simple consiste à utiliser un nom de domaine très semblable (par exemple avec une faute grammaticale ou orthographique), comme « http://www.societegeneral.fr » au lieu de « http://www.societegenerale.fr ». L'attaquant aura préalablement acheté un nom de domaine proche de l'original, généralement une variante orthographique.

De même le site « france-impotsgouv.fr », a pu être utilisé pour usurper « impots.gouv.fr ».

Vérification de l'absence d'arobase dans l'URL

Dans les années 1990 et au début des années 2000, une méthode très utilisée était la possibilité de laisser au sein de l'URL le nom d'utilisateur et le mot de passe dans le cadre d'une authentification HTTP. L'URL prend alors la forme « http://login:motdepasse@www.domaine.tld ».

À cette époque, il était fréquent que les URLs comportent une longue chaîne de caractère pour identifier la session de l'utilisateur. Par exemple, une telle URL pouvait ressembler à « http://www.domaine.tld/my.cnf?id=56452575711&res=lorem-ipsum-dolor&quux=2&lang=fr&sessid=jP3ie3qjSebbZRsC0c9dpcLVe2cAh0sCza3jcX7mSuRzwY4N0v1DBB71DMKNkbS »

Les attaquants concevaient dès lors une URL ressemblant à celle ci-dessus, en écrivant le nom de domaine usurpé comme login. Par exemple, pour convaincre l'utilisateur que le site qu'il visite est bien www.societegenerale.fr, et que l'adresse IP du serveur de l'attaquant est 88.132.11.17, l'URL pouvait être « http://www.societegenerale.fr/espaceclient:id=56452575711&res=lorem-ipsum-

dolor&quux=2&lang=frsessid=jP3ie3qjSebbZRsC0c9dpcLVe2cAh0sCza3jcX7mSuRzwY4N0v1DBB71DMKNkbS@88.132.11.17 ».

Du fait de cette technique d'hameçonnage, les navigateurs web ont été améliorés afin de prévenir leurs utilisateurs lorsqu'ils détectent cette manœuvre. Ainsi, dans le cas précédent, le navigateur Firefox proposerait le message suivant :

> Vous êtes sur le point de vous connecter au site « 88.132.11.17 » avec le nom d'utilisateur « a », mais ce site web ne nécessite pas d'authentification. Il peut s'agir d'une tentative pour vous induire en erreur.
> « 88.132.11.17 » est-il bien le site que vous voulez visiter ?

Cette technique d'hameçonnage est donc aujourd'hui minoritaire.

Vérifier l'absence de caractères Unicode

Une méthode plus élaborée pour masquer le nom de domaine réel consiste à utiliser des caractères bien choisis parmi les dizaines de milliers de caractères du répertoire Unicode. En effet, certains caractères spéciaux ont l'apparence des caractères de l'alphabet latin. Ainsi, l'adresse web « http://www.paypal.com/ » a la même apparence que « http://www.paypal.com/ », mais est pourtant bien différente car elle renvoie vers un site web différent. De même, http://www.airfrạnce.com aura la même apparence que « http://www.airfrance.com », surtout si le navigateur souligne l'adresse internet.

Une contre-mesure à cette attaque est de ne pas permettre l'affichage des caractères hors du répertoire ASCII, qui ne contient que les lettres de A à Z, les chiffres et de la ponctuation. Cette dernière contre-mesure est cependant difficilement compatible avec l'internationalisation des noms de domaine, qui requiert le jeu de caractères Unicode.

Vérifier les certificats électroniques

Il existe depuis les années 1990 une parade technique à l'hameçonnage : le certificat électronique. Toutefois, l'interface utilisateur des navigateurs Web a longtemps rendu les certificats incompréhensibles pour les visiteurs. Cette interface était connue sous les traits d'un petit cadenas. Il était simplement expliqué au grand public que le cadenas signifie que la communication est chiffrée, ce qui est vrai, mais ne protège aucunement contre l'hameçonnage. Dans les années 2000, des certificats étendus ont été inventés. Ils permettent d'afficher plus clairement l'identité vérifiée d'un site.

Écrire manuellement les URL

Une personne contactée au sujet d'un compte devant être « vérifié » doit chercher à régler le problème directement avec la société concernée ou se rendre sur le site web en tapant manuellement l'adresse dans la barre d'adresse dans son navigateur web plutôt qu'en cliquant sur un lien qui lui aurait été fourni. Il faut savoir que les sociétés bancaires n'utilisent jamais la communication par courriel pour corriger un problème de sécurité avec leurs clients. En règle générale, il est recommandé de faire suivre le message suspect à la société concernée, ce qui lui permettra de faire une enquête.

Autres techniques

Les filtres anti-spam aident à protéger l'utilisateur des criminels informatiques par le fait qu'ils réduisent le nombre de courriels que les utilisateurs reçoivent et par conséquent les risques d'hameçonnage. Le logiciel client de messagerie Mozilla Thunderbird comporte un filtre bayésien très performant (c'est un filtre anti-spam auto-adaptatif).

Les fraudes concernant les banques en ligne visent à obtenir l'identifiant et le mot de passe du titulaire d'un compte. Il devient ensuite possible au fraudeur de se connecter sur le site web de la banque et d'effectuer des virements de fonds vers son propre compte. Pour parer à ce type de fraude, la plupart des sites bancaires en ligne n'autorisent plus l'internaute à saisir lui-même le compte destinataire du virement : il faut, en règle générale, téléphoner à un service de la banque qui reste seul habilité à saisir le compte destinataire dans une liste de comptes. La conversation téléphonique est souvent enregistrée et peut alors servir de preuve.

D'autres banques utilisent une identification renforcée, qui verrouille l'accès aux virements si l'utilisateur n'indique pas la bonne clé à huit chiffres demandée aléatoirement, parmi les soixante-quatre qu'il possède. Si la clé est la bonne, l'internaute peut effectuer des virements en ligne..

Attaque par DoS & DDoS:

Une attaque par déni de service (abr. DoS attack pour Denial of Service attack en anglais) est une attaque informatique ayant pour but de rendre indisponible un service, d'empêcher les utilisateurs légitimes d'un service de l'utiliser. À l'heure actuelle la grande majorité de ces attaques se font à partir de plusieurs sources, on parle alors d'attaque par déni de service distribuée (abr. DDoS attack pour Distributed Denial of Service attack).

Il peut s'agir de :

l'inondation d'un réseau afin d'empêcher son fonctionnement ;

la perturbation des connexions entre deux machines, empêchant l'accès à un service particulier ;

l'obstruction d'accès à un service pour une personne en particulier ;

également le fait d'envoyer des milliards d'octets à une box internet.

L'attaque par déni de service peut ainsi bloquer un serveur de fichiers, rendre impossible l'accès à un serveur web ou empêcher la distribution de courriel dans une entreprise.

L'attaquant n'a pas forcément besoin de matériel sophistiqué. Ainsi, certaines attaques DoS peuvent être exécutées avec des ressources limitées contre un réseau de taille plus importante et plus moderne. On appelle parfois ce type d'attaque « attaque asymétrique » en raison de la différence de ressources entre les protagonistes.

Les attaques en déni de service se sont modifiées au cours du temps (voir historique).

Tout d'abord, les premières n'étaient perpétrées que par un seul « attaquant » ; rapidement, des attaques plus évoluées sont apparues, impliquant une multitude de « soldats ». On parle alors de DDoS (distributed denial of service attack). Certains pirates informatiques se sont spécialisés dans la « levée » d'armées de « zombies », qu'ils peuvent ensuite louer à d'autres personnes ou groupes malveillants pour attaquer une cible particulière. Avec la forte augmentation du nombre d'échanges commerciaux sur Internet, le nombre de chantages au déni de service a très fortement progressé

Type d'attaques:

On appelle « attaque par déni de service » toutes les actions ayant pour résultat la mise hors ligne d'un serveur. Techniquement, couper la connexion entre un serveur et un client, pour but malfaisant, peut être considéré comme une attaque par déni de service. Dans les faits, les attaques par déni de service sont opérées en saturant la bande passante d'un serveur défini.

Exploitation des failles ou des limites des machines

Une des attaques les plus courantes consistait à envoyer un paquet ICMP de plus de 65 535 octets. Au-dessus de cette limite, les piles IP ne savaient pas gérer le paquet proprement, ce qui entraînait des erreurs de fragmentation UDP, ou encore les paquets TCP contenant des « flags » illégaux ou incompatibles.

Les piles actuelles résistent à ce type d'attaques. Néanmoins, les délais de traitement de ce genre de paquets restent plus longs que ceux nécessaires pour traiter les paquets légitimes. Ainsi, il devient commun voire trivial de générer une consommation excessive de processeur (CPU) par la simple émission de plusieurs centaines de milliers d'anomalies par seconde, ce qu'un outil tel que hping3 permet en une unique ligne de commande…

ex. : [root@localhost root]# hping3 -SARFU -L 0 -M 0 -p. 80 www.cible.com—flood

Avec l'arrivée du haut débit et l'augmentation de la puissance des ordinateurs personnels, le potentiel d'attaque a été décuplé, mettant en évidence la faiblesse des installations développées il y a plusieurs années. Cette augmentation permet à quasiment toutes les anomalies d'être à l'origine d'un déni de service, pourvu qu'elles soient générées à un rythme suffisamment important.

Par exemple :

- l'usage des champs « réservés » de l'en-tête TCP ;
- le positionnement d'un numéro de séquence d'accusé de réception dans un paquet SYN ;
- des paquets dont l'en-tête de couche 4 (TCP/UDP) est tronqué en dépit de checksums corrects.

Attaque par déni de service SYN Flood

Une **attaque SYN Flood** est une attaque visant à provoquer un déni de service en émettant un nombre important de demandes de synchronisation TCP incomplète avec un serveur.

Quand un système (client) tente d'établir une connexion TCP vers un système offrant un service (serveur), le client et le serveur échangent une séquence de messages.

Le système client commence par envoyer un message SYN au serveur. Le serveur reconnaît ensuite le message en envoyant un SYN-ACK message au client. Le client finit alors d'établir la connexion en répondant par un message ACK. La connexion entre le client et le serveur est alors ouverte, et le service de données spécifiques peut être échangé entre le client et le serveur. Voici une vue de ce flux de messages :

Client Serveur

------ -------

SYN --------- →

← --------- SYN-ACK

ACK --------- →

Le risque d'abus se pose à l'endroit où le système de serveur a envoyé un accusé de réception (SYN-ACK) au client, mais ne reçoit pas le message ACK. Le serveur construit dans sa mémoire système une structure de données décrivant toutes les connexions. Cette structure de données est de taille finie, et elle peut être débordée en créant intentionnellement trop de connexions partiellement ouvertes.

Créer des connexions semi-ouvertes s'accomplit facilement avec l'IP spoofing. Le système de l'agresseur envoie des messages SYN à la machine victime ; ceux-ci semblent être légitimes, mais font référence à un système client incapable de répondre au message SYN-ACK. Cela signifie que le message ACK final ne sera jamais envoyé au serveur victime.

Normalement il y a un délai d'attente associé à une connexion entrante, les semi-connexions ouvertes vont expirer et le serveur victime pourra gérer l'attaque. Toutefois, le système agresseur peut simplement continuer à envoyer des paquets IP falsifiés demandant de nouvelles connexions, plus rapides que le serveur victime.

Dans la plupart des cas, la victime aura des difficultés à accepter toute nouvelle connexion réseau entrante. Dans ces cas, l'attaque n'affecte pas les connexions entrantes, ni la possibilité d'établir des connexions réseau sortant. Toutefois, le système peut saturer la mémoire, ce qui provoque un crash rendant le système inopérant.

UDP Flooding

Ce déni de service exploite le mode non connecté du protocole UDP. Il crée une UDP Packet Storm (génération d'une grande quantité de paquets UDP) soit à destination d'une machine soit entre deux machines. Une telle attaque entre deux machines entraîne une congestion du réseau ainsi qu'une saturation des ressources des deux hôtes victimes. La congestion est plus importante du fait que le trafic UDP est prioritaire sur le trafic TCP. En effet, le protocole TCP possède un mécanisme de contrôle de congestion, dans le cas où l'acquittement d'un paquet arrive après un long délai, ce mécanisme adapte la

fréquence d'émission des paquets TCP et le débit diminue. Le protocole UDP ne possède pas ce mécanisme. Au bout d'un certain temps, le trafic UDP occupe donc toute la bande passante, ne laissant qu'une infime partie au trafic TCP.

L'exemple le plus connu d'UDP Flooding est la « Chargen Denial of Service Attack ». La mise en pratique de cette attaque est simple, il suffit de faire communiquer le service chargen d'une machine avec le service echo d'une autre. Le premier génère des caractères, tandis que le second se contente de réémettre les données qu'il reçoit. Il suffit alors à l'attaquant d'envoyer des paquets UDP sur le port 19 (chargen) à une des victimes en usurpant l'adresse IP et le port source de l'autre. Dans ce cas, le port source est le port UDP 7 (echo). L'UDP Flooding entraîne une saturation de la bande passante entre les deux machines, il peut donc neutraliser complètement un réseau.

Packet Fragment

Les dénis de service de type Packet Fragment utilisent des faiblesses dans l'implémentation de certaines piles TCP/IP au niveau de la défragmentation IP (ré-assemblage des fragments IP).

Une attaque connue utilisant ce principe est Teardrop. L'offset de fragmentation du second fragment est inférieur à la taille du premier ainsi que l'offset plus la taille du second. Cela revient à dire que le deuxième fragment est contenu dans le premier (overlapping). Lors de la défragmentation, certains systèmes ne gèrent pas cette exception et cela entraîne un déni de service. Il existe des variantes de cette attaque : bonk, boink et newtear. Le déni de service Ping of Death exploite une mauvaise gestion de la défragmentation au niveau ICMP, en envoyant une quantité de données supérieure à la taille maximum d'un paquet IP. Ces différents dénis de services aboutissent à un crash de la machine cible.

Smurfing

Cette attaque utilise le protocole ICMP. Quand un ping (message ICMP ECHO) est envoyé à une adresse de broadcast, celui-ci est démultiplié et envoyé à chacune des machines du réseau. Le principe de l'attaque est de truquer les paquets ICMP ECHO REQUEST envoyés en mettant comme adresse IP source celle de la cible. L'attaquant envoie un flux continu de ping vers l'adresse de broadcast d'un réseau et toutes les machines répondent alors par un message ICMP ECHO REPLY en direction de la cible. Le flux est alors multiplié par le nombre d'hôtes composant le réseau. Dans ce cas tout le réseau cible subit le déni de service, car l'énorme quantité de trafic générée par cette attaque entraîne une congestion du réseau.

Empoisonnement de l'adresse MAC du routeur de sortie

Lors d'une requête ARP, l'attaquant peut répondre en associant la route de sortie à une adresse MAC inexistante. De cette manière, les paquets envoyés sont alors perdus dans le réseau. Et le réseau perd son accès à internet.

Exemples de mode d'attaque

- Ping 'O Death (ping de la mort) : il s'agit de saturer un routeur ou un serveur en envoyant un nombre important de requêtes ICMP REQUEST dont les datagrammes dépassent la taille maximum autorisée. Des patches existent afin de se prémunir de ce type d'agression sous les systèmes MacOs, Windows NT/9x, Sun [Oracle] Solaris, Linux et Novell Netware.

- Land - Blat : il s'agit d'envoyer un paquet forgé (« spoofé ») contenant le flag SYN sur un port donné (comme 113 ou 139 par exemple) et de définir la source comme étant l'adresse de la station cible. Il existe un certain nombre de patches pour ce « bug » pour les systèmes UNIX et Windows.

- Jolt : spécialement destinée aux systèmes Microsoft (NT, 9x et 2000), cette attaque permet de saturer le processeur de la station qui la subit. La fragmentation IP provoque, lorsque l'on envoie un grand nombre de fragments de paquets identiques (150 par seconde), une saturation totale du processeur durant toute la durée de l'attaque. Des pré-patches existent déjà pour tenter de contrer ce type d'attaque.

- TearDrop - SynDrop : problème découvert dans l'ancien noyau du système Linux dans la partie concernant la fragmentation des paquets IP. Il s'agit d'un problème de reconstruction du paquet. Lorsque le système reconstitue le paquet, il exécute une boucle qui va permettre de stocker dans un nouveau *buffer* tous les paquets déjà reçus. Il y a effectivement un contrôle de la taille du paquet mais uniquement si ce dernier est trop grand. S'il est trop petit cela peut provoquer un problème au niveau du noyau et planter le système (problème d'alignement des paquets). Ce problème a également été observé sur les systèmes Windows (NT/9x) et des patches sont dès à présent disponibles.

- Ident Attack : ce problème dans le daemon identd permet aisément de déstabiliser une machine UNIX qui l'utilise. Un grand nombre de requêtes d'autorisation entraîne une instabilité totale de la machine. Pour éviter cela, il faut installer une version plus récente du daemon identd ou alors utiliser le daemon pidentd-2.8a4 (ou ultérieur).

- Bonk - Boink : même problème que le *TearDrop* mais légèrement modifié afin de ne pas être affecté par les patches fournis pour le TearDrop. Il existe de nouveaux patches mieux construits qui permettent également d'éviter ce nouveau type d'attaque.

- Smurf : ce programme utilise la technique de l'*ICMP Flood* et l'amplifie de manière à créer un véritable désastre sur la (ou les) machines visées. En fait, il utilise la technique du *Broadcast Ping* afin que le nombre de paquets ICMP envoyés à la station grandisse de manière exponentielle causant alors un crash presque inévitable. Il est difficile de se protéger de ce type d'attaque, il n'existe aucun patch mais des règles de filtrage correctes permettent de limiter son effet.

- WinNuke : il s'agit d'un programme permettant de « *crasher* » les systèmes Windows NT/95 par l'envoi de données de type « OOB » (*Out of Band*) lors d'une connexion avec un client Windows. NetBIOS semble être le service le plus vulnérable à ce type d'attaque. Apparemment, Windows ne sait comment réagir à la réception de ce type de paquet et « panique ». De nombreux patches existent contre ce type d'attaque et les versions postérieures de Windows (à partir de Windows 98/2000) sont dès à présent protégées.