FRAMEWORK W3AF

Régis SENET

W3af ou bien encore Web Application Attack and Audit Framework est, comme son nom l'indique, un framework permettant d'automatiser l'audit ainsi que les attaques à l'encontre des applications web.

Cet article explique...
- L'utilisation de w3af.
- La récupération et l'utilisation d'information.

Ce qu'il faut savoir...
- Les bases des sites web
- Les bases des attaques Web (Injection SQL / Injection de code / Inclusion de fichier)

Depuis l'augmentation de l'importance d'Internet dans la vie quotidienne de nombreuses personnes, la sécurité des sites web reste plus que jamais une inquiétude majeure. W3AF ou Web Application Attack and Audit Framework permet d'automatiser les attaques et les audits à l'encontre des sites Internet afin de vous prémunir contre les diverses attaques possibles par des individus malintentionnés.

Pourquoi protéger vos applications web ?

La sécurité des sites Internet est aujourd'hui l'un des aspects de la sécurité en entreprise le plus souvent négligé alors qu'il devrait être une priorité dans n'importe quelle organisation. De plus en plus, les pirates informatiques concentrent leurs efforts sur les applications web afin d'obtenir une approche des informations confidentielles et abuser des données sensibles comme les détails de client, les numéros de carte de crédit et autre. Les applications web réalisant des achats en ligne, des authentifications d'utilisateurs ou utilisant simplement tous types de contenu dynamique permettent à l'utilisateur d'interagir avec des données présents dans une base de données. Sur certaines applications, ces données peuvent être personnelles voir sensibles. Si ces applications web ne sont pas sécurisées, votre base de données entière de renseignements sensibles court un risque réel.

Figure 1. *Live CD Samurai*

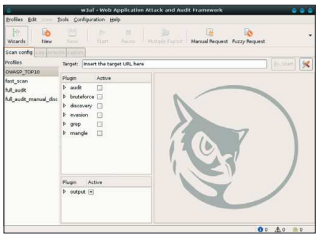

Figure 2. *W3AF via l'interface graphique*

Comme tous les systèmes informatiques, une application web doit répondre à trois caractéristiques :

- Confidentialité
- Disponibilité
- Intégrité

La sécurisation des réseaux ainsi que l'installation d'un pare-feu ne fournit aucune protection contre les attaques web car celles-ci sont lancées sur le port 80 (le port par défaut pour les sites Internet) qui doit rester ouvert. Pour la stratégie de sécurité la plus complète, il est donc urgent que vous auditiez régulièrement vos applications web pour vérifier la présence de vulnérabilités exploitables.

Pourquoi s'attaquer à une application web ?

Les failles web permettent des actions de plus en plus importantes de la part des pirates informatiques. Il est fini le temps où le piratage d'un site Web s'est contenté d'afficher une simple fenêtre sur la page de l'utilisateur ou bien le vol d'un cookie.

De nos jours, le piratage d'une application Web est nettement plus dangereux que cela :

- Défaçage complet ou partiel d'un site Internet.
- Accès aux données sensibles des utilisateurs.

Il est bel et bien temps d'inclure les sites web dans la politique de sécurité des entreprises et ceci de manière draconienne. Pour faire, nous allons maintenant vous présenter w3af.

Qu'est ce que w3af ?

W3af ou bien encore Web Application Attack and Audit Framework est, comme son nom l'indique, un framework permettant d'automatiser l'audit ainsi que les attaques à l'encontre des applications web. Pour ceux d'entre vous connaissant Métasploit, w3af peut être comparé à ce dernier en matière de pen-test sur les applications web.

W3af est un framework très complet placé sous licence GPL (General Public License) entièrement écrit en Python avec un code extrêmement bien commenté permettant ainsi à n'importe quel développeur potentiel de créer ses propres modules/exploits.

Grossièrement, w3af peut se décomposer en trois catégories :

- Découverte
- Audit
- Attaque

Les plugins de « découverte » ont pour but de rechercher des formulaires, des urls ou plus généralement tout point potentiel d'injection de code malveillant. Un exemple classique de plugin de découverte est web spider. Ce plugin prend une URL en entrée et retourne un ou plusieurs points d'injection.

Les plugins d' « audit » attendent les points d'injection découverts par les plugins de découverte et envoient des données construites spécifiquement à tous ces derniers afin de trouver des vulnérabilités. Un exemple classique de plugin audit est un plugin qui recherche des vulnérabilités d'injection SQL.

Les plugins d' « attaque » ont pour but d'exploiter les vulnérabilités trouvées par les plugins d'audit et de découverte. Ils retournent en général un Shell sur le serveur distant, ou un dump des tables distantes dans le cas des exploits d'injections SQL.

Origine du projet

Le projet w3af à vu le jour au début de l'année 2007 grâce aux travaux de son unique développeur Andrés Riancho. Andrés est un chercheur connu et reconnu dans le monde de la sécurité informatique notamment dans le domaine des applications web. W3af est actuellement à sa version 1.0-rc1

Son principal objectif est de rendre le web le plus sécuritaire possible vu les enjeux qui sont maintenant en train de transiter dessus.

Le framework w3af est très portable et peut s'utiliser sur n'importe quelle plateforme tant que cette dernière supporte le Python (Linux, WinXP, Vista, OpenBSD, etc.)

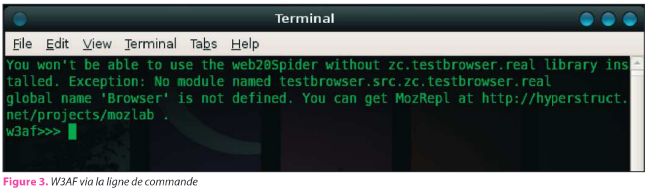

Figure 3. *W3AF via la ligne de commande*

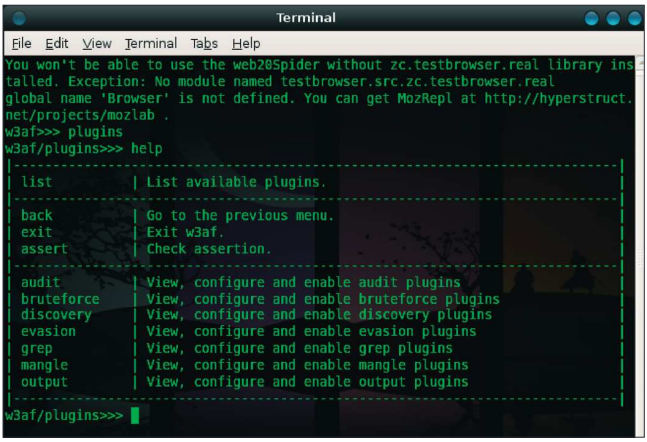

Figure 4. *Liste des options disponibles en ligne de commande*

A partir du moment où l'environnement Python est présent sur la machine accueillant w3af, il existe trois moyens afin de s'en servir :

- Téléchargement et installation des paquets (Solution sous linux)
- Téléchargement et exécution des binaires (Solution sous Windows)
- Utilisation de w3af contenu dans le LiveCD Samurai (cf. Figure 1)

Au cours de cet article, nous allons donc utiliser le LiveCD Samurai Web Testing Framework afin de pouvoir utiliser w3af dans les meilleures conditions possibles.

Figure 5. *Site internet cible*

Pour simple information, Samurai Web Testing Framework est un LiveCD préconfiguré pour les tests de pénétration des sites web. Ce LiveCD contient les meilleurs outils de cette catégorie qu'ils soient Open Source ou bien gratuits. Ce LiveCD est disponible à l'adresse *http://samurai.inguardians.com/*

Objectif

Avec la réussite que rencontre w3af dans les tests de pénétration web, Andrés Riancho a pour objectif de faire de w3af le meilleur scanner d'application web Open Source ainsi que le meilleur framework d'exploitation des failles pour les applications web. Pour reprendre

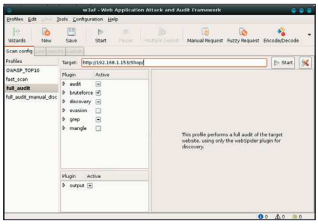

Figure 6. *Lancement de l'attaque*

ces propres propos, il voudrait que w3af devienne le nmap du web, c'est-à-dire l'outil totalement incontournable.

W3AF et ses possibilités

Avant de rentrer dans les détails techniques que propose w3af, il est important de préciser qu'il est possible d'utiliser w3af de deux manières différentes :

- Via son interface graphique (cf Figure 2)
- Via sa ligne de commande (cf Figure 3)

L'interface graphique de w3af est une interface graphique particulièrement soignée et simple d'utilisation basée sur la librairie GTK. L'utilisation de l'interface graphique n'est en aucun cas restrictive du fait qu'elle permet d'utiliser w3af à 100% de ces capacités.

Il est également possible d'utiliser la ligne de commande pour se servir de w3af. Il est possible, via la ligne de commande, d'exécuter exactement les mêmes commandes que grâce à l'interface graphique.

D'un coté un peu plus technique, w3af se divise en deux parties : le core gérant l'ensemble des processus ainsi que la communication entre les plugins et les plugins. Précédemment, nous avons dit que w3af pouvait grossièrement se décomposer en trois parties : découverte, audit et attaque.

Maintenant, nous allons tenter de découvrir w3af un peu plus en profondeur en dévoilant les 8 catégories distinctes que ce dernier possède :

- Découverte
- Audit
- Attaques
- Grep
- Modificateurs de requête
- Evasion
- Brute Force
- Affichage

Comme nous avons pu le dire précédemment, les plugins de « découverte » ont pour objectif de rechercher des points d'injection dans un site web (url, formulaire, page d'authentification).

Les plugins d' « audit » récupèrent les points d'injection trouvés précédemment par les plugins de découverte et tentent de trouver des vulnérabilités spécifiques à toutes les possibilités.

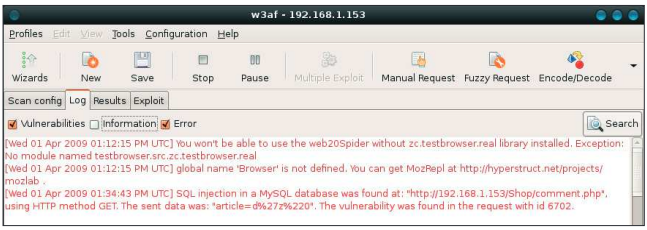

Figure 7. *Vérification des fichiers de log*

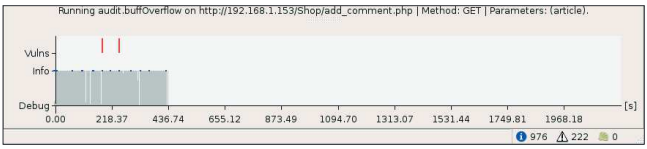

Figure 8. *Graphique des failles en temps réel*

Figure 9. *Choix du niveau d'affichage des logs*

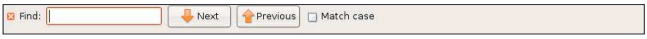

Figure 10. *Recherche dans les fichiers de log*

Les plugins d' « attaque » exploitent les vulnérabilités trouvées par les plugins d'audit. Ils retournent en général un Shell sur le serveur distant, ou un dump des tables distantes dans le cas des exploits d'injections SQL.

Les plugins de type « grep » analysent le contenu de l'ensemble des pages et tentent de trouver des vulnérabilités sur les pages interrogées. Certains plugins vont, par exemple, tenter de récupérer des commentaires dans les pages HTML possédant certains mots clé comme « password », « admin » etc.

Les plugins « modificateurs de requête » permettent, comme leurs noms l'indiquent, de modifier les requêtes ainsi que les réponses du serveur avant de les réacheminer. Il est important de comprendre que grâce à ce genre d'outils, les contrôles mis en place côté client par du JavaScript par exemple peuvent facilement être contournés comme la gestion des longueurs maximale d'un champ grâce à l'attribut « maxlength ».

Les plugins d' « évasion » tentent de contourner l'ensemble des règles mises en place par des IDS (Intrusion Detection System) ou IPS (Intrusion Prevention System) afin d'être le plus furtif possible.

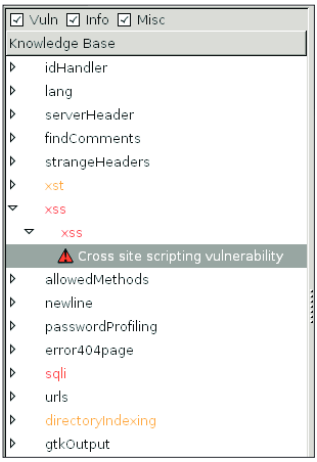

Figure 11. *Résultat de l'audit*

Les plugins de « bruteforce » permettent de réaliser des attaques par force brute contre les formulaires d'identifications par exemple.

Dernièrement, les plugins d' « affichage » quand à eux représentent la manière via laquelle les plugins vont communiquer avec l'utilisateur. Les plugins d'affichage enregistrent les données dans un fichier texte ou HTML.

La documentation officielle réalisée par Andres Riancho ainsi que sa version française traduite par Jérôme Athias (JA-PSI) couvre principalement l'utilisation de w3af en ligne de commande. Ces documentations sont disponibles à l'adresse suivante : *http://w3af.sourceforge.net/#documentation*

Pour ne pas reprendre leurs excellents travaux dans lequel quasiment tout est dit, nous allons présenter une attaque complète via l'interface graphique.

Avant de vous exposer un cas concret, nous allons faire une liste non exhaustive de quelques plugins rangés par catégories.

Audit

- SQL injection detection
- XSS detection
- SSI detection
- Local file include detection
- Remote file include detection
- Buffer Overflow detection
- OS Commanding detection
- Response Splitting detection

Découverte

- Pykto
- Hmap
- fingerGoogle
- googleSpider
- webSpider

Grep

- collectCookies
- directoryIndexing
- findComments
- pathDisclosure
- strangeHeaders

Affichage

- console
- htmlFile

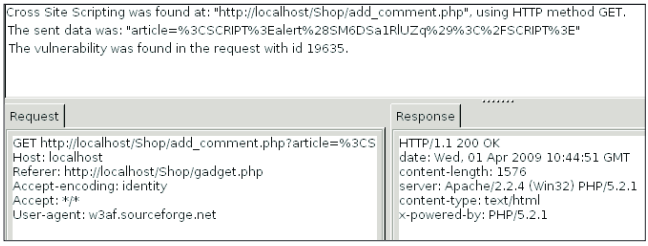

Figure 12. *Détail de la faille*

- textFile

Modificateur de requête

- sed, un éditeur de requête http

Evasion

- reversedSlashes
- rndCase
- rndHexEncode

Attaque

- davShell
- fileUploadShell
- googleProxy
- localFileReader
- mysqlWebShell

Nous allons à présent nous attaquer à un site présentant divers articles sur lesquels il est possible de faire des commentaires (cf Figure 5).

W3AF en pratique

Afin d'avoir les résultats les plus précis possibles, il est important de bien choisir les options ainsi que les plugins que nous voulons utiliser. Dans notre cas, nous allons utiliser des profils préconfigurés pour réaliser notre scan.

W3af dispose de quatre profils par défaut :

- OWASP Top 10 intégrant les plugins d'audit, de découverte et de grep
- Fast Scan intégrant les plugins d'audit et de découverte
- Full Audit intégrant les plugins d'audit, brute force, découverte et grep
- Full audit manual discintégrant les plugins d'audit, brute force, découverte et grep

Afin d'avoir un résultat clair et précis, nous allons commencer par un « Full Audit »

Il est donc simplement nécessaire de cliquer sur « Full Audit » dans le menu « Profiles », de préciser l'adresse

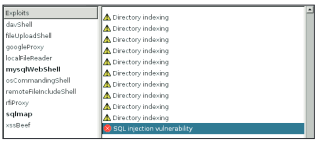

Figure 13. *Détail de la faille*

Figure 14. *Tentative d'exploitation d'une faille XSS*

Figure 15. *Exploitation d'une faille XSS*

du site web cible et de lancer l'analyse en cliquant sur « Start »

A présent, l'ensemble des plugins se trouvant dans les diverses catégories sélectionnées vont s'exécuter un à un pour avoir le plus de résultat possible.

Il est possible grâce à l'onglet « Log » de visualiser rapidement tout ce que w3af a trouvé comme informations, erreurs ou vulnérabilités (cf Figure 7).

Il est possible de voir que la page *comment.php* est vulnérable à des injections de type SQL. Cette page sera donc un page à étudier par la suite.

Dans la même fenêtre que celle précédemment vue, il est également possible d'avoir une interprétation graphique des analyses en cours comme il est possible de le voir sur la Figure 8. Ce graphique nous montre de nombreux détails comme par exemple le temps (en seconde) de l'analyse réalisée par w3af ou bien encore les modules qui sont exécutés en temps réel (audit.bufferOverflow au moment de la prise du screenshot)

Après quelques minutes, l'analyse du site cible se termine et il est enfin possible d'analyser les résultats que nous propose w3af. Grâce à la barre principale, nous sommes en mesure de faire un premier filtre très rapide mais tout aussi simpliste sur le type d'information qu'a relevé w3af. Il est possible de sélectionner « Vulnérabilités » et/ou « Information » et/ou « Erreur » (cf Figure 9).

Nous pouvons également faire des recherches plus précises grâce à la partie (cf Figure 10) permettant de rechercher des mots ou des bouts de mots dans l'ensemble des informations disponibles.

Une fois les logs analysés, il vous est possible d'allez dans l'onglet « Results » afin d'avoir de plus amples renseignements concernant les informations précédemment trouvées (cf Figure 11).

Via cette interface vous pouvez résumer l'ensemble des informations que w3af à réussi à trouver sur l'ensemble du site cible.

Il est également très facile grâce à cette interface d'identifier des vulnérabilités de type Injection XSS, injection SQL. Tout comme pour les logs, il est possible de filtrer les résultats en fonctions de leur type (Vulnérabilités, informations et problème de configuration) afin d'effectuer un rapide coup d'œil sur les problèmes potentiels.

Des informations plus précises sont présentes dans la partie de droite permettant de comprendre quelle page est vulnérable et à quel type d'attaque. Il nous est également possible de voir les requêtes envoyées au serveur ainsi que les réponses de ce dernier pour des possibles modifications via les plugins prévus à cet effet (cf Figure 12).

Dernièrement, l'onglet « Exploit », nous permet simplement de savoir quel plugin a réussi à trouver la vulnérabilité que nous allons tenté d'exploiter (cf Figure 13).

Grâce à la Figure 14., nous remarquons que la vulnérabilité de type injection SQL fut trouvé grâce aux deux plugins « mysqlWebShell » et « sqlmap »

Nous allons donc vérifier si les dires de w3af sont vrais en tentant des tests manuels. D'après w3af, la page add_comment.php est vulnérable à des attaques de type Injection XSS.

Ce commentaire est une injection XSS vraiment très basique spécifiant simplement qu'une boite de dialogue va s'ouvrir dans le cas ou le page est réellement vulnérable. Enregistrons le commentaire et allons le visualiser (cf Figure 15).

Sur la page permettant de visualiser les commentaires, nous voyons bien la présence de notre boite de dialogue spécifiant ainsi que l'injection XSS a bien fonctionnée et donc que w3af a bel et bien fait son travail.

Conclusion

En conclusion, Web Application Attack and Audit Framework est un framework d'audit et d'exploitation des vulnérabilités des applications web extrêmement complet, pratique et simple d'utilisation

Tant son interface graphique que sa ligne de commande sont réellement complètes et permettent aux professionnels de la sécurité des systèmes d'informations, des audits de qualités extrêmement précis.

A noter également que w3af est présent dans l'excellente distribution entièrement consacrée au test de pénétration des applications web : Samurai Web Testing Framework.

A PROPOS DE L'AUTEUR...

Régis SENET est actuellement étudiant en dernière année à l'école Supérieur d'informatique Supinfo. Passionné par les tests d'intrusion et les vulnérabilités Web, il tente de découvrir la sécurité informatique d'un point de vue entreprise. Il est actuellement en train de s'orienter vers le cursus CEH, LPT et Offensive Security.

Contact : *regis.senet@supinfo.com*

Page d'accueil : *http://w3af.sourceforge.net/*