Intrusion Detection mit Snort und Kibana

Paul Robert Kästner und Samuel Ferraz-Leite

SNORT

- Network Intrusion Detection System
- ein Netzwerktraffic-Analysetool
- ein daemon lauscht an einem konfigurierten Netzwerkinterface und zeichnet Netzwerktraffic auf
- Aufzeichnung auf Layer 2, danach folgt Zusammensetzung in verschiede Protokolle durch Preprozessor Module
- Präprozessor-Module enthalten auch konfigurierbare Normalisierungsmodule, die normabweichende Pakete modifizieren da mittels solchen Angriffe auf das IDS selbst möglich sind
- Wichtigsten und Meistverwendete Präprozessoren:
    - Fraq IP Defragementer
    - Session Präprozessor
    - Stream5 Präprozessor
    - HTTP-Inspect Präprozesor sind der (Herstellen der Paketanordnung) und der TCP Reassembly Präprozessor Stream 5, welcher die TCP-Segmente zusammensetzt und der TCP Reassembly Präprozessor Stream 5, welcher die TCP-Segmente zusammensetzt
- Snort unter Linuxdistributionen weit verbreitet - fehlt in Debian 8 (Jessie), vorhanden in Debian 7 (Wheezy) und Debian 9 (Stretch, derzeit noch im Stadium testing)
- Konfigurationen und Rulesets befinden sich in einem Linuxsystem unter /etc/snort/rules
- es existiert eine Hauptkonfiguration /etc/snort/snort.conf
- Erkennung hauptsächlich über Snort Regeln, oder auch Module
- Verschiedene Module behandeln, verschiedene Angriffsformen, die nicht über Regeln abzudecken sind \
    - ARP Spoofing
    - Port-Scan
    - IP-Spoofing
    - DoS Angriff auf Layer 3 und 4 (ICMP flood, TCP Syn flood, etc.)
- Software selbst ist Open Source, jedoch werden Abonnements für verschiedene Regelsätze angeboten
- es existiert ein Community-Regelsatz der kostenfrei ist und von der Community aktiv gepflegt und regelmäßig aktualisiert wird

RULES

- Snort Rule beschreibt eine Signatur einen Datenflusses
- Datenfluss kann entweder eingehend, ausgehend oder bidirektional sein
- Rule besteht grob aus einem Kopfteil(head)und einem Inhaltsteil(body), dabei spezifiziert der head die folgenden Erkennungsmerkmale:
    - durchzuführende Aktion (alert, pass, drop)
    - zu erkennendes Protkoll auf Layer 3 (IP, ICMP) oder Layer 4 (TCP, UDP)
    - zu filternde Quell-IP Adressen (Aufzählung, Ranges, Subnetze) und Quell-Ports(Aufzählung)
    - zu filternde Fließrichtung des Datenstromes als Pfeilsymbol das von Quelle auf Ziel, umgekehrt zeigt, oder als Doppelpfeil eine bidirektionalen Strom deklariert
    - zu filternde Ziel-IP Adressen (Aufzählung, Ranges, Subnetze) und Ziel-Ports(Aufzählung)
- Aktionen beschreiben, wie mit den Paketen auf die mittels der jeweiligen Signatur erkannt wurden umgegangen werden soll, diese können zum Beispiel verworfen werden (drop), ohne Reaktion durchgelassen werden (pass), oder es kann ein alert ausgelöst werden, wobei die jeweiligen Pakete passieren dürfen, jedoch eine Eintrag über das vorgekommene Ereignis in einer Ausgabe vermerkt wird
- Die Art der Ausgabe kann in mitunter in eine Datei unter einem bestimmten Format (unified2) oder direkt in eine MySQL Datenbank geschrieben werden
- direkte Ausgabe in eine Datenbank ist nicht empfohlen, da die notwendigen Module sehr alt sind und kaum gewartet wurden und es bei diesen Modulen bekannterweise zu Performancebeeinträchtigungen kommen kann, welche sehr kritisch für die Echtzeitanalyse ist, wobei es zu Paketen kommen kann, welche verworfen werden, weil sie nicht prozessiert werden können.

- Barnyard2 ist ein Open Source Interpreter für das unified2 Format von Snort und verfügt über einen Spooler der diese Ausgabedaten liest und in eine Datenbank schreibt (Postgres oder MySQL können dafür konfiguriert werden)
- Der Inhaltsteil einer Regel beschreibt die Merkmale des zu erkennenden Datenstromes sowie enthält er Metainformation über die Signatur
- dazu gehört unter anderem die Angabe einer Nachricht oder auch Beschreibung für die Signatur selbst, sowie eine interne Signatur-Id, eine Revisionsnummer und verschiedene Eigenschaften des zu erkennenden Datenstromes
- essentiell dafür sind Suchmuster, die entweder als einfache Zeichenkette oder als regulärer Ausdruck angegeben und entsprechend vom IDS verarbeitet werden
- durch den HTTP-Inspect Präprozessor ist es möglich Merkmale des HTTP Protokolls konkret zu filtern, so zum Beispiel können direkt HTTP Header und Body für die Suche verwendet werden, sowie einzelne Eigenschaften, die hauptsächlich aus dem HTTP Header ableitbar sind
- darunter fallen zum Beispiel bzgl. HTTP Requests die HTTP Methode, HTTP URI, Cookies
- HTTP Response Eigenschaften können auch analysiert werden, dafür ist es jedoch notwendig, dass die Fließrichtung entsprechend angegeben wurde
- Ferner bietet die Rule-Engine von SNORT die Möglichkeit quantitative und bedingte Analysen durchzuführen
- Bzgl. quantitativer Analyse ist es möglich zu erkennen ob eine Regel in einem festlegbaren Zeitintervall angewendet werden konnte, wobei zusätzlich noch entweder eine Quell oder Ziel-IP angegeben werden kann
- dieser Regelmechanismus ermöglicht unter anderem die Erkennung von DoS Angriffen
- ein weitere Funktionalität der SNORT Rule-Engine ist die bedingte Analyse
- über Flags die im Kontext der Rule-Engine Flowbits genannt werden, kann die Anwendung von Regeln gesteuert werden, indem einzelne Regeln überprüfen ob durch eine bestimmte andere Regel ein solches Flowbit gesetzt wurde.
- Die Flowbits werden über eine Bezeichnung identifiziert und eine multiple Verwendung verschiedener Flowbits innerhalb einer Regel ist erlaubt, in diesem Fall entspricht die Überprüfung mehrerer Flowbits einer UND-Verknüpfung der jeweiligen Vorbedingungen
- Mit Hilfe dieser Funktion der Rule-Engine ist es möglich bzgl. einiger Angriffe wie z.B. bei den Webattacke Cross-Site Scripting und SQL-Injection zu überprüfen, ob diese jeweils erfolgreich waren und den entstandenen Schaden zu klassfizieren.
- Solche Regeln sind sehr komplex und individuell auf einen laufenden Dienst zugeschnitten und es können nur Fälle abgedeckt werden, bei denen in der Response individuelle Muster erkennbar sind oder erwartete Muster fehlen