Wat zijn bestaande web scan applicaties?

In de afgelopen decennia zijn er heel wat web security scanners op de markt verschenen, zowel commercieel als open source (gratis). OWASP heeft een lijst gemaakt van 40 web security scanners, deze worden vaak gecategoriseerd als Dynamic Application Security Testing (DAST). Alle DAST tools zijn zoals eerder vermeld commercieel of open source en hebben allen hun eigen sterktes en zwaktes [OWASP DAST]. OWASP heeft een Benchmark project uitgebracht die de effectiviteit van een DAST tool meet[OWASP Benchmark].

In dit deel zal ik onderzoek doen naar een aantal DAST tools om inzicht te krijgen in de werkwijze van de verschillende scanners. Ieder scanner verschilt in omvang, scan proces, software architectuur, het analytische vermogen, de wijze van rapporteren, etc.. Het is interessant om een aantal te bestuderen om zo hun werkwijze te leren kennen. Dit zal mij nieuwe inzichten bieden en zal mij helpen mijn eigen web security scanner te verbeteren.

Een DAST tool heeft drie kern processen:

Crawl & Scan

Voordat er gescand kan worden moeten er eerst resources worden opgehaald. De crawler zorgt daarvoor door een delen van een HTML pagina te scrapen. Scrapen is het proces van het selectief uitknippen van html elementen zoals de anchor tag <a>, wat als attribuut een hyperlink. Wanneer er voldoende gecrawld is kan een scan gestart worden.

Detect & Alert

Bij het detecteren van kwetsbaarheden is precisie dat telt. Tijdens de scan worden probeert de DAST tool kwetsbaarheden in een web applicatie te detecteren. Om te kunnen meten hoe goed en accuraat een DAST tool kwetsbaarheden detecteert kan de OWASP Benchmark gebruikt worden.

Prioritize & Manage.

Na een scan worden alle gevonden kwetsbaarheden geprioriteerd in een geïndexeerde lijst en gerapporteerd. De resultaten van de scans worden opgeslagen en kunnen herzien worden voor analytische doeleindes.

Spider

Pen-test

De scanners die ik voor dit deel heb gekozen zijn:

* Acunetix
* ZAP Zed Attack Proxy

Ik zal in dit onderzoek twee documenten behandelen: Acunetix brochure en de start gids van OWASP ZAP. De twee documenten bieden inzicht in de werking van beide DAST tools.

Acunetix

Volgens een brochure van Acunetic heeft 70% van alle websites een kwetsbaarheid dat kan leiden tot diefstal van gevoelige data [Acunetix Brochure]. Acunetix maakt een paar goede punten in het brochure zoals dat hackers zich concentreren op het volgende: Componenten van webapplicaties als winkelwagens, formulieren, login pagina’s en dynamische content. Dit zijn voor een grootdeel waar de web security scanner die ik ontwikkel op zal focussen. Een ander punt die Acunetix maakt is dat web applicaties 24/7 toegankelijk zijn en dat zij controle hebben over kostbare data omdat zij, als het niet via een API gaat, in de meeste gevallen directe contact hebben met de backend, hoe anders moeten zij aan de data komen. Dit allemaal wordt in de eerste paragraaf toegelicht wat naar mijn mening een zeer goede inleiding is voor het document. Verder in het document vermeld acunetix dat netwerk securtiy amper bescherming biedt tegen web applicatie aanvallen, dit komt doordat de netwerk poorten 80/443 altijd openstaan. De reden dat deze poorten altijd openstaan is omdat zij iedere bezoeker moeten toelaten van de website. Anders heeft het geen nut om een website te hosten.

Web versie

Zoals je kunt lezen is Acunetix een bedrijf dat zich bezig houdt met de security van web applicaties. Hiervoor hebben zij een DAST tool ontwikkeld. Acunetix heeft een desktop versie ontwikkeld en een web versie. Ik zal in dit deel de web versie behandelen. De Acunetix Web Interface is een gebruikersvriendelijke web applicatie die de gebruiker na het inloggen naar een dashboard brengt. Vanaf het dashboard kunnen gebruikers vier management taken uitvoeren. Het configureren en beheren van Targets, het starten van een scan, het bekijken van statistieken en het genereren van een rapport. Verder zijn er nog andere kleinere taken zoals het beheren van de gebruikers profiel. Targets zijn websites die de gebruiker als doelwit heeft geregistreerd voor de voorbereiding van een scan. Na het instellen van een target kan de gebruiker de scan starten. Voor het starten van de scan kan de gebruiker nog een aantal opties instellen. De opties zijn verdeelt over vier tabs, General, Crawl, HTTP, Advanced. In de tab General kan de type scan ingesteld worden, de snelheid van de scan en de logingegevens. In de Crawl tab staan alle opties voor het instellen van de crawler, een crawler zorgt ervoor dat de benodigde informatie wordt verzameld voor de scan. Zoals de hyperlinks en formulieren. De HTTP tab is voor het authentiseren van een gebruiker tijdens de scan. De scan kan een login of registratie formulier tegenkomen tijdens de scan en weet dan met deze optie wat er ingevuld moeten worden. De laatste tab Advanced is ervoor de geavanceerde instellingen van de scan. Opties als technologie, Custom Headers, Custom Cookies en Allowed Hosts kan de gebruiker instellen om de scan te finetunen. Na de scan kan de gebruiker navigeren naar het Scan Stats & Info scherm, hierop kan er naar de statistieken gekeken worden voor het analyseren van het resultaat, ook kan de gebruiker een lijst opvragen met alle uitgevoerde aanvallen en wat het resultaat daarvan is, dit vereist wel technische kennis over IT security. Verder kan er nog naar de site map gekeken worden waar een lijst met alle gescande bestanden staan. Voor elk bestand staat er welke kwetsbaarheden gevonden zijn. Als laatste kan de gebruiker een rapport genereren op basis van de gescande resultaten. Bij het genereren van een rapport kan de gebruiker een template kiezen die de thema een doelgroep van het rapport bepalen. Acunetix is met al de opgenoemde features één van de voorlopers op gebied van web security scanners.

Scannen met Acunetix

Zap Zed Attack Proxy

Zed Attack Proxy is een flagship project van Open Web Application Security Project. Het is een open-source security scanner tool dat onderhouden wordt door OWASP. Het is gratis te downloaden van de website van OWASP en ook biedt OWASP project samenwerking voor software ontwikkelaars die een bijdrage willen maken door te werken aan nieuwe features voor de applicatie. Het project is open-source wat betekent dat OWASP de broncode publieke heeft vrijgegeven. Het project kan worden gedownload op Github hiervoor het een software ontwikkelaar geen toestemming voor nodig. Met een simpele git clone wordt er een kopie van het hele project op jouw computer geplaats.

De security scanner tool is special voor het testen van web applicaties ontwikkelt. Eerder in dit document heb ik kort samengevat wat een security scanner in het algemeen doet. OWASP heeft in het document [OWASP ZAP 2.6 Getting Started Guide] dit onderwerp zelf behandelt. In het hoofdstuk Security Testing Basics definiëren zij security testen als “Software security testen is het proces van beoordelen en testen van systemen om security risico’s en kwetsbaarheden te ontdekken”. Er worden hier twee termen gebruikt die tot de basis taken behoren van een security scan tool: beoordelen en testen. Zij definiëren het beoordelen als het analyseren en ontdekken van kwetsbaarheden zonder de poging tot het exploiteren van deze kwetsbaarheden. Het term testen wordt gedefinieerd als het ontdekken en poging tot exploiteren van kwetsbaarheden. Beide taken hebben als doel kwetsbaarheden te ontdekken, maar het verschil zit in wat zij doen met de kwetbaarheden die ontdekt zijn. To exploit or not to exploit.

In het hoofdstuk [Security Testing Basics] verdelen zij de basis van security testen in vier categorieën:

**Kwetsbaarheid beoordeling**

Het systeem is gescand en geanalyseerd voor security problemen.

**Penetratie test**

Het systeem ondergaat gesimuleerde aanvallen en het systeem wordt geanalyseerd.

**Runtime test**

Het systeem wordt geanalyseerd en het ondergaat een security test van een eindgebruiker.

**Code review**

De code van het systeem wordt gereviewd en het wordt specifiek geanalyseerd op security kwetsbaarheden.

ZAP proxy

Het doel van het document is natuurlijk om een introductie te maken voor OWASP ZAP. Ik heb eerder al een aantal dingen verteld over de DAST tool, maar ik zal er wat verder op in gaan in dit deel. Fundamenteel is ZAP een interceptie proxy tool, het staat tussen de eindgebruiker en webbrowser. De DAST tool onderschept berichten die gestuurd worden door de webbrowser naar de eindgebruiker in dit geval de tester. Na de onderschepping worden de berichten door ZAP geïnspecteerd en als nodig aangepast, dit staat ook wel bekent als de ‘man in the middle’ aanval. Mocht er al een proxy in gebruik zal, vele bedrijven hebben dit, dan kan Zap verbinding maken met de gebruikte proxy.

[Diagram van zap proxy]

ZAP client

ZAP biedt voor alle grote besturingssysteem platformen een versie en is gemaakt voor zowel experts op gebied van cyber security en beginners. Zap biedt ook vele add-ons voor de ZAP DAST tool, deze zijn te vinden op de ZAP marktplaats. ZAP wordt onderhouden door een grote gemeenschap die onderhoud verrichten en geregeld nieuwe add-ons (toevoegingen van features) ontwikkelen en publiceren op de marktplaats. De ZAP client applicatie maakt gebruik van een User Interface waar de eindgebruiker de verschillende taken kan uitvoeren. Het design van de UI kwam mij bekent voor omdat het gebruik maakt van de Java Swing thema Nimbus. Dit betekent ook dat ZAP proxy ontwikkeld is in Java. De ZAP UI bestaat uit 6 onderdelen: menubalk, takenbalk, boomstructuur venster, werkruimte venster, informatie venster, footer. Voordat er een pentest(penetratie test) uitgevoerd kan worden zal de proxy als eerst geconfigureerd moeten worden. De UI van Zap maakt het configureren zeer gemakkelijk, zoals het invoeren van een nieuwe SSL certificaat of het instellen van een nieuwe proxy port. De Zap client biedt verschillende scan configuraties voor het testen van een webapplicatie. De meest opvallende is de Snelle start test, deze test optie krijgt de eindgebruiker te zien wanneer de applicatie is opgestart. Om hiervan gebruik te maken voert de eindgebruiker de url in de tekstbalk en druk vervolgens op ‘Aanval’ om de test te starten. Er is wel een disclaimer, om een website te testen heb je wel toestemming nodig. De ZAP client zal, na het starten van de test, de website doorzoeken naar webpagina’s om deze vervolgens elke gevonden webpagina passief te scannen. Na het passief scannen zal de ZAP client overgaan naar het actief scannen van de webpagina’s. Het doel van een passieve scan is om het voorwerk te doen voor de actieve scan. De passieve scan leest en neemt alle verkeer op dat tussen de browser en website wordt gecommuniceerd. Dit betreft de GET/POST requests en de responses ervan. Dit is de wijze waarop een webbrowser client communiceert met de webserver, door request(een verzoek voor een webpagina) en een response(antwoord op het verzoek). Na dit proces analyseert ZAP client de data en kijkt of er ‘known issues’ (bekende problemen) gevonden zijn. Actief scannen is meer gericht op het aanvallen van de gevonden ‘known issues’. Bij het actief scannen worden er echte aanvallen uitgevoerd, dat betekent dus dat het doelwit risico’s kan lopen. Dit is ook een reden waarom je eerst toestemming moet hebben voordat je een test mag uitvoeren. Voor meer informatie over de ‘known issues’, deze worden in het hoofdstuk OWASP top tien lijst behandeld.

Wat is het resultaat na het scannen van de DVWA voor beide DAST tools?

In dit hoofdstuk ga ik kijken hoe de twee DAST tools de DVWA website scannen en wat voor bevindingen zij maken na de scan. Het doel van dit hoofdstuk is om een aantal vragen te beantwoorden zodat ik meer inzicht krijg in de werking van een DAST tool. Vragen die ik graag beantwoord wil hebben zijn: “op wat wordt er gescand, hoe ziet de datastructuur eruit, wat voor data er verzamelt wordt?, hoelang duurt een scan, hoe worden de rapporten gegenereerd”?

Omdat ik Acunetix als eerst heb behandelt in dit document zal ik daarmee beginnen. De DAST tool van Acunetix biedt een webapplicatie aan. Voor gebruik van deze webapplicatie heb ik als eerst een account moeten aanmaken, ook heb ik gemerkt dat het gaat om een trail versie. Acunetix is in tegenstelling van OWASP ZAP geen open-source software applicatie, maar dat betekent niet dat OWASP ZAP minder goed presteert als security scanner.

Wat is DVWA, het doelwit van deze tests?

DVWA staat voor Damn Vulnerable Web Application. De naam zegt het al, deze website is special gemaakt voor pen-testen en is daarvoor zeer kwetsbaar voor ‘known issues’ aanvallen zoals injecties en Cross site scripting. Het is een PHP/MySQL webapplicatie dat als doel heeft een hulpmiddel te zijn voor security professionals om hun skills te testen in een veilig en legale test omgeving.

Om te beginnen…

Bij het starten van DVWA verschijnt er in de browser een login pagina waar de gebruiker zijn gebruikersnaam en wachtwoord moet intypen om gebruik te kunnen maken van de web applicatie. Gelukkig is dit een lokale installatie en kan ik in mijn lokale database spieken om zo aan de twee accountgegevens te komen. De login scherm is voor een DAST tool een hindernis, wanneer er geen inloggegevens zijn meegegeven bij een scan dan kan de DAST tool niet verder scannen dan de login pagina. Dit resulteert dan in een incomplete scan een website (het doelwit) en geeft vrij weinig informatie over de security toestand van het doelwit. Zowel Acunetix en ZAP bieden de mogelijkheid om voor de scan inloggegevens mee te geven waarmee zij tijdens de confrontatie met het inlogscherm probleem loos verder kunnen gaan door in te loggen. DVWA kent een aantal configuraties maar, waar ik het meest in geïnteresseerd ben is de Security level van de webapp. Je kunt de Security level instellen op low(laag), medium, high(hoog) of impossible high(onmogelijk hoog). Dit zal de kwetsbaarheid level van DVWA veranderen en zorgt voor een dynamische test omgeving. Voor deze test zal ik de Security level op low plaatsen om zoveel mogelijk kwetsbaarheden te vinden.

Beide DAST tools zullen de lokale website <http://localhost:8000>, Hierop draait DVWA. Omdat er een inlog scherm is zullen beide DAST tools de inloggegevens username: admin en password: password gebruiken. Ik zal de scan configuratie erbij zetten voor elke DAST tool. De mogelijke configuraties verschilt per DAST tool. De configuraties kunnen op sommige punten verschillen

Test van Acunetix

Acunetix scan configuratie

* Target: <http://localhost:8000> – dit is de url van de doelwit
* Business Criticality: normal – dit bepaalt hoe crusiaal deze scan is voor jouw onderneming
* Scan speed: fast – snelheid van de scan
* Authenticatie: username: ‘admin’ & password: ‘password’ - inloggegevens
* Scan type: full scan – dit is een volledige scan er zijn naast deze 5 andere soorten.
* Report: none – Keuze om gelijk een rapport te genereren
* Schedule: Instant – planning van de scan. Voor deze scan wil ik 1 instantie.

Scan resultaat

Algemene informatie over de scan

* De scan heeft 11 minuten geduurd.
* Er zijn 35,170 request gedaan.
* Er zijn 121 locaties gevonden en gescand.
* Gevonden kwetbaarheden
  + Er zijn 11 hoge kwetsbaarheden gevonden met een hoge risico
  + Er zijn 32 kwetsbaarheden gevonden met een medium risico
  + Er zijn 78 kwetsbaarheden gevonden met een lage risico

Acunetix geeft DVWA, met low security level, een risico level van 3 wat de hoogst haalbaar is. Acunetix zegt zelf over deze risco level: ‘Kwetsbaarheden die gecategoriseerd zijn al meest gevaarlijk, het doelwit loopt maximale risico op om gehackt te worden en de diefstal van data’.

[afbeelding van risico level]

Hoe ziet de datastructuur eruit?

Als je een DAST tool ontwikkeld dan is het van belang om data over de scan te verzamelen en deze op een gestructureerde manier op te slaan in een database. Data is cruciaal voor het analyseren van een scan, de gebruiker wil immers achteraf kunnen zien hoe de scan is verlopen. Acunetix verzameld om deze reden veel data en presenteert het op een gebruiksvriendelijke manier naar de eindgebruiker toe.

Na de scan heb ik de data van scan bekeken om te kunnen achterhalen welke entiteiten, tabellen en kolommen Acunetix gebruikt. Dit heb ik gedaan omdat ik wil weten hoe de datastructuur eruit ziet, wat van belang zal zijn bij het ontwikkelen van mijn proof of concept. Er zijn naar mijn mening vier entiteiten: Doelwit, Scan, Kwetsbaarheid en Rapport. Ik heb voor ieder entiteit een tabel gemaakt met de bijbehorende kolommen en daarbij de beschrijving van de kolom.

* Doelwit – Algemene informatie over de website
* Scans – Informatie over de configuratie van een scan
* Kwetsbaarheid – informatie over de gevonden kwetsbaarheden van het doelwit
* Rapport – Algemene informatie over de rapport

|  |  |
| --- | --- |
| Doelwit(Target) informatie | |
| Adres | Host naam |
| Server | Server naam |
| Besturing systeem | Naam van besturing systeem |
| Technologieën | Programmeer talen die gebruikt zijn |
| Responsive | Is de website schaalbaar of niet |

|  |  |
| --- | --- |
| Scan | |
| Doelwit | Base URL van website |
| Scan Type | Er zijn 6 verschillende scan soorten. |
| Rooster | Dit geeft aan wanneer een scan is ingepland |
| Status | Dit geeft de voortgang van de scan aan |
| Datum | Start en einde van een scan |
| Requests | De aantal HTTP request die gemaakt zijn |
| Locaties | De aantal gescande web pagina |

|  |  |
| --- | --- |
| Kwetsbaarheid | |
| Risico level | Geeft aan hoe gevaarlijk een kwetsbaarheid is |
| Kwetsbaarheid | Type kwetsbaarheid |
| URL | URL waar de kwetsbaarheid is gevonden |
| Parameter | De parameter die gebruikt is bij de aanval |
| Status | Status van de kwetsbaarheid |
| Laats gezien | Datum van wanneer de kwetsbaarheid gevonden is. |

|  |  |
| --- | --- |
| Rapport | |
| Rapport Template | Type template |
| Rapport Type | Type rapport |
| Doelwit | URL van doelwit |
| Gemaakt op | Datum waarop rapport is gemaakt |
| Status | Vooruitgang van rapport |

Test van OWASP ZAP

OWASP ZAP configuratie

OWASP ZAP werkt als een proxy die http requests/responses onderschept tussen browsers en webservers. Hiervoor heb ik een proxy netwerk opgezet op mijn lokale server met de port nummer: 81. Zodra de webbrowser: localhost:81 verbinding maakt met de webserver: 192.168.0.101 zal OWASP ZAP de http communicatie onderscheppen en het uitlezen en aanpassen zodat er aanvallen verricht kunnen worden. De sitemap van DVWA bestaat uit een aantal sub mappen waaronder de sitemap vulnerabilities, hierin staan de pagina’s waar aanvallen verricht kunnen worden. Een normale quick scan slaat deze map over. Om OWASP ZAP naar deze map te wijzen moet er een nieuwe Context gemaakt worden van de sitemap vulnerbilities. Zo kan OWASP ZAP de submappen probleemloos vinden.

* Target: http://192.168.0.101 – dit is de url van de doelwit
* Proxy server: <http://localhost:81>
* Context: http://192.168.0.101/vulnerbilities
* Authenticatie: username: ‘admin’ & password: ‘password’
* Scan type: Quick scan

Scan Resultaat

Vanwege de kleine scope zijn er veel minder request gedaan wat er voor zorgde dat de scan sneller klaar was.

* De scan heeft 19 seconde geduurd.
* Er zijn 3,073 request gedaan.
* Er zijn 170 locaties gevonden en gescand.
* Gevonden kwetbaarheden
  + Er zijn 2 hoge kwetsbaarheden gevonden met een hoge risico
  + Er zijn 4 kwetsbaarheden gevonden met een medium risico
  + Er zijn 8 kwetsbaarheden gevonden met een lage risico

Hoe ziet de datastructuur eruit?

Ik heb gemerkt dat OWASP ZAP veel meer data verzameld dan Acunetix. Acunetix is wat betreft het presenteren van data veel gebruiksvriendelijker en overzichtelijker. OWASP ZAP presenteert bijna alle data in tabellen vergelijkbaar met een MySQL tabel. Het analyseren van de data is dus ook en veeleisende klus. Om dit probleem op te lossen geeft OWASP ZAP de optie om rapporten te genereren. Er zijn drie type rapporten HTML, XML en MD(Mark Down) waarvan HTML en MD de meest overzichtelijke, XML is zeer onduidelijk te lezen.

De entiteiten, tabellen en kolommen zal ik zoals bij Acunetix in tabellen plaatsen. Ik heb 6 entiteiten gevonden: Waarschuwingen, Spider, Actieve scan, Geschiedenis, Http Sessies en Parameters.

Waarschuwingen – Waarschuwingen zijn de kwetsbaarheden die tijdens de scan zijn gedetecteerd.

Spider – De Spider zoekt naar alle pagina’s en indexeert hun als voorbereiding op de aanval. Ook bepaalt de spider wat de scope is van de scan.

Actieve scan – De actieve scan voert alle GET en POST aanvallen uit op de geïndexeerde doelwitten.

Geschiedenis – De geschiedenis houdt een overzicht bij van alle gemaakte requests.

Http Sessies – De Http Sessies zijn de sessies die gestart worden door de browser. OWASP ZAP gebruikt dit om te controleren of er een actieve sessie bestaat.

Parameters – Er wordt een lijst met parameters opgeslagen. Deze worden gebruikt om de aanvallen te verrichten.

|  |  |
| --- | --- |
| Waarschuwingen | |
| URL | Uniform Resource Locator |
| Risico | Geeft aan hoe gevaarlijk een kwetsbaarheid is |
| Vertrouwen | Geeft aan hoe hoog het vertrouwen is |
| Parameter | De parameter die gebruikt is in de query |
| Aanval | De tekst dat gebruikt voor de aanval |
| Bewijs | Bijgeleverde bewijs voor onderbouwing |
| CWE ID | Common Weakness Enumeration id |
| WASC ID | Web Application Security Consortium |
| Source | Checkt of het passief is of actief |

|  |  |
| --- | --- |
| Spider | |
| Verwerkt | Check of de URL is verwerkt in de scope |
| Methode | De HTTP Verb die gebruikt is |
| URI | Uniform Resource Identifier |
| Markeringen | Een opmerking bij een record |

|  |  |
| --- | --- |
| Actieve Scan | |
| Id | Id van de record |
| Request Tijdstempel | Datum en tijd van de request |
| Response Tijd | De reactie tijd van de server |
| Methode | De HTTP verb die gebruikt is |
| URL | Uniform Resource Locator |
| Code | De status code van de Response |
| Groote Resp. Header | De header van de Response |
| Reden | Dit geeft aan of de Response is gelukt of niet |
| RTT | Round-trip time |
| Grootte Resp. Body | De html van de response |

|  |  |
| --- | --- |
| Geschiedenis | |
| Id | De id van de record |
| Request Tijdstempel | Datum en tijd van de request |
| Methode | De HTTP verb die gebruikt is |
| URL | Uniform Resource Locator |
| Code | Status code van de Response |
| Reden | Dit geeft aan of de Response is gelukt of niet |
| RTT | Round-trip time |
| Grootte Resp. Body | De html van de Response |
| Hoogste Waarschuwing | Waarschuwing niveau |
| Opmerking | - |
| Tags | Categorieën van Requests |

|  |  |
| --- | --- |
| Http Sessies | |
| Actief | Geeft aan welke sessie OWASP ZAP gebruikt |
| Naam | Naam van sessie |
| Waardes van Sessie Tokens | De sessie token |
| Overeenkomende Berichten | De aantal berichten die verzonden zijn |

|  |  |
| --- | --- |
| Params | |
| Type | Type parameter |
| Naam | Naam van parameter |
| Gebruikt | Aantal keer gebruikt |
| Waarden | Hoeveel waarden er zijn |
| Veranderingen | Hoe vaak de parameter veranderd is |
| Markeringen | Opmerking |
| Waarden | De waarde van een parameter |