

Discover. Determine. Defend.



Sourcefire Vulnerability Research Team (VRT)

SOURCEfire®
Security for the real world.



Die aktuelle IPS-Landschaft

Hersteller von Intrusion Prevention-Systemen (IPS) werben häufig damit, wie viele unterschiedliche Bedrohungen sie erkennen, können und wie schnell Sie auf neue Bedrohungen reagieren können. Viele Firmen gehen blind davon aus, dass diese Angaben stimmen. Diese Leichtgläubigkeit ist jedoch fehl am Platze, wenn entsprechende Nachweise fehlen.

Nicht überprüfbarer Schutz: Würden Sie irgendein Kopfschmerzmittel von einem Straßenhändler kaufen? Oder würden Sie eher eine Spalt-Tablette, Tomapyrin oder ein anderes zugelassenes Kopfschmerzmittel in der Apotheke kaufen? Die meisten Hersteller von IPS stellen gerne kühne Behauptungen auf, für die es keinen Nachweis gibt und die auch nicht nachprüfbar sind.

Unvollständiger Schutz: Würden Sie eine Alarmanlage für Ihr Auto kaufen, die Diebe davon abhält, über das Fenster auf der Fahrerseite einzudringen, die für die Beifahrerseite jedoch keinen Schutz bietet? Die meisten Hersteller von IPS bieten angeblich „Schutz“ vor Sicherheitslücken, decken dabei aber nur ganz bestimmte Angriffe ab.

Schutz nach dem Auftreten einer Bedrohung: Wenn Sie wüssten, dass sich Ihr Kind wahrscheinlich bei seinen Freunden eine gefährliche Krankheit einfängt, würden Sie erst handeln, wenn die Freunde erkrankt sind? Oder würden Sie das Kind sofort impfen lassen, bevor die Krankheit zuschlagen kann? Die meisten Hersteller von IPS bieten erst dann Schutz vor Schwachstellen, nachdem diese Angriffe bereits öffentlich bekannt geworden sind und Unternehmen angegriffen wurden.

Unzuverlässiger Schutz: Wenn die Feueralarmanlage in Ihrem Gebäude alle paar Tage Alarm bei der Feuerwehr schlagen und die Sprinkleranlage auslösen würde, obwohl es zu gar keinem Feuer gekommen ist, würden Sie diese dann weiter betreiben? Was wäre, wenn die Anlage einen echten Brand gar nicht erkennen würde? Die meisten Hersteller von IPS verwenden Signaturen, die die auslösenden Bedingungen einer Sicherheitslücke gar nicht richtig abdecken – und dadurch Fehlalarme auslösen oder echte Gefahrensituationen erst gar nicht melden.

Da die Angreifer immer raffinierter werden, müssen die Hersteller von IPS Schutz bieten, der folgendes nachprüfbar abwehrt:

- Alle möglichen Angriffe
- bevor bestimmte Angriffsmethoden bekannt werden
- ohne Fehlalarme auszulösen oder echte Gefahrensituationen überhaupt zu melden

Wenden Sie sich an das Sourcefire Vulnerability Research Team (VRT), die einzige Forschungsgruppe der Branche, die derartigen Schutz anbietet. Das VRT liefert den Regelsatz, der hinter dem 3D-System von Sourcefire steht. Dabei findet eine auf Schwachstellen basierende Methodik Anwendung, die dem Bedarf selbst komplexester IPS-Installationen gerecht wird.

Nachprüfbarer Schutz: Das Sourcefire VRT und die Regelsprache SNORT®

Viele Hersteller von IPS brüsten sich damit, wie schnell sie auf neu bekannt werdende Schwachstellen reagieren können. Nehmen wir beispielsweise das monatlich erscheinende „Microsoft Tuesday“. Microsoft informiert die Endanwender dort über zahlreiche Sicherheitslücken und gibt Patches heraus.



Da die Signaturen der meisten Hersteller von IPS nicht zugänglich sind, können die Kunden überhaupt nicht nachprüfen, vor was sie eigentlich geschützt sind. Oft behaupten die Hersteller von IPS, dass sie Schutz vor Schwachstellen bieten, was in Wirklichkeit aber gar nicht der Fall ist. Es kann aber auch vorkommen, dass durch (rechtmäßige oder unrechtmäßige) Verwendung der Funktionalität ein Alarm ausgelöst wird.

Das Sourcefire VRT schreibt den Standardcode für die Snort-Regeln, die beim Sourcefire 3D-System Anwendung finden. Snort-Regeln können von jedermann beliebig eingesehen und analysiert werden. Es lässt sich problemlos feststellen, ob sie auch wirklich die angegebenen Schwachstellen abdecken. Das Format der Snort-Regeln ist inzwischen ein Industriestandard, der von Sicherheitsexperten auf der ganzen Welt eingesetzt wird.

Das offene Format der Snort-Regeln ermöglicht den Kunden folgendes:

- Prüfen, ob eine Regel umfassenden Schutz vor Schwachstellen bietet.
- Erstellen neuer Regeln oder Ändern bestehender Regeln, um Probleme mit benutzerdefinierten oder proprietären Diensten feststellen zu können.
- Nutzen der bereits bestehenden, von hunderttausenden von Snort-Anwendern bereitgestellten Regeln

Um ebenfalls „offene Regeln“ zur Verfügung stellen zu können, haben mehrere Hersteller von IPS, die geschlossene Signaturen verwenden, Snort mittlerweile in ihre Produkte eingebaut! Es handelt sich dabei jedoch niemals um die standardmäßige Inspektionsmethode und ist bislang meist nur unzulänglich integriert.

Warum Signaturen und Exploit-basierende Erkennung nur wenig nutzen

Hacker, Computerfreaks, und Sicherheitsexperten bringen jede Woche Hunderte von neuen Exploits – Angriffe auf Schwachstellen – heraus. Diese Exploits werden dann im Handumdrehen modifiziert. Manche davon werden so umgeschrieben, dass sie in Angriffstools wie Metasploit, CANVAS oder CORE IMPACT integriert werden können. Wieder andere werden so modifiziert, dass sie verschiedene Shellcodes nutzen oder aber weitere Angriffsvektoren hinzugefügt werden. Demzufolge werden ständig zahlreiche neue Exploit-Varianten für jede Schwachstelle erstellt.

Viele Hersteller von IPS bekämpfen diese sich verändernden Angriffe, indem sie Exploit-spezifische Signaturen für neue Exploit-Varianten erstellen, sobald diese auftauchen. Signaturen funktionieren nur, wenn ein bestimmtes Exploit unterschiedliche Markierungen oder Eigenschaften aufweist. Die Hersteller müssen alle neu herausgegebenen Exploit-Varianten aufspüren, eine Signatur dafür schreiben, die Signatur testen und diese schließlich an ihre Kunden weitergeben. Bei diesem Lösungsansatz treten folgende Probleme auf:

1. **Exploit-basierter Schutz ist kein umfassender Schutz.**

Zu einer Schwachstelle gibt es nahezu endlos viele mögliche Exploits. Ein Exploit-basiertes System bietet nur für einen geringen Teil dieser Exploits effektiven Schutz. Hacker können bestehende Exploits so modifizieren, dass eine Schwachstelle weiterhin ausgenutzt wird, so dass diese von einem auf Signaturen basierenden IPS nicht entdeckt werden können.

2. **Die Exploit-basierte Erkennung hinkt immer hinter der aktuellen Bedrohung hinterher.**

Ein Hersteller von auf Exploits basierenden IPS kann erst dann eine Erkennung für ein Exploit entwickeln, wenn dieses tatsächlich aufgetreten ist. Dieses Exploit kann bereits Unternehmensnetzwerke lahm legen, während es noch identifiziert und klassifiziert wird. Bis das Exploit endlich identifiziert ist und eine Signatur dafür erstellt und getestet wurde, bietet ein Hersteller von auf Exploits basierenden IPS keinerlei Schutz.

3. **Signaturen sind unzuverlässig und lösen Fehlalarme aus oder melden echte Gefahrensituationen erst gar nicht.**

Da Signaturen keine Protokolle modellieren und keine exakten Auslösebedingungen identifizieren, sind sie wesentlich anfälliger für Fehlalarme oder die Wahrscheinlichkeit, dass sie Gefahrensituationen gar nicht erkennen. Bei legitimen Datenverkehr kommen manchmal Signaturzeichenfolgen vor, bei illegitimen Verkehr fehlen sie jedoch oft. Dadurch verringert sich das Vertrauen, das in die generierten IPS-Ereignisse gesetzt wird, erheblich.

4. **Für eine Exploit-basierte Erkennung sind ständige Aktualisierungen erforderlich.**

Immer wenn ein Hersteller von Exploit-basierten IPS eine neue Signatur herausgibt, muss diese von den Kunden heruntergeladen werden. Versäumt man mal an einem Tag diesen Download, kann dies bereits bedeuten, dass man nicht mehr umfassend vor aktuellen, sich weiter verbreitenden Bedrohungen geschützt ist. Der Signatursatz muss in erheblichem Maße verwaltet werden, wird schnell schwerfällig und begrenzt die Performance.

Warum Regel- und Schwachstellen-basierender Schutz wirklich sinnvoll sind

1. **Regeln bieten Schutz vor einer möglichen Ausnutzung von Schwachstellen.**

Wenn ein Protokollmodell richtig definiert ist und die Auslösebedingungen der Schwachstelle korrekt programmiert wurden, schützt die entsprechende Regel vor allen möglichen Exploits dieser Schwachstelle. Eine einzige Regel eines auf Schwachstellen basierenden IPS kann Hunderte bekannter Exploits abdecken – und darüber hinaus unendlich viele mögliche Exploit-Versuche. Ein Exploit-basiertes IPS erfordert hunderte von Signaturen und muss ständig aktualisiert werden, soll es diesen Schutz auch nur annähernd erreichen.

2. **Regeln schützen Kunden, bevor Exploits herausgegeben werden.**

Kunden, die auf Regel-basierenden Schutz setzen, schirmen ihre Systeme im Netzwerk bereits im Vorfeld auftretender Exploits ab. Bei Schwachstellen-basierendem Schutz gibt es zwischen dem Erscheinen eines bestimmten Exploits und einer Aktualisierung des IPS keinen Zeitpunkt, zu dem das System ungeschützt ist. Bereits wenn neue Exploits zu einer bestehenden Sicherheitslücke auftauchen, sind die Systeme voll und ganz geschützt.

3. **Regeln lösen zuverlässig aus, ohne Fehlalarme oder die Wahrscheinlichkeit, dass Gefahrensituationen nicht erkannt werden.**

Mit Hilfe von Protokollen wird gewährleistet, dass lediglich die betroffenen Felder und Kommunikationszustände auf Angriffe hin überwacht werden. Durch die Festlegung der erforderlichen Auslösebedingungen für eine Schwachstelle wird sichergestellt, dass nur illegitimer Datenverkehr angezeigt wird.

4. **Auf Schwachstellen basierender Schutz sorgt für eine überschaubare Anzahl von Aktualisierungen und Regeln.**

Eine Schwachstellen-basierende Regel kann hunderte bekannter (und unbekannter) Exploits abdecken. Es sind wesentlich weniger Aktualisierungen notwendig. Außerdem decken weitaus weniger Regeln viel mehr Exploits ab.

Das Sourcefire 3D-System und das Sourcefire VRT nutzen vollkommen Schwachstellen-basierende Regeln um Schutz zu bieten, bevor neue Gefahren auftreten.

Regelmethodik des Sourcefire VRT

Die Methodik zur Erstellung von Regeln durch das Sourcefire VRT umfasst folgende vier Komponenten:

- Untersuchen der Schwachstelle
- Modellieren des Protokolls
- Identifizieren der Auslösebedingungen
- Testen und Überprüfen der Annahmen



Das Ergebnis ist ein Regelsatz, der:

- in der Lage ist, alle möglichen Exploits einer Schwachstelle zu erkennen, bevor es zu der Bedrohung kommt
- für maximale Performance optimiert wurde
- alle Gefahrensituationen erkennt und möglichst keine Fehlalarme auslöst

Untersuchen der Sicherheitslücke

Sourcefire investiert beträchtliche Summen in die Untersuchung von Schwachstellen. Das VRT von Sourcefire beschäftigt einige der renommiertesten Sicherheitsexperten der Branche, darunter Autoren verschiedener Referenzbücher zum Thema Sicherheit.

Mehr als jedes andere Unternehmen konzentriert sich das Sourcefire VRT auf die Identifizierung aller möglichen Auslösebedingungen für Schwachstellen und die Erstellung von Protokollmodellen für Protokolle, die möglicherweise Angriffen ausgesetzt werden. Dazu sind erheblicher Sachverstand bei der Programmierung von Sicherheitsmaßnahmen, sowie genaue Kenntnisse der Netzwerkprotokolle und möglichen Angriffswege erforderlich. Außerdem müssen sich weniger gängige Protokolle sowie Anwendungen mit geschlossenem Quellcode zurückentwickeln lassen.

Als Entwickler von Snort bietet Sourcefire den einzigartigen Vorteil der Gemeinschaft der Snort-Nutzer. Mit mehr als 3 Millionen Downloads und über 150.000 aktiven Anwendern ist Snort das am weitesten verbreitete Intrusion Prevention-Produkt. Die Nutzergemeinschaft gibt bei neu erkannten Bedrohungen und Schwachstellen frühzeitig Warnmeldungen aus und trägt zu einem erweiterten Regelwerk der „Gemeinde“ bei. Außerdem liefert sie als zusätzliche Schutzmaßnahme sofort Rückmeldungen bei auftretenden Fehlalarmen und fungiert als Qualitätskontrolle bei neuen Regeln. Die Snort-Community bietet darüber hinaus Zugriff auf Anwendungen und Systeme, die intern nicht zur Verfügung stehen.

Das Sourcefire VRT nutzt außerdem eine Vielzahl kostenpflichtiger Forschungsberichte sowie öffentlicher und privater Security-Feeds, um Informationen zu neu herausgegebenen Sicherheitslücken zu erhalten.

Modellieren des Protokolls

Vor einigen Jahren konnte man Waren in einem Geschäft nur an einer Kasse bezahlen, an der Kassierer saßen. Diese mussten den Strichcode des Artikels einscannen und das Ergebnis auf einem Bildschirm überprüfen. Wurde die Playstation zum Preis von €200 vielleicht als Diätcola zum Preis von €1,25 verbucht? Da ein Strichcode-Scanner nichts über den Artikel wusste, zu dem der Strichcode gehört, stellte dies eine kostspielige Fehlerquelle für das Geschäft dar. Es konnte auch passieren, dass ein Betrug erst gar nicht aufgedeckt wurde, wenn der Kassierer versäumte, die Transaktion noch einmal gegenzuprüfen. Es machte keinen Sinn, dem Scanner die Entscheidung zu übertragen und den Kassierer aus dem Spiel zu lassen.

Heute gibt es bereits bedienerlose automatische Kassenanlagen, die einen Artikel wiegen und seinen Strichcode einlesen. Die meisten Artikel haben ein bestimmtes Gewicht und weisen einen bestimmten Strichcode auf. Künftig werden Kassensysteme wohl in der Lage sein, die Form der eingescannten Artikel visuell zu erkennen. Je besser das Protokollmodell wird und je mehr Informationen es zu einem Artikel erhält, desto eher ist es möglich, der Maschine die automatische Entscheidungsfindung zu überlassen. Die Protokollmodellierung ähnelt dem Festlegen von visueller Form, Gewicht und Strichcode eines Artikels, so dass dieser später erkannt werden kann: Genaues Identifizieren, wie ein

sicherheitsanfälliges Protokoll aussieht und funktioniert, damit sicherheitsanfällige Bereiche ausgemacht werden können. Die Informationen in einem Protokollmodell lassen sich in drei Komponenten unterteilen:

- Identifikatoren für Protokoll und Protokollversion
- Identifikatoren für den Kommunikationszustand
- Struktur der Pakete und Nachrichten

Sobald das Protokollmodell erstellt ist, wird es vom Sourcefire VRT genutzt:

- um die Erkennung auf das sicherheitsanfällige Protokoll zu begrenzen
- um die Erkennung auf die Kommunikationszustände zu begrenzen, in denen die Sicherheitslücke ausgenutzt werden könnte
- um die Erkennung auf die Felder eines Pakets und die Felder einer Nachricht zu begrenzen, die möglicherweise sicherheitsanfällig sind

Die Snort-Regelsprache ermöglicht die Modellierung beliebiger Protokolle innerhalb einer Regel. Tritt bei einem neuen Protokoll, das zuvor als sicher angesehen wurde, eine neue Schwachstelle auf, kann das VRT ganz schnell Schutzmaßnahmen dafür entwickeln. Andere Hersteller brüsten sich mit zahlreichen vorgefertigten Protokoll-Decodern, bieten jedoch nicht die Möglichkeit, Protokolle auf Signaturebene zu modellieren. Dadurch erfolgt die Erkennung bei neuen Protokollen nur sehr langsam.

Protokoll-Identifikatoren

SSL ist eines der gängigsten Protokolle, das heutzutage im Internet angewendet wird, da sich hiermit die Web-basierte Kommunikation verschlüsseln lässt. SSL-Datenverkehr findet typischerweise an Port 443 statt (dem Standardport für HTTPS). An diesem Port können jedoch verschiedene SSL-Varianten sowie andere Verschlüsselungsprotokolle kommunizieren. SSL v2, SSL v3, TLS und PCT sind recht gängige Möglichkeiten. Wenn eine Schwachstelle nur in SSL v2 ausgenutzt werden kann, führt die Suche nach Exploits in SSL v3-, TLS- oder PCT-Datenverkehr möglicherweise zu Fehlalarmen und verringert die Performance.

Deshalb ist die Identifikation bestimmter Protokolle und Protokollversionen Teil der Protokollmodellierung, so dass andere Protokollversionen und Protokolle aus der Überprüfung ausgeschlossen werden können.

Kommunikationszustände

Eine SSL-Sitzung wird folgendermaßen aufgebaut:

1. Ein TCP-Dreifach-Handshake findet statt.
2. Das Client-System sendet eine Client-Begrüßungsnachricht (HELLO).
3. Das Server-System antwortet mit einer Server-Begrüßungsnachricht (HELLO).
4. Beide Seiten tauschen wichtige Informationen aus, um festzulegen, welche Verschlüsselungsart verwendet werden soll.
5. Beide Seiten beginnen mit der verschlüsselten Kommunikation.

Eine Schwachstelle dürfte nur bei einem dieser Schritte oder „Kommunikationszustände“ vorhanden sein. Um Fehlalarme zu vermeiden und Performanceeinbußen zu begrenzen, sollten nur die betroffenen Zustände auf eine Ausnutzung einer bestimmten Sicherheitslücke hin untersucht werden.

Bei einer Sicherheitslücke in der Client-Begrüßungsnachricht (HELLO) auf bestimmten Servern sollte nur dann auf Exploits dieser Schwachstelle hin untersucht werden, wenn sich die Verbindung im Zustand „Client-HELLO“ befindet. Während des Server-HELLO, des Austauschs wichtiger Informationen oder der verschlüsselten Kommunikation sollte keine Überprüfung auf diese Schwachstelle hin ausgeführt werden.



Paketstruktur und Felder

Gehen wir davon aus, dass die Client-Begrüßungsnachricht (HELLO) (Status 2) sicherheitsanfällig ist.

Die Client-Begrüßungsnachricht (HELLO) in SSL v2 umfasst folgende Informationen:

```
char MSG-CLIENT-HELLO
char CLIENT-VERSION-MSB
char CLIENT-VERSION-LSB
char CIPHER-SPECS-LENGTH-MSB
char CIPHER-SPECS-LENGTH-LSB
char SESSION-ID-LENGTH-MSB
char SESSION-ID-LENGTH-LSB
char CHALLENGE-LENGTH-MSB
char CHALLENGE-LENGTH-LSB
char CIPHER-SPECS-DATA[ (MSB<<8) | LSB]
char SESSION-ID-DATA[ (MSB<<8) | LSB]
char CHALLENGE-DATA[ (MSB<<8) | LSB]
```

Bei einem Server ist möglicherweise nur eines dieser Felder potenziell sicherheitsanfällig. Nehmen wir einmal hypothetisch an, dass ein Programm das Feld CIPHER-SPECS-DATA fehlerhaft parst. Damit Exploits dieser Schwachstelle erkannt werden, sollten alle anderen Felder ignoriert werden.

Modellieren des Protokolls: Zusammenfassung

Wenn die Erkennung auf bestimmte Felder und Kommunikationszustände begrenzt wird, die möglicherweise sicherheitsanfällig sind, schließt eine gute Protokollmodellierung weitgehend die Gefahr von Fehlalarmen aus und erhöht die Performance.

Identifizieren der Auslösebedingungen

Auslösebedingungen sind die Bedingungssätze, die erfüllt werden müssen, damit sich ein Angreifer eine Schwachstelle zunutze machen kann. In der einfachsten Form vergleicht eine Auslösebedingung die Länge des Inhalts eines relevanten Feldes (das im Protokollmodell definiert ist) mit einem bestimmten Wert oder Wertesatz. Bei „erfolgreichem“ Vergleich ist auch die Auslösebedingung erfolgreich. Es gibt viele verschiedene Kategorien von Auslösebedingungen – beispielsweise:

- **Bereichs- und Schreibfehler – Speicherüberlauf, Um-eins-daneben-Fehler, usw.** Dabei handelt es sich um gängige, einfache Programmierfehler, die heutzutage für zahlreiche Sicherheitslücken verantwortlich sind.
- **Synchronisations- und Zeitfehler – Sicherheitslücken, die auftreten, wenn Ereignisse außerhalb einer erwarteten Sequenz auftreten.** Viele DoS-Angriffe gehören in diese Kategorie.
- **Protokollfehler – statischer Inhalt, statische Kennwörter, usw.** Ein Standard-Kennwort, das mit einem Produkt geliefert und oftmals nicht geändert wird, kann in diese Kategorie eingeordnet werden.

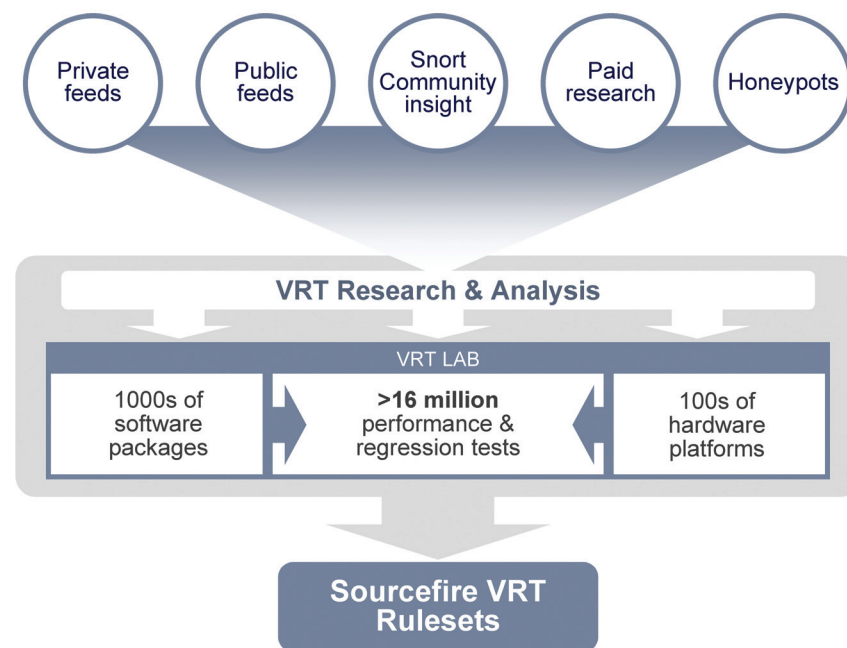
Beim Bau eines Hauses muss eine bestimmte Reihenfolge eingehalten werden: Bevor der Rohbau erstellt werden kann, muss ein Fundament gelegt werden. Der Rohbau muss fertig gestellt werden, bevor die Rohr- und Stromleitungen verlegt werden. Die Rohr- und Stromleitungen müssen eingerichtet sein, bevor der Trockenbau erfolgen kann usw.

Wie beim Hausbau müssen die Auslösebedingungen in einer bestimmten Reihenfolge geprüft werden, wenn es mehrere davon gibt. Um eine Auslösebedingung prüfen zu können, muss ein IPS ein Feld in einem Paket lesen. Da die Felder variable Größe aufweisen, weiß das IPS oftmals einfach durch Lesen dieses Feldes, wo es mit dem Lesen des nächsten Feldes beginnen muss.

Testen und Überprüfen der Annahmen

Nachdem ein Satz Auslösebedingungen festgelegt wurde, werden die Regeln einige Zeit lang Tests unterzogen. Damit soll gewährleistet werden, dass die Ausführung der Regeln in der Praxis keine Leistungseinbußen oder Fehlalarme nach sich zieht und dass alle Gefahrensituationen zuverlässig erkannt werden.

Das Sourcefire VRT führt ein automatisches Testpaket mit über 16 Millionen Überprüfungen gemäß Regelsatz aus. Automatische Testfälle werden erstellt und zum Testpaket hinzugefügt, wenn neue Schwachstellen gefunden und getestet werden. Diese Testfälle prüfen, ob die Regel nur dann ausgelöst wird, wenn der Satz der erforderlichen Auslösebedingungen auch wirklich im aktuellen Netzwerkdatenverkehr auftritt.



In der Praxis werden umfangreiche Tests beim Datenverkehr mit einer Vielzahl von verschiedenen Anwendungen und Betriebssystemen ausgeführt. Sourcefire setzt Sicherheitspartner und hunderttausende von Anwendern in der Snort-Community für weitere Überprüfungen ein. So wird gewährleistet, dass die Regeln des Sourcefire VRT zuverlässigen und präzisen Schutz bieten. Sourcefire prüft mit Hilfe hochwertiger Vorrichtungen zum Testen der Netzwerkperformance, dass Regeländerungen keine Leistungsprobleme hervorrufen.

Regelmethode des Sourcefire VRT: Alles zusammen – ein einfaches Beispiel

2001 wurde bei der RPC-Kommunikation mit dem ToolTalk Datenbankserver eine Schwachstelle bei den Betriebssystemen von HP, IBM, der SCO Group, SGI und Sun Microsystems (Bugtraq ID 122, CVE-1999-0003) gemeldet. Diese Schwachstelle ermöglichte es einem entfernten Angreifer, schädlichen Code mit den Rechten eines Superusers auszuführen.

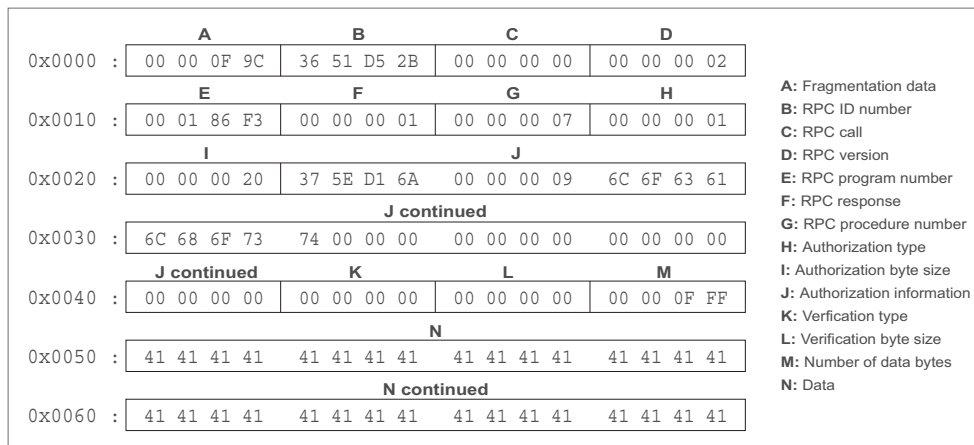
Exploit-Code wurde zur Verfügung gestellt, und verschiedene Hersteller von IPS erstellten Signaturen, die die Verwendung dieses spezifischen Exploit-Codes erkannten – beispielsweise die Zeichenfolge „\x80\x1c\x40\x11“, die in einem bestimmten Exploit vorkam.



Um dieser Schwachstelle zu begegnen, folgte das Sourcefire VRT der von ihm festgelegten Methodik. Man nutzte das RPC-Modell, identifizierte das sicherheitsanfällige Feld und den gefährdeten Kommunikationszustand, identifizierte die Auslösebedingung der Schwachstelle, erstellte eine Regel und prüfte die Annahmen hinter dieser Regel. Der Prozess wird nachfolgend beschrieben.

Protokollmodell

RPC-Pakete sind folgendermaßen strukturiert:



Protokollidentifikation

- Die Bytes 8 bis 11 weisen auf einen RPC-Aufruf hin. Für diese Bytes besteht also eine inhaltliche Übereinstimmung für „00 00 00 00“. Wir können direkt dorthin gehen und folgendermaßen in diesen vier Bytes nach dieser inhaltlichen Übereinstimmung suchen.

```
-- content:"|00 00 00 00|"; offset:8; depth:4;
```
- Die Bytes 12 bis 15 geben an, dass wir RPC Version 2 sehen.

```
-- content:"|00 00 00 02|"; offset:12; depth:4;
```
- Die Bytes 16 bis 19 geben die RPC-Programmnummer an. In diesem Fall interessiert uns „00 01 86 F3“. Wir können also zu dieser Position springen und folgendermaßen nach dieser bestimmten Programmnummer suchen:

```
-- content:"|00 01 86 F3|"; offset:16; depth:4;
```
- Wir wissen, dass es sich, wenn man ab diesem Punkt im Paket um vier Bytes weitergeht, um ein aus vier Bytes bestehendes Feld handelt, das die RPC-Prozedurnummer angibt. Uns geht es um „00 00 00 07“. Wir gehen also vier Bytes weiter und suchen folgendermaßen nach dieser Prozedurnummer:

```
-- content:"|00 00 00 07|"; distance:4; within:4;
```

Kommunikationszustand

- Uns geht es nur um Sitzungen, die bereits hergestellt wurden. Dies prüfen wir folgendermaßen:

```
-- flow:to_server,established;
```

Relevante Felder

- Zunächst müssen wir uns ansehen, wie viele Daten als Autorisierungsdaten verschickt werden, und dann hinter diese Daten springen. Diese Länge wird an einer Position vier Bytes weiter hinten im Paket gespeichert. Es handelt sich dabei um ein aus vier Bytes bestehendes Feld. Deshalb gehen wir folgendermaßen vor:

```
-- byte_jump:4,4,relative,align;
```

Das bedeutet, dass wir uns die Länge der übertragenen Daten ansehen und um diese Länge weitergehen können, um nach dem nächsten wichtigen Datenteil zu suchen.
- Der nächste Datenteil, der uns interessiert, ist die Menge der als Prüfung übermittelten Daten. Wir wissen, dass es sich hierbei um ein Feld handelt, das eine Länge von vier Bytes aufweist und vier Bytes weiter hinten im Paket zu finden ist. Somit können wir folgendermaßen vorgehen:

```
-- byte_jump:4,4,relative,align;
```

Das bedeutet, dass wir uns die Länge der übermittelten Daten ansehen und erneut vorwärts gehen können, um an diesen Daten vorbeizugehen.

Auslösebedingungen

- Wir befinden uns nun an einem Punkt in den Paketdaten, an dem wir prüfen können, wie viele Daten an das RPC-Programm gesendet werden. Wir wissen, dass die Informationen in einem aus vier Bytes bestehenden Feld unmittelbar hinter den Prüfdaten stehen. Da wir diese Daten übersprungen haben, können wir jetzt im Feldwert lesen und ihn mit einem Wert vergleichen, der anzeigen könnte, dass zu viele Daten übertragen wurden. Dazu können wir folgendermaßen vorgehen:

```
-- byte_test:4,>,128,0,relative;
```

Das bedeutet, wir lesen die vier Bytes ein, die angeben, wie viele Daten übertragen werden, und wir vergleichen diesen Wert mit einem Dezimalwert von 128, um festzustellen, ob er größer ist als dieser Wert. Wenn alle diese Bedingungen erfüllt sind, haben wir es mit einer Übereinstimmung zu tun, und es liegt möglicherweise ein Überlaufversuch vor.

Die vollständige Regel sieht folgendermaßen aus:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any \
(msg:"RPC tooltalk TCP overflow attempt"; \ flow:to_server,established; \
content:"|00 00 00 02|"; depth:4; offset:12;\
content:"|00 01 86 F3|"; depth:4; offset:16; \
content:"|00 00 00 07|"; within:4; distance:4; \
byte_jump:4,4,relative,align; \
byte_jump:4,4,relative,align; \
byte_test:4,>,128,0,relative; \
content:"|00 00 00 00|"; depth:4; offset:8; \ reference:bugtraq,122; \
reference:cve,1999-0003; \
classtype:misc-attack; sid:1965; rev:8;)
```

Diese Regel generiert jetzt Ereignisse für alle Versuche, die RPC-Schwachstelle von ToolTalk auszunutzen, im Gegensatz zu dem gerade veröffentlichten Exploit-Code.

Auswirkung und Kontext: Real-time Network Awareness (RNA) von Sourcefire

Sourcefire RNA bietet Sicherheitsanalysten mehr Kontext, über den die IPS der Konkurrenz nicht verfügen. Sourcefire RNA zeichnet sich durch folgende Vorteile aus:

- Umfassende Kenntnis aller Endpunkte.** Sourcefire RNA-Sensoren entdecken in Echtzeit die Systeme und Datenverkehrsmuster in Netzwerken. Sie horchen in Netzwerke, um neue und bestehende Hosts, Betriebssysteme und -Dienste zu identifizieren und mögliche Sicherheitslücken zu entdecken. Im Gegensatz zu den IPS-Lösungen der Konkurrenz, die mit nur selten aktualisierten, aktiven Scans, die viel Bandbreite beanspruchen, arbeiten, verwenden RNA-Sensoren umfassendes, passives Horchen. RNA-Sensoren können gezielte aktive Scans ausführen, um mehr Informationen zu bestimmten Hosts zu erhalten.
- Umfassende Erkenntnisse über das Netzwerk.** Die RNA-Sensoren von Sourcefire können anormale Datenverkehrsmuster, DDoS-Angriffe sowie interne Verbreitung von Würmern mittels der NBA-Funktion (Network Behavior Analysis, Analyse des Netzwerkverhaltens) erkennen. Sourcefire RNA erstellt Profile des Netzwerkdatenverkehrs und sucht mit Hilfe von statistischen Schwellwerten nach Abweichungen von diesen Basisdaten.

Defense Center von Sourcefire bestimmt die kritischsten Sicherheitsereignisse in Netzwerken anhand einer Kombination aus Daten von mehreren Intrusion-Sensoren und RNA-Sensoren von Sourcefire. Diese RNA-Informationen nutzen die Defense Center von Sourcefire, um Angriffe auf möglicherweise sicherheitsgefährdete Rechner zu priorisieren und andere zu depriorisieren. So kann unverzüglich auf hochkritische Ereignisse reagiert werden. Eine Abstimmung der Sensoren ist dabei nur noch in sehr geringem Maße oder sogar überhaupt nicht mehr erforderlich.



Auf Schwachstellen basierender Schutz im Vorfeld einer Bedrohung: Beispiele aus der Praxis

Sasser: Der Sasser-Wurm, der ungepatchte Systeme angegriffen und infiziert hat, auf denen Microsoft Windows XP® und Windows 2000® ausgeführt werden, trat 2004, 17 Tage nach der Herausgabe des ersten Patches, erstmalig auf. Die meisten Firmen hatten den Patch noch nicht installiert, als Sasser zuschlug. Sasser infizierte über eine Million PCs und brachte viele kritische Netzwerke zum Erliegen.

- **13. April 2004:** Microsoft identifiziert eine LSASS-Sicherheitslücke, die Windows XP und Windows 2000, Teil des Advisory MS04-011, betrifft.
- **15. April 2004:** Das Sourcefire VRT gibt Regeln heraus, die vor allen Exploits dieser Schwachstelle Schutz bieten.
- **30. April 2004:** Das SANS Internet Storm Center registriert Aktivitäten, die auf einen möglichen Wurm hinweisen, der auf die LSASS-Sicherheitslücke abzielt. Der Wurm versucht, LSASS mit Hilfe von DCE/RPC als Angriffsvektor auszunutzen.
- **1. May 2004:** Der Sasser-Wurm ist identifiziert.
- **2. May 2004:** Sourcefire benachrichtigt seine Kunden, dass die bestehenden Regeln des VRT für die Schwachstelle MS04-011 bereits Schutz vor den Exploits des Sasser-Wurms bieten.
- **3. May 2004:** Es sind verschiedene Varianten des Sasser-Wurms aufgetaucht: Sasser.B, Sasser.C und Sasser.D. Diese Varianten werden ebenfalls von den bestehenden Regeln des VRT abgedeckt.

Zotob: Der Zotob-Wurm trat erstmalig Mitte 2005 auf. Er richtet sich gegen Systeme, auf denen Microsoft Windows ausgeführt wird, und nutzt eine Schwachstelle im Plug-and-Play-Dienst aus. Er trat nur fünf Tage nach der Herausgabe eines Patches erstmalig auf und zwang die Firmen dazu, schnell zu reagieren.

- **9. August 2005:** Microsoft identifiziert eine kritische Sicherheitslücke im Windows Plug-and-Play-Dienst, Advisory MS05-039.
- **12. August 2005:** Das Sourcefire VRT gibt Regeln heraus, die vor allen möglichen Exploits der Plug-and-Play-Schwachstelle Schutz bieten.
- **14. August 2005:** Der Zotob-Wurm, der diese Sicherheitslücke ausnutzt, wird in Aktion identifiziert.
- **15. August 2005:** Sourcefire gibt bekannt, dass seine Kunden mit den bestehenden Regeln des VRT für die Schwachstelle MS05-039 bereits vor dem Zotob-Wurm geschützt sind.
- **17. August 2005:** Es sind verschiedene Zotob-Varianten aufgetaucht, und das VRT hat geprüft, dass alle von den am 12. August herausgegebenen ursprünglichen Snort-Regeln abgedeckt werden. Außerdem sind verschiedene weitere Exploits aufgetaucht, die die Plug-and-Play-Schwachstelle nutzen (Rbot, Sdbot, CodBot, drei IRCbot-Varianten sowie zwei Bozori-Varianten).
- **19. August 2005:** Sourcefire gibt Anleitungen zur Verwendung von RNA für die Visualisierung der Verbreitung des Zotob-Wurms heraus.

In allen Fällen hat das VRT von Sourcefire bereits vor Eintreten der Bedrohung für eine umfassenden Schutz gesorgt.



Zusammenfassung

Die Forschungen des VRT von Sourcefire sind ein wichtiger Bestandteil des 3D-Systems von Sourcefire. Eine nachweisbare Erfolgsbilanz zeigt, dass das VRT von Sourcefire seine Kunden bereits vor einem größeren Ausbruch von Malware schützt, darunter Nachi, Blaster, Sasser, Zotob sowie viele weitere, ohne weitere Aktualisierungen zur Abdeckung neuer Varianten zu erfordern.

Das 3D-System von Sourcefire nutzt folgende Funktionen, um Schutz zu bieten, bevor neue Bedrohungen entstehen:

- **Schwachstellen-basierende Regeln**
- **Eine leistungsstarke offene Regelsprache**
- **Die VRT-Methodik:**
 - **Untersuchen der Schwachstelle**
 - **Modellieren des Protokolls**
 - **Erstellen der Auslösebedingungen**
 - **Testen der Annahmen**
- **RNA:**
 - **Identifiziert die Art der Bedrohungen, die für Kundennetzwerke eine mögliche Gefahr darstellen**
 - **Abstimmung ist nur in sehr geringem Maße oder überhaupt nicht erforderlich**
 - **Liefert umfassende Kenntnisse über das Netzwerk und die Endpunkte**

Das 3D-System von Sourcefire – basierend auf dem 3D-Konzept (Discover, Determine, Defend = Entdecken, Entscheiden, Abwehren) zur Sicherung von Netzwerken in Echtzeit – vereint die Technologien für das Intrusion- und Schwachstellen-Management. Damit erhalten Kunden eine höchst effektive Netzwerksicherheitslösung zum Schutz gegen alle Bedrohungen aus allen möglichen Vektoren zu jeder Zeit.