



Die WAF und der Client

VON THOMAS KOHL, TERRITORY MANAGER GERMANY, AUSTRIA, SWITZERLAND, EASTERN EUROPE, DENY ALL

Bis heute war der wesentliche Fokus einer Web-Application-Firewall die Absicherung der auf Web basierenden Applikationen und deren Daten vor Hackerangriffen. Der Bedrohung von Unternehmensdaten durch Diebstahl und Manipulation jeglicher Art wird heute teilweise schon durch den Einsatz von WAF-Systemen begegnet. Aber es gibt in diesem Zusammenhang eine weitere Bedrohung für die Applikationen: Angriffe durch Spyware und Trojaner.

Es ist schwierig, Endsysteme frei von Schadsoftware und -programmen zu halten. Sobald diese Malware einmal auf einem Client/PC vorhanden ist, kann sie extrem intelligente Angriffe ausführen und in einigen Fällen sogar von außen gesteuert werden.

Dies ist eine neue Qualität von Bedrohung, die in den jüngsten Monaten immer häufiger in Unternehmensnetzen entdeckt wurde, mit zunehmender Tendenz. Zum Schutz vor solchen Angriffen muss die Web-Application-Firewall ihren Wirkungsgrad erweitern. Wesentlich hierbei ist ein Schutzmechanismus zur Sicherung des Kunden-Clients vor den beschriebenen Gefahren. Das Schlagwort hierfür lautet »Client Sanitization«, also die Erweiterung der Applikations-Sicherheit hin zum Client, der in diesem Zusammenhang oft als passive Gefahrenquelle fungiert.

Anders als traditionelle Antivirus-Lösungen schützt diese neuartige Security-Technologie ausgewählte Web-Anwendungen (wie Buchungsdienste, Online-Anwendungen), die webbasiert auf vertrauliche Informationen zugreifen oder diese verarbeiten. Der Schutzmechanismus wird in das zu schützende Programm

integriert und schützt dieses gegen Angriffe von außen.

Zunächst handelt es sich hierbei um eine weitere Sicherheitsinstanz, die in die bestehende Security-Architektur integriert werden muss. Idealerweise wird diese Funktion als weiteres Sicherheitsmodul in einer bereits installierten Web-Application-Firewall direkt aktiviert. Somit erfolgt eine Ausweitung der bisherigen reinen Application-Security (Black-, Scoring-, White-List usw.) um eine Schutzfunktion gegen jegliche Form von Angriffen auf Client-Ebene. Und das in einer Lösung. Diese neue Funktionalität arbeitet, gesteuert durch die zentrale Software-Instanz, in dem Browser-Programm des Clients. Die Schutzfunktion ist direkt mit den zu schützenden Informationen verknüpft. Die IT-Abteilung hat somit die Möglichkeit, den Applikationsverantwortlichen und den Anwendern ein geschütztes Produkt anzubieten, das sicheres Arbeiten von möglicherweise infizierten Clients aus erlaubt.

Ein wesentlicher Vorteil und somit eine organisatorische/wirtschaftliche Vereinfachung dieser Security-Funktion sind, dass für die Aktivierung sowie für den Betrieb weder separate

Installationen noch Administratorrechte seitens der End-User erforderlich sind. Des Weiteren ist es möglich, diese Security-Funktion nur für ausgewählte Anwendungen zu aktivieren.

Interne Programmüberwachung

Der neuartige technische Ansatz dieses Security-Moduls ist eine eigene Plattform, die in dem zu schützenden Programm arbeitet, es werden also interne Programmereignisse und die interne Programmausführung überwacht.

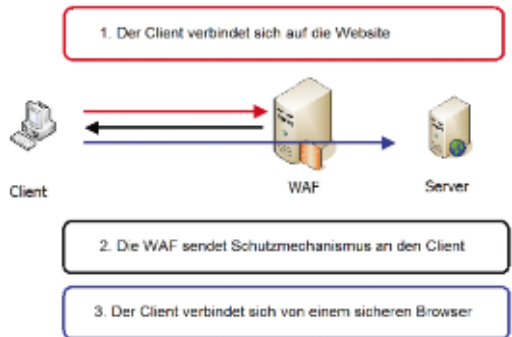
Das übergeordnete Ziel ist die sichere Ausführung eines Programms auf einem möglicherweise infizierten System. Sensible Daten im Programm sollten also gegen Diebstahl und Manipulation durch bösartige Programme geschützt sein. Dies ist durch einen »sauberen« (frei von Schadprogrammen) Programmraum auf dem Client zu erreichen, der durch die Funktion der Programmstartkontrolle, die aus einem Laderprogramm besteht, eingerichtet wird. Der Programmlader erstellt einen neuen Prozess und friert diesen in einem sehr frühen Zustand ein, deutlich bevor beispielsweise Debugger eingesetzt werden können. Direkt damit verbunden wird das Schutzprogramm in den neuen Prozess eingefügt und entschlüsselt. Dieses neue Programm enthält eine hohe Anzahl verschiedener Sensor- und Kontrollmechanismen, die die weitere Programmausführung kontrollieren. Die Ausführung des neuen und von Anfang an sauberen Prozesses wird fortgesetzt, und das Programm startet.

Folgende drei Hauptelemente bilden gemeinsam diesen neuartigen Security-Mechanismus:

- Programmausführungskontrolle
- Code-Injektionskontrolle
- Programmschnittstellenkontrolle

Programmausführungskontrolle

Der Programmausführungskontrollmechanismus schützt das Programm gegen Manipulation des Programmflusses durch externe Programme. Beispielsweise sind neue Prozessfolgen nur zulässig, wenn sie von innen heraus gestartet wur-



Schematischer Ablauf der Client-Security

den. Andere Programme, die Prozesse in dem geschützten Programm beispielsweise mit der Windows-Funktion »Create Remote Thread« erstellen wollen, werden somit blockiert.

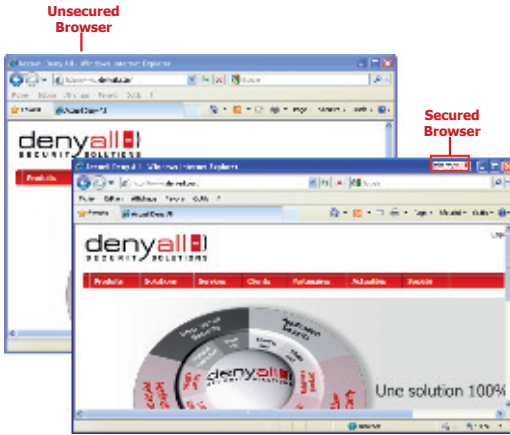
Code-Injektionskontrolle

Der Kontrollmechanismus für eine neue Code-Injektion in ein geschütztes Programm basiert auf der Überwachung und Kontrolle der Verarbeitung von neuen Code-Modulen, die in den Adressraum des Programms gebracht werden. Dies ist ein allgemeiner Mechanismus, der auf viele unterschiedliche Weisen eingesetzt werden kann.

Programmschnittstellenkontrolle

Diese neuartige Security-Funktion schützt externe Schnittstellen und den externen Datenstrom zum und vom Programm (beispielsweise für die Tastaturschnittstelle) auf folgende Weise:

- Verweigerung des Zugriffs zur Schnittstelle, so dass bösartige Programme nicht in der Lage sind, Informationen aus dem Datenstrom zu laden oder dort Daten einzufügen.
- Veränderung der Originaldaten im Datenstrom, so dass sie von den bösartigen Programmen nicht erkannt werden können, die sich erfolgreich in den Datenstrom eingeklinkt haben.



Unsecured / Secured Browser – optisch fast kein Unterschied

Die Schnittstelle, die beispielsweise Eingaben von der Tastatur empfängt, wird mit einer Kombination beider Methoden geschützt. Bei Microsoft Windows dürfen sich weitere Programme in die Tastaturschnittstelle einklinken, etwa mit der Funktion »SetWindows-Hook«. Dies kann dazu führen, dass ein Hook-Modul injiziert und dann über die Tastaturschnittstelle in das Programm eingeschleust wird. Die Schutzmechanismen blockieren solche injizierten Module (es sei denn, die Operation wurde aus dem Programm heraus injiziert). Um die Sicherheit der Tastaturschnittstelle noch weiter zu erhöhen, wird zusätzlich ein Rauschmodul implementiert. Dieses Modul erzeugt ein willkürliches Rauschen, beispielsweise in Form von Tastatureingaben auf dem Tastaturkanal, damit der Originaldatenstrom von der Tastatur für böswärtige Programme, die sich möglicherweise in den Datenstrom eingeklinkt haben, nicht erkennbar ist.

Auch die Programmschnittstelle für Windows-Meldungen ist geschützt, indem bestimmte Meldungen geblockt werden können, wenn sie nicht vom Programm selbst gesendet werden.

Kompatibilität

Ein wesentlicher Aspekt für den Erfolg einer solchen Security-Funktionalität ist die beinahe

unbeschränkte Einsatzflexibilität, bezogen auf die Vielfalt der bei den Kunden eingesetzten Browser-Hersteller und versionen. Hierfür muss ein komplexes und stets aktuelles Kompatibilitätskonzept in diesem Sicherheitsmechanismus integriert sein. Es gibt drei Hauptaspekte im Bezug auf Kompatibilität:

- Kompatibilität mit dem geschützten Programm
- Kompatibilität mit anderen Programmen im gleichen System
- Kompatibilität mit dem Betriebssystem

Ein zentraler Kompatibilitätsaspekt mit dem geschützten Programm ist die Garantie, dass keine Inkompatibilitätsprobleme die Schutzfunktionen beeinflussen. Andererseits dürfen die durchgesetzten Sicherheitsmechanismen auch keine Abweichungen vom normalen Programmverhalten zur Folge haben. Idealerweise sollte dies sowohl bei infizierten als auch nicht infizierten Systemen gleichermaßen der Fall sein.

Der aktuelle Sicherheitsmechanismus ist kompatibel mit allen 32-Bit-Versionen des Windows-Betriebssystems von Windows 2000 bis Windows Server 2008. Auch die 64-Bit-Versionen werden teilweise unterstützt.

Der Hauptaspekt der Kompatibilität mit den Betriebssystemen ist, dass es nicht abstürzen oder sich aufhängen darf, wenn Sicherheitsfunktionen ausgeführt werden, da eine der wesentlichen Anforderungen an Sicherheitsmechanismen der IT die sehr hohe Verfügbarkeit des Systems ist. Dies erweist sich nicht immer als trivial, da bei der Durchsetzung von Sicherheitsfunktionen oft das normale Verhalten des Betriebssystems geändert werden muss.

Aufgrund der Tatsache, dass die Sicherheitsmechanismen in Programme integriert sind und auf programminternen Prozessen basieren, gibt es mit anderen Programmen, die auf dem gleichen System ausgeführt werden, keine Kompatibilitätsprobleme. Einige gewollte Sonderfälle sind jedoch erwähnenswert. Debugger sind beispielsweise nicht in der Lage, auf die geschützte

Anwendung zuzugreifen. Nur der Anbieter der Anwendung sollte dazu in der Lage sein, niemals ein Endanwender, auch wenn er gute Absichten hat. Der Parallelbetrieb von beispielsweise Antivirenprogrammen, die auf Systemebene implementiert und gestartet sind, erweist sich als unkritisch, bezogen auf die Inkompatibilitätsprobleme.

Einsatzgebiete

Durch die neuartige Web-Application-Client-Security-Lösung »Client Sanitization« eröffnet sich IT-Abteilungen die Möglichkeit, für die eingesetzten Applikationen im Intra-, Extra- und Internet eine neue und noch höhere Güte von Sicherheit – ausgeweitet auf die Clients – zu bieten, als dies mit den bisherigen Web-Application-Firewalls bereits möglich ist. Mit dieser Funktion ist sichergestellt, dass ausgewählte unternehmenskritische Applikationen und die dazugehörigen Daten, egal von wo aus auf sie zugegriffen wird, gezielt geschützt werden können. Nutzt ein Anwender eine mit diesem Service geschützte Anwendung, wird das Sicherheitsmodul automatisch in den Internet-Browser des Anwenders übertragen. Somit wird eine geschützte Online-Sitzung ermöglicht. Typische Beispiele sind Online-Banking, SaaS (Software as a Service), Authentifizierungsmechanismen, e-Government, e-Payment sowie e-Health-Systeme.

Weitere Beispiele für die Einsatzmöglichkeit in unternehmenseigenen Intra- und Extranet-Strukturen zeigen auf, dass es unabhängig ist, von wo aus Projektmitarbeiter auf beispielsweise Forschungsdaten oder Außendienstmitarbeiter auf Kundendaten im Finanzsektor zugreifen. Es wird mit der neuen Security-Technologie sichergestellt, dass keine sensiblen Daten durch entsprechende Schadprogramme verloren gehen oder manipuliert werden. So ist es sogar möglich, dass beispielsweise ein Mitarbeiter aus einem Internet-Cafe am Flughafen gesichert auf die Unternehmensdaten zugreifen kann. Diese

Sicherheit wird hierbei zusätzlich gegeben, da nach dem Beenden der Arbeit und dem Schließen der Applikation auch die entsprechende gesicherte Session auf dem Client gelöscht wird. Dies gewährt den Mitarbeitern eines Unternehmens ein sehr hohes Maß an Flexibilität für ihre Arbeit bei gleichzeitiger modernster und höchster Sicherheit für die Applikationen und Daten des Unternehmens.

Schlussfolgerung

Mit dieser neuen Sicherheitsfunktion »Client Sanitization« wird den IT-Abteilungen und somit auch den Unternehmen selbst eine neue Security-Stufe geboten, die ein Zusammenrücken der Applikation und des Clients zu einer einheitlichen, gesicherten Arbeitsplattform darstellt. Durch die hohe Flexibilität und Kompatibilität sowie die einfache Betriebsführung ist diese Lösung ein weiterer Schritt, die Sicherheit für auf Web basierende Applikationen und deren dazugehörige Daten maßgeblich zu erhöhen. Durch die Integration dieser Technologie als Funktionsmodul in einer Web-Application-Firewall entsteht ein noch höherer Wirkungsgrad einer WAF bei gleichzeitig wirtschaftlichem Betrieb. Haupteigenschaften dieser Web-Application-Client-Security-Lösung:

- IT-Abteilung kann Security-Funktion für dezidierte Anwendungen anbieten
- Auf Sitzung basierender Schutz
- Geschützter Zugriff auf Anwendung trotz infizierten Systemen
- Extrem geringe zusätzliche Rechnerbelastung führt zu stabiler Systemgeschwindigkeit
- Keine separate Installation durch Endbenutzer erforderlich
- Keine Administratorrechte für Endbenutzer erforderlich
- Keine systemweiten Störungen mit Endanwendersystemen
- Einfache Aktivierung als Zusatzmodul in einer WAF